User's Manual

32

RH850/F1KH, RH850/F1KM

User's Manual: Hardware

Renesas microcontroller
RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.10 Dec, 2018

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Notes for CMOS devices

(1) Voltage application waveform at input pin: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

(2) Handling of unused input pins: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) Precaution against ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) Status before initialization: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) Power ON/OFF sequence: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) Input of signal during power off state: Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers This manual is intended for users who wish to understand the functions of the RH850/F1KH, RH850/F1KM and design application systems using the following RH850/F1KH, RH850/F1KM microcontrollers:

Purpose This manual is intended to give users an understanding of the hardware functions of the RH850/F1KH, RH850/F1KM shown in the *Organization* below.

Organization This manual is divided into two parts: Hardware (this manual) and Architecture (RH850G3KH User’s Manual: Software).

Hardware	Software
Pin functions	Overview
CPU function	Processor Model
On-chip peripheral functions	Register Reference
Flash memory programming	Exceptions and Interrupts
	Memory Management
	Instruction Reference
	Reset
	Appendix

How to read this manual It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the RH850/F1KH, RH850/F1KM.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850G3KH User’s Manual: Software (R01US0165E) available separately.

This RH850/F1KH, RH850/F1KM Hardware User’s Manual corresponds to Rev.1.10.

The RH850/F1KH-D8 description relates to a revision level of Rev.1.00.

The xxA section describes the functionality of RH850/F1KH-D8.

The xxB section describes the functionality of RH850/F1KM-S4.

The xxC section describes the functionality of RH850/F1KM-S1.

The xxAB section describes the functionality of RH850/F1KH-D8 and RH850/F1KM-S4.

The xxBC section describes the functionality of RH850/F1KM-S4 and RH850/F1KM-S1.

The xxx section describes the functionality of all related products.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representation: xxx (overscore over pin or signal name)

Memory map address: Higher addresses on the top and lower addresses on the bottom

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... xxxx or xxxx_B
Decimal ... xxxx
Hexadecimal ... xxxx_H

Prefix indicating power of 2 (address space, memory capacity):

K (kilo): 2^{10} = 1,024

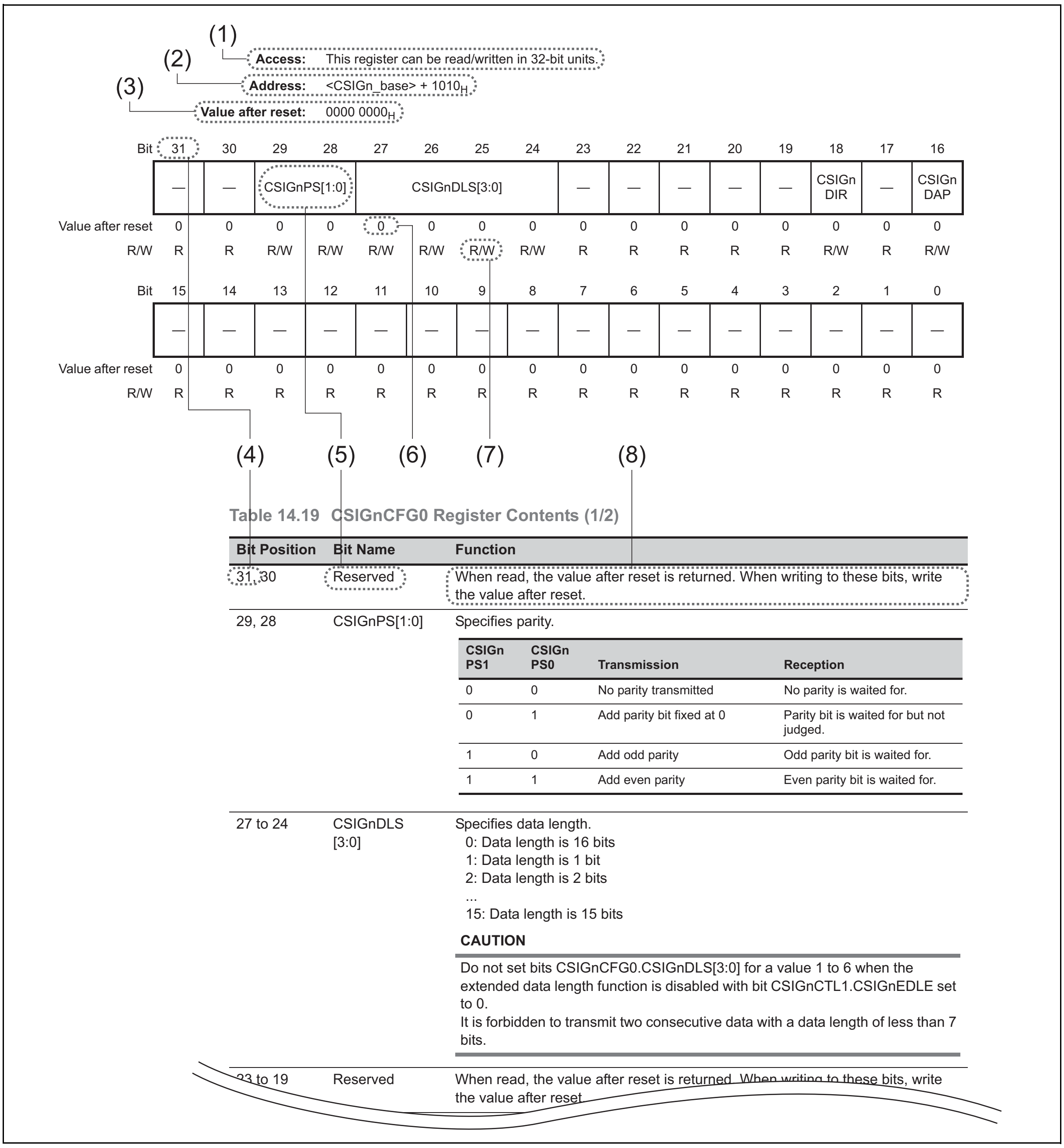
M (mega): $2^{20} = 1{,}024^2$

G (giga): $2^{30} = 1{,}024^3$

Description of Registers

Each register description includes register access, register address, and register value after a reset, a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables are described below.



Access: This register can be read/written in 32-bit units.

Address: <CSIGn_base> + 1010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGnPS[1:0]		CSIGnDLS[3:0]				—	—	—	—	—	CSIGn DIR	—	CSIGn DAP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.19 CSIGnCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.
27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** Do not set bits CSIGnCFG0.CSIGnDLS[3:0] for a value 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.
23 to 19	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	No parity is waited for.
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.
1	0	Add odd parity	Odd parity bit is waited for.
1	1	Add even parity	Even parity bit is waited for.

(1) Access

The register can be accessed in the bit unit indicated here.

(2) Address

This is the register address.
For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.
The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.
When clearly identifying the digits of a bit field is required, do so by using a form such as CSIGnDLS[3:0] above.
Indicate reserved bits by using a dash (—).

(6) Value after a reset (in binary notation)

This is the bit values after a reset.

0 : The value after a reset is 0.
1 : The value after a reset is 1.
— : The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W : The bit or field is readable and writable.
R : The bit or field is readable.
Note that all reserved bits are indicated as R. When written, the value specified in the bit chart or the value after a reset should be written.
In case of writing to writable registers that also include non-reserved bits with the R-attribute, writing to the R-attribute bits will be ignored unless otherwise specified.
W : This bit or field is writable. When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

Table of Contents

Section 24 CANFD Interface (RS-CANFD) 1936

Section 47A Electrical Characteristics of RH850/F1KH-D8..4240

Section 1A Overview of RH850/F1KH-D8

1A.1 RH850/F1KH Product Features

The features of the RH850/F1KH are described below.

The RH850/F1KH is a 32-bit single-chip microcontroller with two G3KH CPU core. The key features of the F1KH are low power consumption, high computational processing power, and a wide variety of internal peripheral functions. To reduce supply current in a variety of applications, a wide range of power reducing measures are available. For example, there is a Low Power Sampler (LPS), that can poll signals input to the analog and digital input pins without CPU core interaction, and DeepSTOP mode in which the power supply to the most circuits of the microcontroller can be turned off.

Applications

The RH850/F1KH is ideal for automotive electronics, such as BCM (body control module), gateway, HVAC, lighting modules, and many other applications.

1A.2 RH850/F1KH Functions

Table 1A.1 Overview of Product

			RH850/F1KH-D8		
Product Name			176 Pins	233 Pins	324 Pins
Memory			See **Table 1A.2, Product Lineup**.		
External Memory Access Controller (MEMC)			23 bit Address Bus		24 bit Address Bus
Serial Flash Memory I/F (SFMA)		Bus width	4 bit		
		Mode	SDR		
		Max. clock	40 MHz		
Memory Card I/F (MMCA)		Bus width	Not provided		8 bit
		Mode			Backward-compatible
		Max. clock			20 MHz
CPU	CPU System		G3KH (Dual Core)		
	CPU frequency		240 MHz max.		
	FPU		Single-precision		
	Protection Function	Memory Protection Unit (MPU)	Provided		
		Internal Peripheral-device Guard (IPG)	Provided		
		Processor Element Guard (PEG)	Provided		
DMA			64 channels		
Operating clock	Main Oscillator (MainOSC)		8/16/20/24 MHz		
	Low Speed Internal Oscillator (LS IntOSC)		240 kHz (typ.)		
	High Speed Internal Oscillator (HS IntOSC)		8 MHz (typ.)		
	PLL	PLL0 (for CPU, with SSCG)	Provided		
		PLL1 (for CPU/Peripheral)	Provided		
	Sub Oscillator (SubOSC)		32.768 kHz		
I/O port			144	174	246
A/D converter	ADCA0	Physical input channels	Total 34 ch (12 bit resolution: 16 ch + 10 bit resolution: 18 ch)		
		External multiplexer support for channel number extension	Provided		
		Channels with T&H	Provided		
	ADCA1	Physical input channels	Total 24 ch (12 bit resolution: 16 ch + 10 bit resolution: 8 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)	
		External multiplexer support for channel number extension	Not provided		
		Channels with T&H	Not provided		

Table 1A.1 Overview of Product

Product Name		RH850/F1KH-D8		
		176 Pins	233 Pins	324 Pins
Timer	Timer Array Unit D (TAUD)	1 unit (16 bit resolution timers × 16 channels /unit)		
	Timer Array Unit B (TAUB)	2 units (16 bit resolution timers × 16 channels /unit)		
	Timer Array Unit J (TAUJ)	4 units (32 bit resolution timers × 4 channels /unit)		
	Operating System Timer (OSTM)	10 units		
	Real-Time Clock (RTCA)	1 unit		
	Encoder Timer (ENCA)	1 unit		
	Window Watchdog Timer A (WDTA)	3 units		
Serial interfaces	Clocked Serial Interface G (CSIG)	5 channels		
	Clocked Serial Interface H (CSIH)	5 channels		
	CAN Interface (RS-CANFD)	8 channels		12 channels
	LIN/UART Interface (RLIN3)	8 channels		
	LIN Master Interface (RLIN2)	10 channels	12 channels	16 channels
	I^{2}C Interface (RIIC)	2 channels		
	Clock Extension Peripheral Interface (CXP1)	Not provided		
	Single Edge Nibble Transmission (RSENT)	2 channels		
	FlexRay Interface (FLXA)	2 channel (A ch, B ch)		
	Ethernet AVB (ETNB)	1 channel (MII)		2 channels (MII)
External Interrupts	Maskable	24		
	Non-maskable (NMI)	1		
Other functions	Clock Monitors (CLMA)	For PLL0, PLL1, HS IntOSC, MainOSC		
	Data CRC (DCRA)	4 channels		
	Low-Voltage Indicator (LVI)	Provided		
	Power-On Clear (POC)	Provided		
	Core Voltage Monitors (CVM)	Provided		
	Error Correction Coding (ECC)	For Code flash, Data flash, Local RAM, Retention RAM, Global RAM, CSIH, RS-CANFD, FLXA, ETNB		For Code flash, Data flash, Local RAM, Retention RAM, Global RAM, CSIH, RS-CANFD, FLXA, ETNB, MMCA
	Low Power Sampler (LPS)	Provided		
	PWM Output/Diagnostic (PWM-Diag)	72 channels	80 channels	96 channels
	Motor Control	1 unit		
	Key Return (KR)	8 channels		
	CLOCK OUTPUT (FOUT)	Provided		
	RESET OUTPUT ($\overline{\text{RESETOUT}}$)	Provided		
	Intelligent Cryptographic Unit Master D (ICUMD)	Provided		
	On-Chip debug (OCD)	Provided		
	Boundary Scan	Provided		

Table 1A.1 Overview of Product

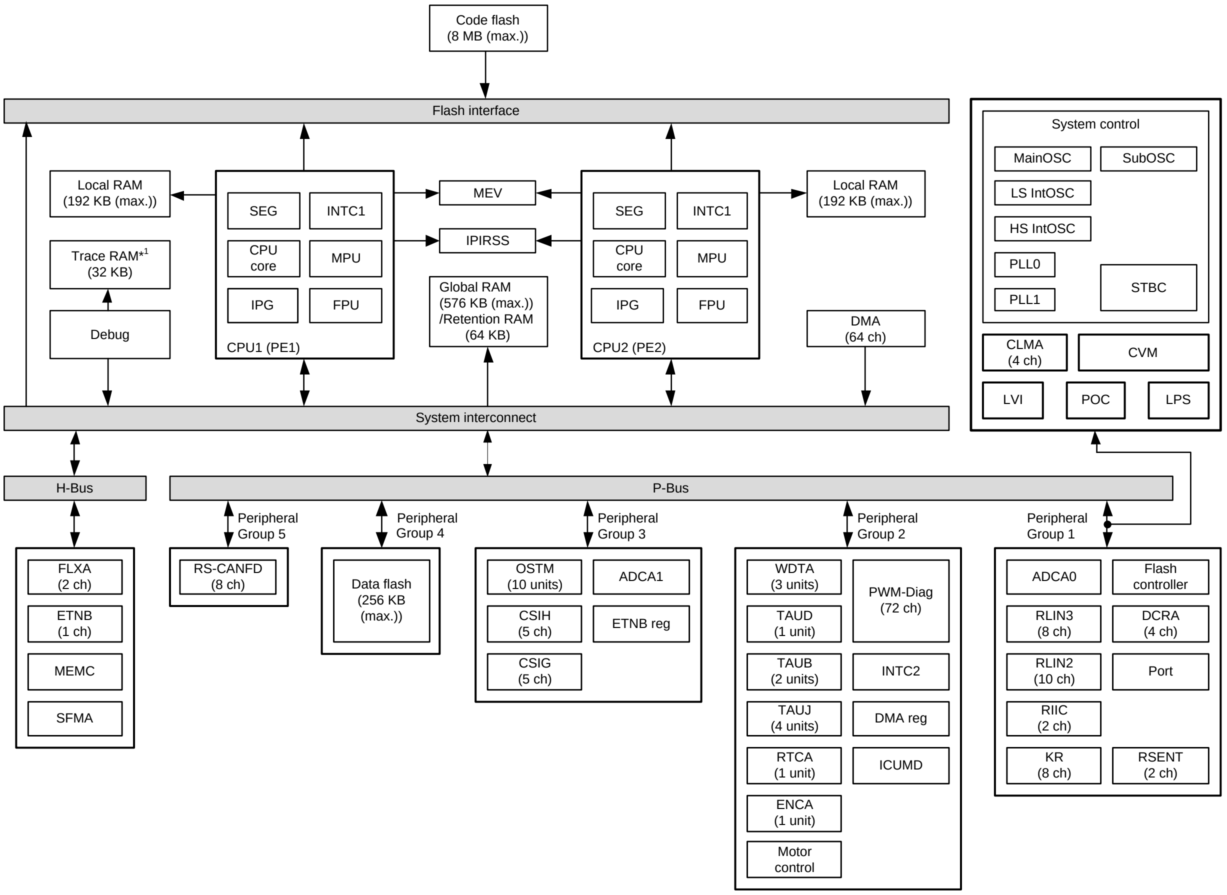
Product Name			RH850/F1KH-D8		
			176 Pins	233 Pins	324 Pins
Voltage supply	Internal supply	REG0VCC (for AWO)	VPOC to 5.5 V		
		REG1VCC (for ISO)	VPOC to 3.6 V		
	Input/output buffer supplies		VPOC to 5.5 V		
	A/D Converter supplies		3.0 to 5.5 V		
Package			176-pin LQFP	233-pin FPBGA	324-pin FPBGA

1A.3 RH850/F1KH Product Lineup

Table 1A.2 Product Lineup

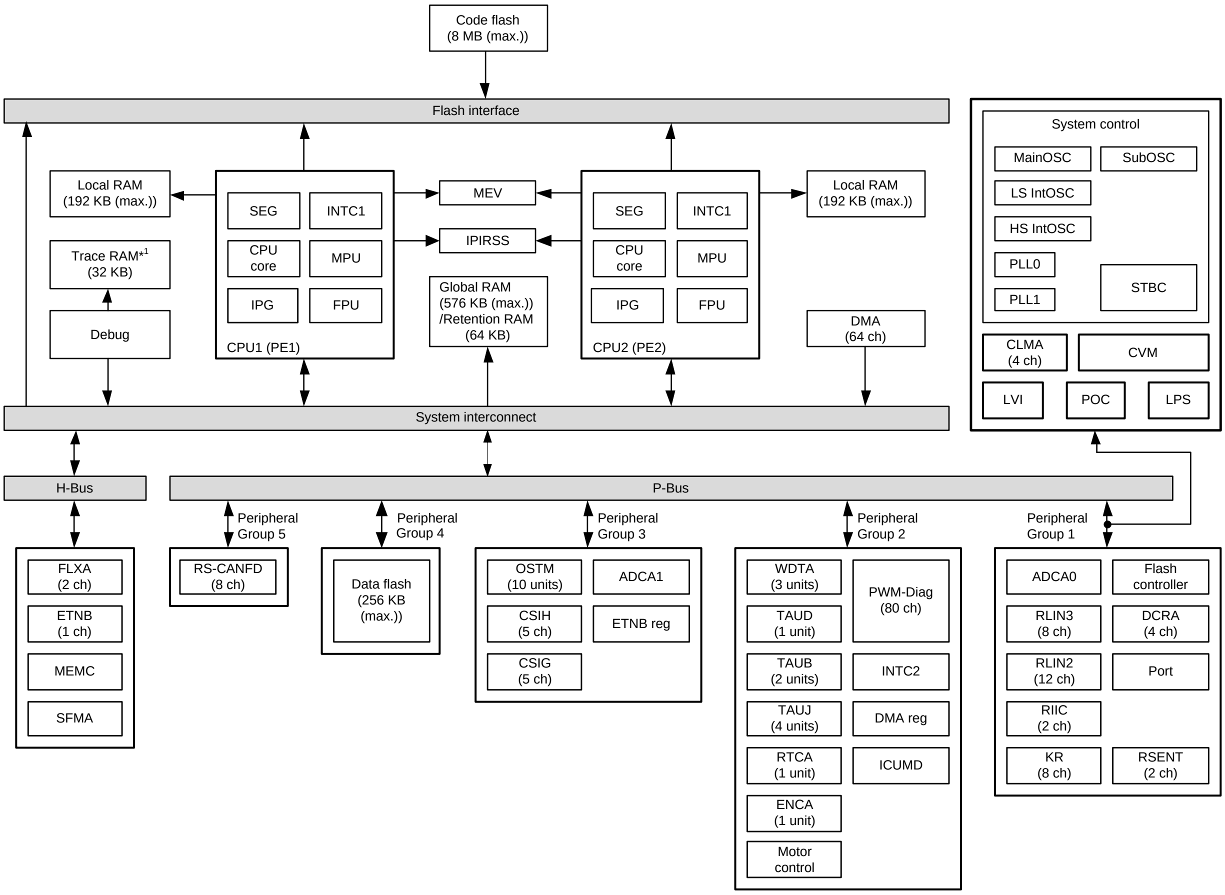
F1KH-D8		Memory							Part Name	
				Local RAM (LRAM)					Operating Temperature (Ta)	
Pin Count	CPU Frequency	Code Flash	Data Flash	CPU1	CPU2	Global RAM (GRAM)	Retention RAM (RRAM)	Trace RAM	–40°C to +105°C Package	–40°C to +125°C Package
176 pins	240 MHz max.	6 MB	256 KB	160 KB	160 KB	512 KB	64 KB	Not available	R7F7017083AFP-C LQFP	—
		8 MB		192 KB	192 KB	576 KB		32 KB	R7F7017093AFP-C LQFP	—
233 pins	240 MHz max.	6 MB	256 KB	160 KB	160 KB	512 KB	64 KB	Not available	R7F7017103ABG-C FPBGA	R7F7017104ABG-C FPBGA
		8 MB		192 KB	192 KB	576 KB		32 KB	R7F7017113ABG-C FPBGA	R7F7017114ABG-C FPBGA
324 pins	240 MHz max.	6 MB	256 KB	160 KB	160 KB	512 KB	64 KB	Not available	R7F7017143ABG-C FPBGA	R7F7017144ABG-C FPBGA
		8 MB		192 KB	192 KB	576 KB		32 KB	R7F7017153ABG-C FPBGA	R7F7017154ABG-C FPBGA

1A.4 RH850/F1KH Product Block Diagrams



Note 1. The trace RAM is only supported by products with 8 MB code flash memory.

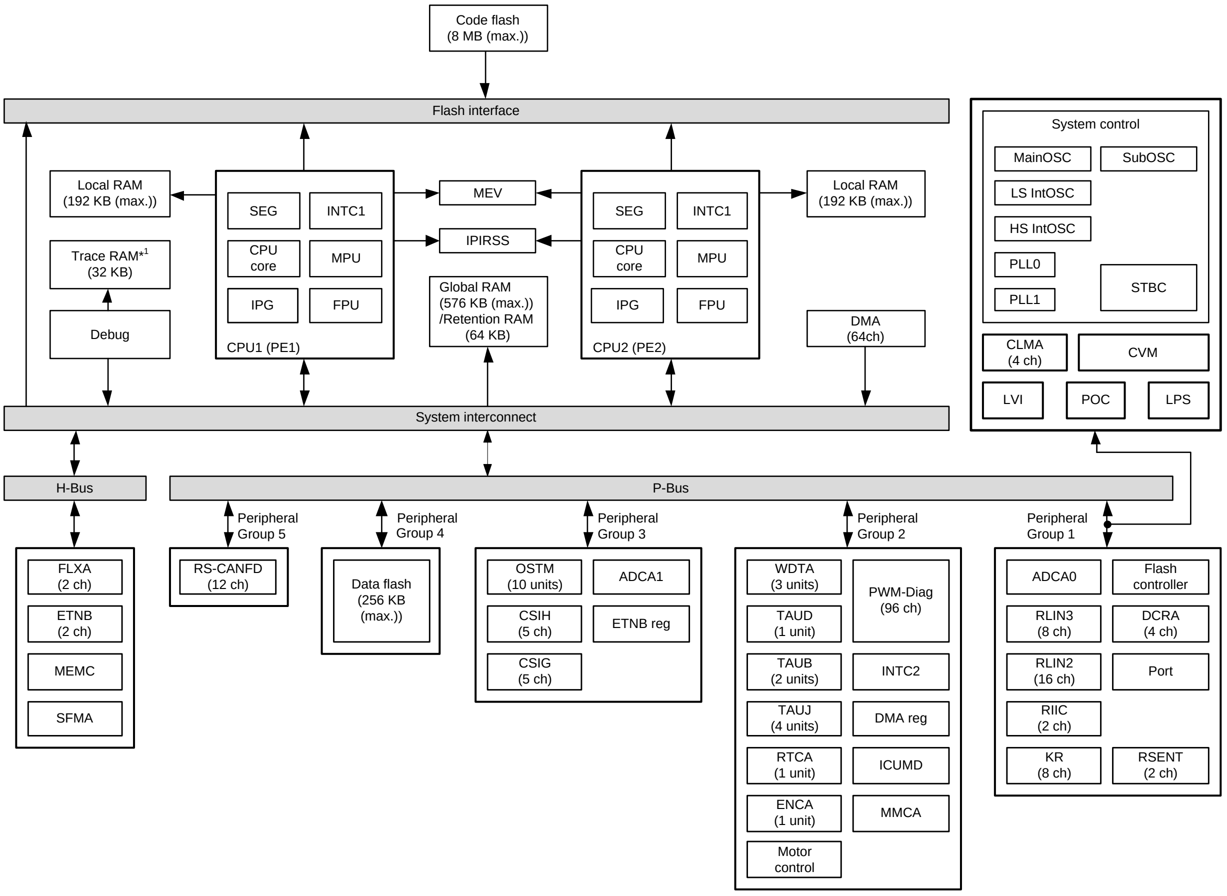
Figure 1A.1 Internal Block Diagram (RH850/F1KH-D8 176-Pin Version)



Note 1. The trace RAM is only supported by products with 8 MB code flash memory.

Figure 1A.2 Internal Block Diagram (RH850/F1KH-D8 233-Pin Version)

Figure 1A.3 Reserved



Note 1. The trace RAM is only supported by products with 8 MB code flash memory.

Figure 1A.4 Internal Block Diagram (RH850/F1KH-D8 324-Pin Version)

Section 1B Overview of RH850/F1KM-S4

1B.1 RH850/F1KM Product Features

The features of the RH850/F1KM are described below.

The RH850/F1KM is a 32-bit single-chip microcontroller with a G3KH CPU core. The key features of the F1KM are low power consumption, high computational processing power, and a wide variety of internal peripheral functions. To reduce supply current in a variety of applications, a wide range of power reducing measures are available. For example, there is a Low Power Sampler (LPS), that can poll signals input to the analog and digital input pins without CPU core interaction, and DeepSTOP mode in which the power supply to the most circuits of the microcontroller can be turned off.

Applications

The RH850/F1KM is ideal for automotive electronics, such as BCM (body control module), gateway, HVAC, lighting modules, and many other applications.

1B.2 RH850/F1KM Functions

Table 1B.1 Overview of Product

Product Name			RH850/F1KM-S4				
			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
Memory			See **Table 1B.2, Product Lineup**.				
External Memory Access Controller (MEMC)			Not provided		23 bit Address Bus		24 bit Address Bus
Serial Flash Memory I/F (SFMA)		Bus width	Not provided	4 bit			
		Mode		SDR			
		Max. clock		40 MHz			
CPU	CPU System		G3KH				
	CPU frequency		240 MHz max.				
	FPU		Single-precision				
	Protection Function	Memory Protection Unit (MPU)	Provided				
		Internal Peripheral-device Guard (IPG)	Provided				
		Processor Element Guard (PEG)	Provided				
DMA			32 channels				
Operating clock	Main Oscillator (MainOSC)		8/16/20/24 MHz				
	Low Speed Internal Oscillator (LS IntOSC)		240 kHz (typ.)				
	High Speed Internal Oscillator (HS IntOSC)		8 MHz (typ.)				
	PLL	PLL0 (for CPU, with SSCG)	Provided				
		PLL1 (for CPU/Peripheral)	Provided				
	Sub Oscillator (SubOSC)		Not provided	32.768 kHz			
I/O port			75	114	144	174	214
A/D converter	ADCA0	Physical input channels	Total 32 ch (12 bit resolution: 16 ch + 10 bit resolution: 16 ch)	Total 34 ch (12 bit resolution: 16 ch + 10 bit resolution: 18 ch)			
		External multiplexer support for channel number extension	Provided				
		Channels with T&H	Provided				

Table 1B.1 Overview of Product

Product Name			RH850/F1KM-S4				
			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
A/D converter	ADCA1	Physical input channels	Not provided	Total 12 ch (12 bit resolution: 8 ch + 10 bit resolution: 4 ch)	Total 24 ch (12 bit resolution: 16 ch + 10 bit resolution: 8 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)	
		External multiplexer support for channel number extension	Not provided				
		Channels with T&H	Not provided				
Timer	Timer Array Unit D (TAUD)		1 unit (16 bit resolution timers × 16 channels /unit)				
	Timer Array Unit B (TAUB)		1 unit (16 bit resolution timers × 16 channels /unit)		2 units (16 bit resolution timers × 16 channels /unit)		
	Timer Array Unit J (TAUJ)		4 units (32 bit resolution timers × 4 channels /unit)				
	Operating System Timer (OSTM)		5 units				
	Real-Time Clock (RTCA)		1 unit				
	Encoder Timer (ENCA)		1 unit				
	Window Watchdog Timer A (WDTA)		2 units				
Serial interfaces	Clocked Serial Interface G (CSIG)		1 channel	2 channels	4 channels		
	Clocked Serial Interface H (CSIH)		4 channels				
	CAN Interface (RS-CANFD)		8 channels				
	LIN/UART Interface (RLIN3)		3 channels	6 channels	8 channels		
	LIN Master Interface (RLIN2)		3 channels	6 channels	10 channels	12 channels	
	I^{2}C Bus Interface (RIIC)		2 channels				
	Clock Extension Peripheral Interface (CXP1)		Not provided				
	Single Edge Nibble Transmission (RSENT)		1 channel	2 channels			
	FlexRay Interface (FLXA)		2 channel (A ch, B ch)				
	Ethernet AVB (ETNB)		Not provided		1 channel (MII)		
External Interrupts	Maskable		14	24			
	Non-maskable (NMI)		1				
Other functions	Clock Monitors (CLMA)		For PLL0, PLL1, HS IntOSC, MainOSC				
	Data CRC (DCRA)		4 channels				
	Low-Voltage Indicator (LVI)		Provided				
	Power-On Clear (POC)		Provided				
	Core Voltage Monitors (CVM)		Provided				
	Error Correction Coding (ECC)		For Code flash, Data flash, Local RAM, Retention RAM, Global RAM, CSIH, RS-CANFD, FLXA		For Code flash, Data flash, Local RAM, Retention RAM, Global RAM, CSIH, RS-CANFD, FLXA, ETNB		

Table 1B.1 Overview of Product

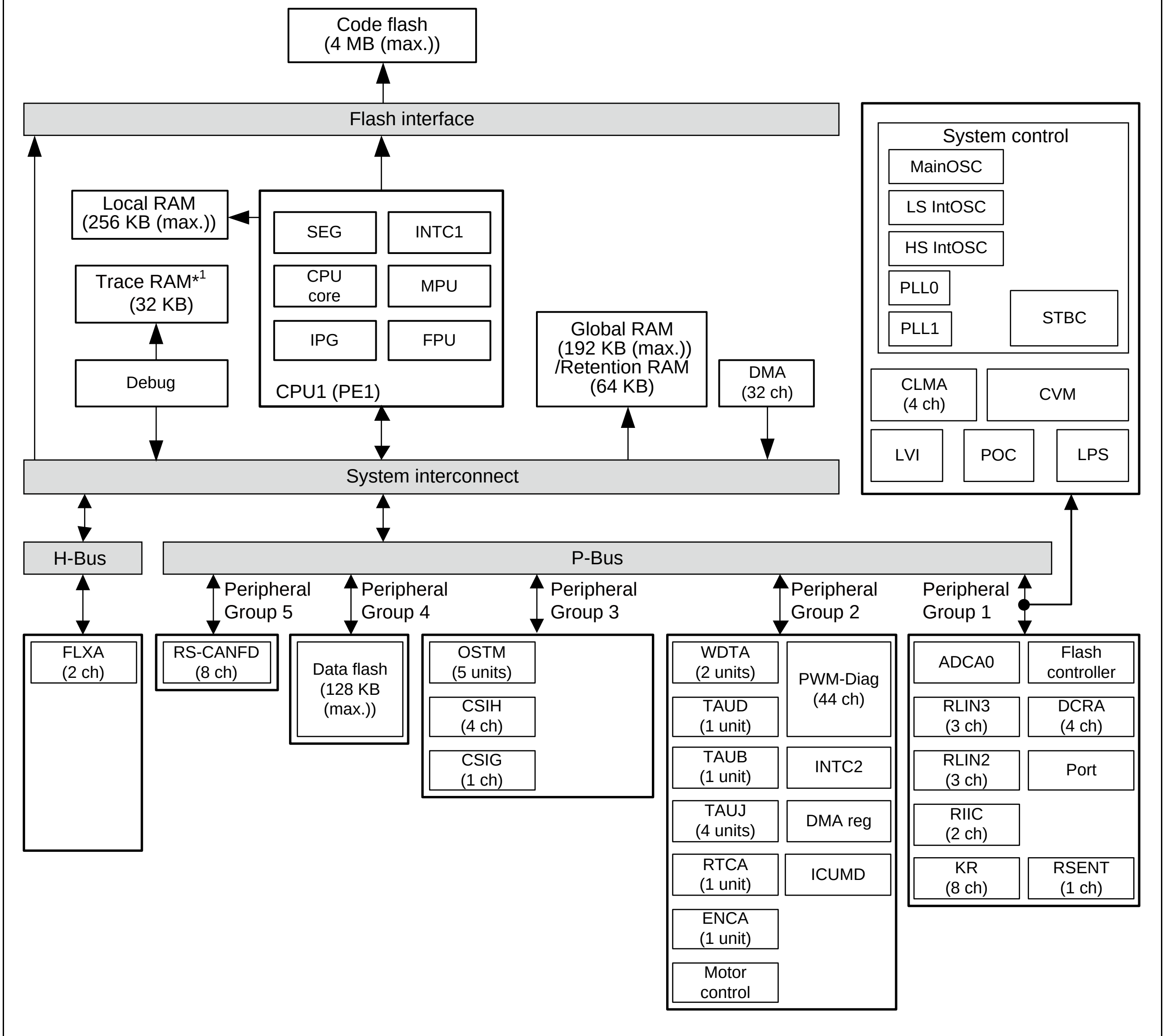
Product Name		RH850/F1KM-S4				
		100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
Other functions	Low Power Sampler (LPS)	Provided				
	PWM Output/Diagnostic (PWM-Diag)	44 channels	64 channels	72 channels	80 channels	96 channels
	Motor Control	1 unit				
	Key Return (KR)	8 channels				
	CLOCK OUTPUT (FOUT)	Provided				
	RESET OUTPUT ($\overline{\text{RESETOUT}}$)	Provided				
	Intelligent Cryptographic Unit Master D (ICUMD)	Provided				
	On-Chip debug (OCD)	Provided				
	Boundary Scan	Provided				
Voltage supply	Internal supply	VPOC to 5.5 V				
	Input/output buffer supplies	VPOC to 5.5 V				
	A/D Converter supplies	3.0 to 5.5 V				
Package		100-pin LQFP	144-pin LQFP	176-pin LQFP	233-pin FPBGA	272-pin FPBGA

1B.3 RH850/F1KM Product Lineup

Table 1B.2 Product Lineup

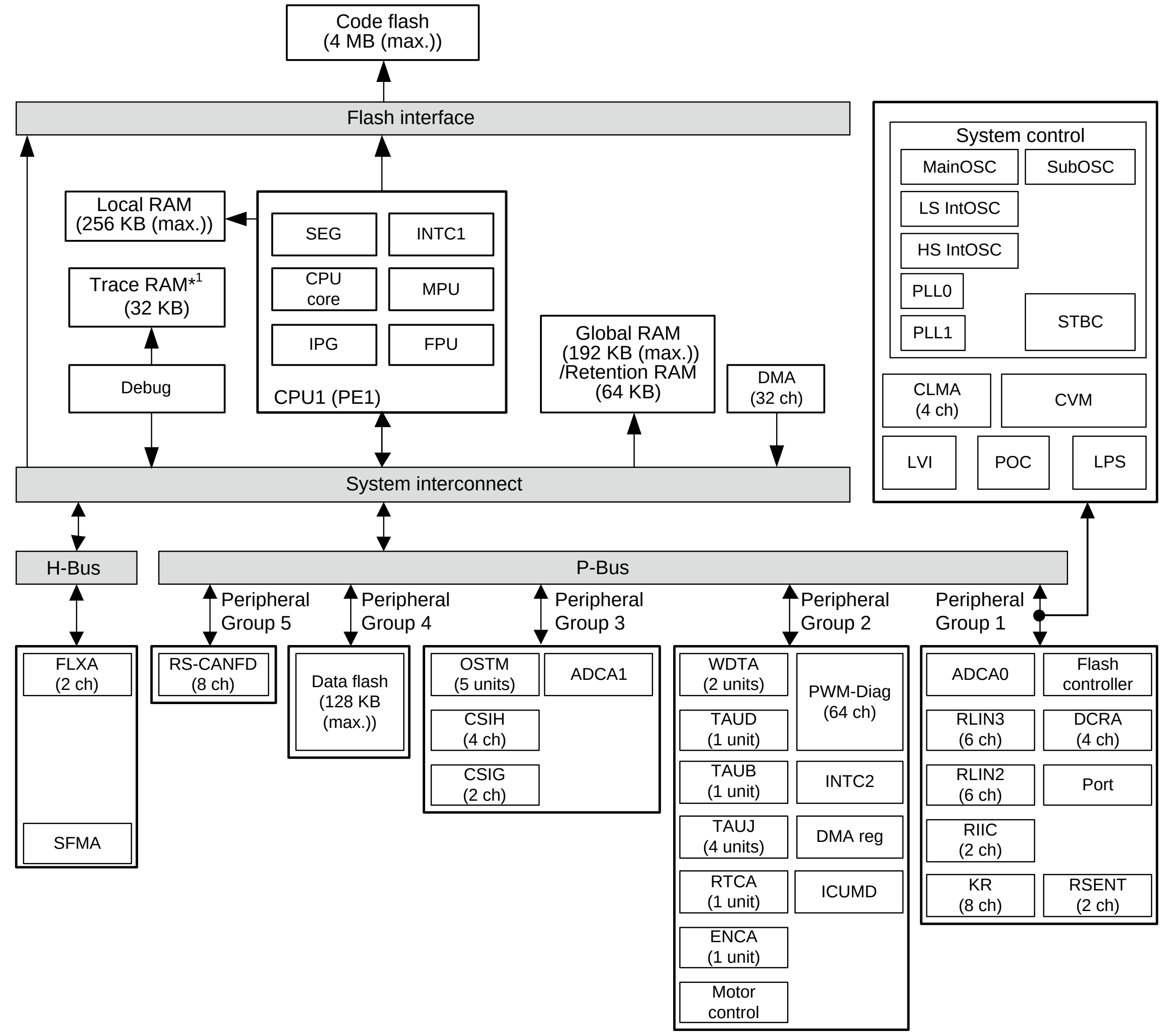
F1KM-S4		Memory						Part Name	
								Operating Temperature (Ta)	
Pin Count	CPU Frequency	Code Flash	Data Flash	Local RAM (LRAM)	Global RAM (GRAM)	Retention RAM (RRAM)	Trace RAM	–40°C to +105°C Package	–40°C to +125°C Package
100 pins	240 MHz max.	3 MB	128 KB	192 KB	128 KB	64 KB	Not available	R7F7016443AFP-C LQFP	—
		4 MB		256 KB	192KB		32 KB	R7F7016453AFP-C LQFP	—
144 pins	240 MHz max.	3 MB	128 KB	192 KB	128 KB	64 KB	Not available	R7F7016463AFP-C LQFP	—
		4 MB		256 KB	192KB		32 KB	R7F7016473AFP-C LQFP	—
176 pins	240 MHz max.	3 MB	128 KB	192 KB	128 KB	64 KB	Not available	R7F7016483AFP-C LQFP	—
		4 MB		256 KB	192KB		32 KB	R7F7016493AFP-C LQFP	—
233 pins	240 MHz max.	3 MB	128 KB	192 KB	128 KB	64 KB	Not available	R7F7016503ABG-C FPBGA	R7F7016504ABG-C FPBGA
		4 MB		256 KB	192KB		32 KB	R7F7016513ABG-C FPBGA	R7F7016514ABG-C FPBGA
272 pins	240 MHz max.	3 MB	128 KB	192 KB	128 KB	64 KB	Not available	R7F7016523ABG-C FPBGA	R7F7016524ABG-C FPBGA
		4 MB		256 KB	192KB		32 KB	R7F7016533ABG-C FPBGA	R7F7016534ABG-C FPBGA

1B.4 RH850/F1KM Product Block Diagrams



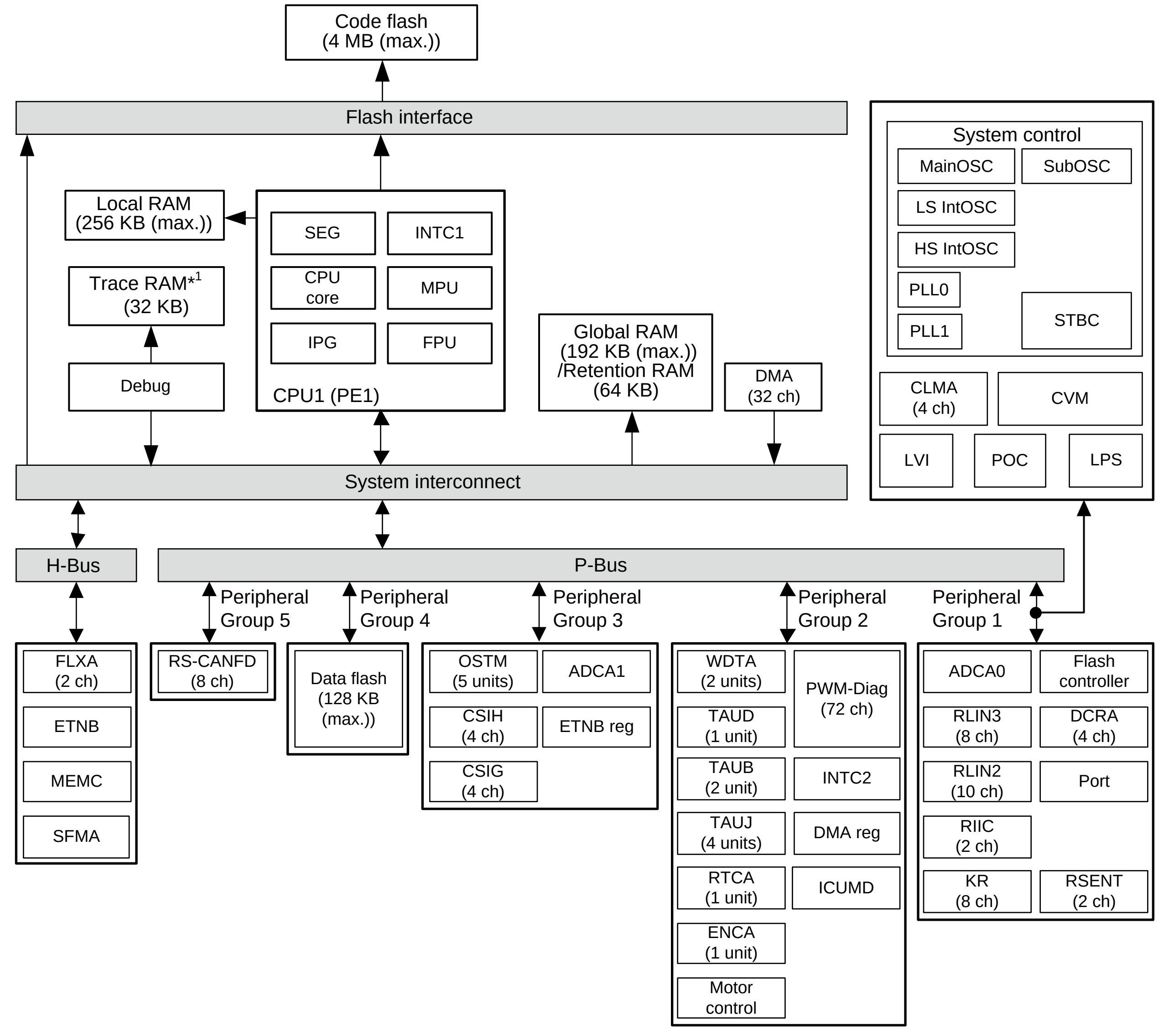
Note 1. The trace RAM is only supported by products with 4 MB code flash memory.

Figure 1B.1 Internal Block Diagram (RH850/F1KM-S4 100-Pin Version)



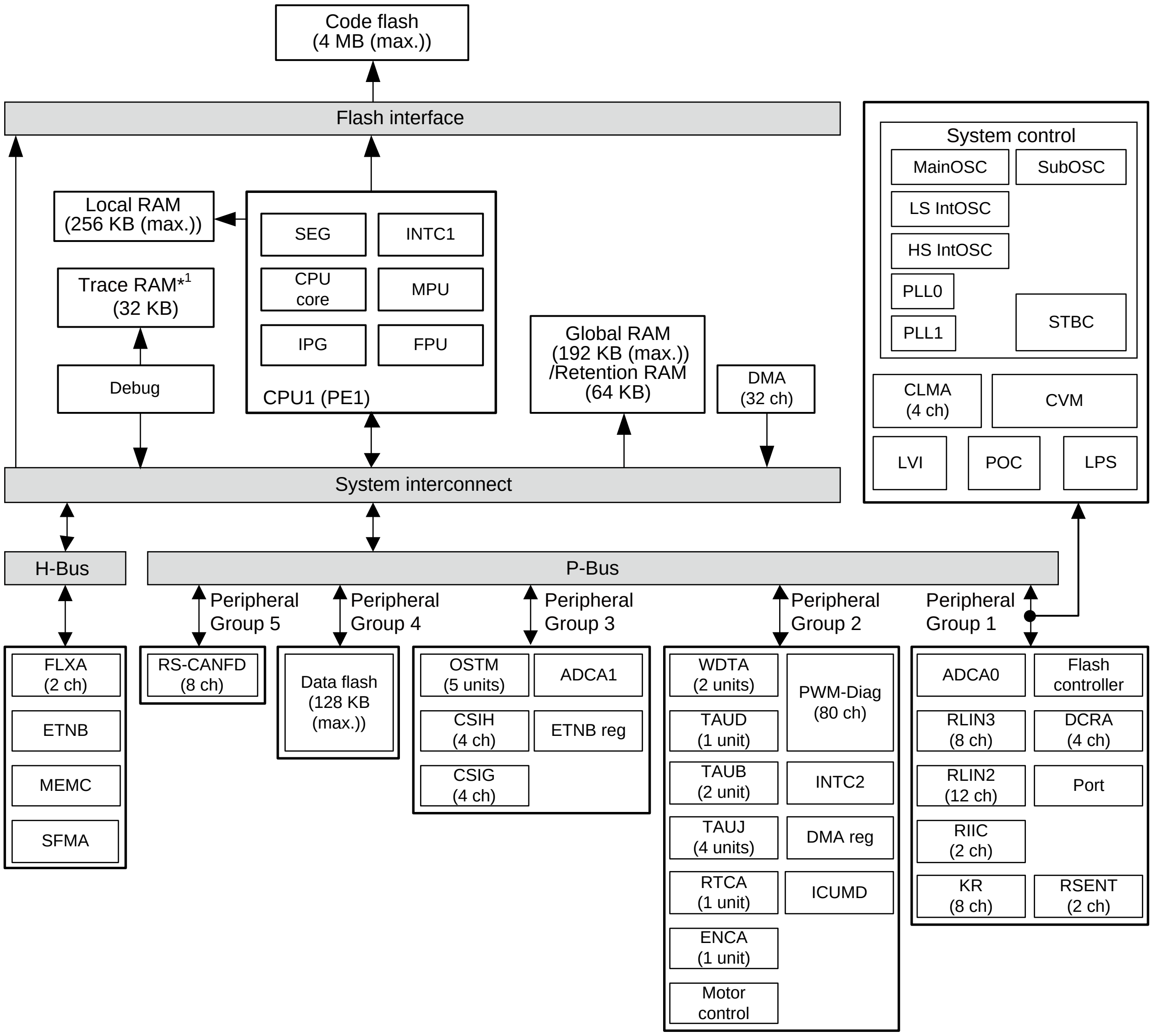
Note 1. The trace RAM is only supported by products with 4 MB code flash memory.

Figure 1B.2 Internal Block Diagram (RH850/F1KM-S4 144-Pin Version)



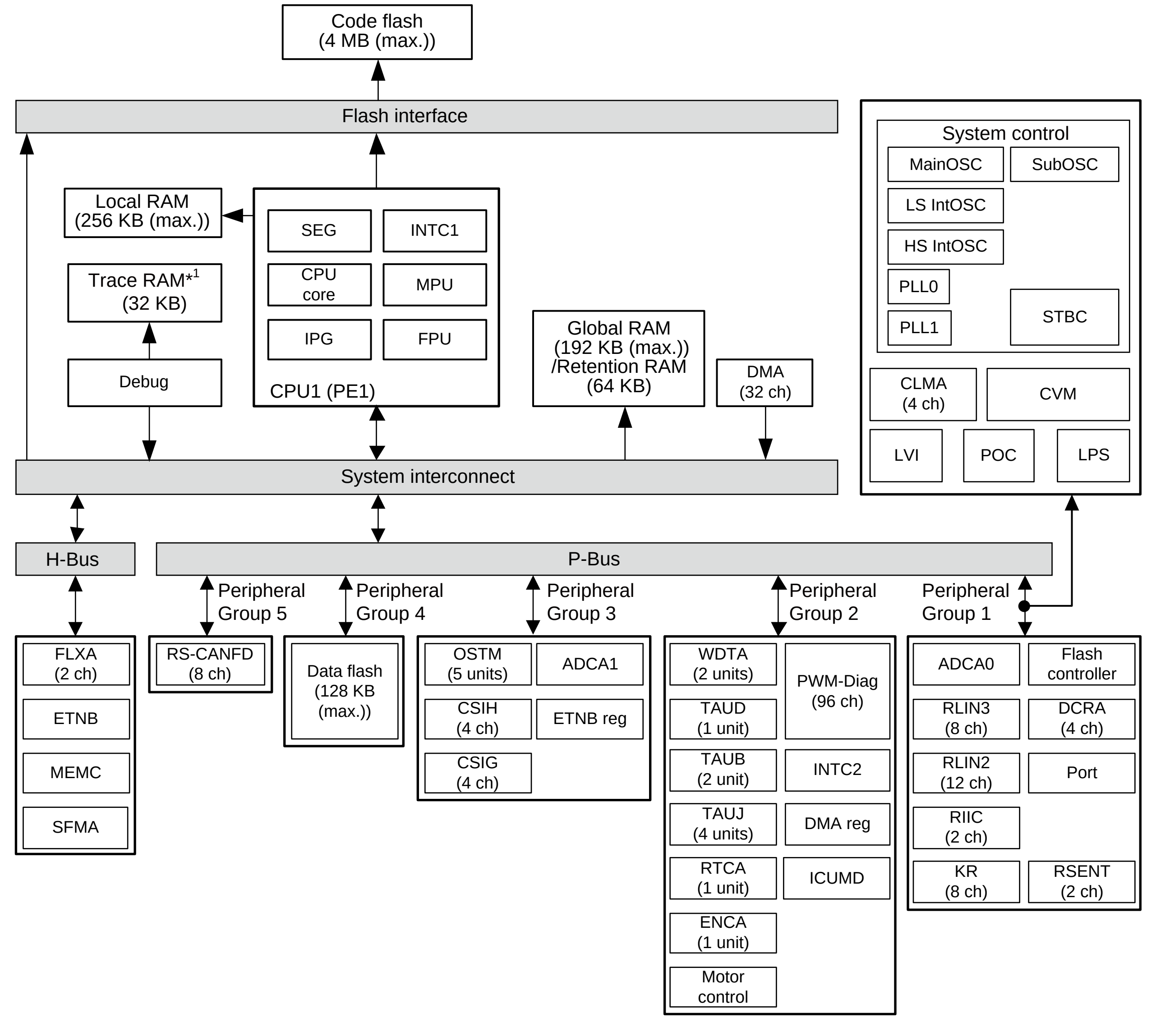
Note 1. The trace RAM is only supported by products with 4 MB code flash memory.

Figure 1B.3 Internal Block Diagram (RH850/F1KM-S4 176-Pin Version)



Note 1. The trace RAM is only supported by products with 4 MB code flash memory.

Figure 1B.4 Internal Block Diagram (RH850/F1KM-S4 233-Pin Version)



Note 1. The trace RAM is only supported by products with 4 MB code flash memory.

Figure 1B.5 Internal Block Diagram (RH850/F1KM-S4 272-Pin Version)

Section 1C Overview of RH850/F1KM-S1

1C.1 RH850/F1KM Product Features

The features of the RH850/F1KM are described below.

The RH850/F1KM is a 32-bit single-chip microcontroller with a G3KH CPU core. The key features of the F1KM are low power consumption, high computational processing power, and a wide variety of internal peripheral functions. To reduce supply current in a variety of applications, a wide range of power reducing measures are available. For example, there is a Low Power Sampler (LPS), that can poll signals input to the analog and digital input pins without CPU core interaction, and DeepSTOP mode, in which the power supply to the most circuits of the microcontroller can be turned off.

Applications

The RH850/F1KM is ideal for automotive electronics, such as BCM (body control module), gateway, HVAC, lighting modules, and many other applications.

1C.2 RH850/F1KM Functions

Table 1C.1 Overview of Product

Product Name			RH850/F1KM-S1			
			48 Pins	64 Pins	80 Pins	100 Pins
Memory			See **Table 1C.2, Product Lineup**.			
External Memory Access Controller (MEMC)			Not provided			
CPU	CPU System		G3KH			
	CPU frequency		120 MHz max			
	FPU		Single-Precision			
	Protection Function	Memory Protection Unit (MPU)	Provided			
		Internal Peripheral Guard (IPG)	Provided			
		Processor Element Guard (PEG)	Provided			
DMA			16 channels			
Operating clock	Main Oscillator (MainOSC)		8/16/20/24 MHz			
	Low Speed Internal Oscillator (LS IntOSC)		240 kHz(typ.)			
	High Speed Internal Oscillator (HS IntOSC)		8 MHz(typ.)			
	PLL	PLL0 (for CPU, with SSCG)	Not provided			
		PLL1 (for CPU/Peripheral)	Provided			
	Sub Oscillator (SubOSC)		Not provided			
I/O port			33	49	65	81
A/D converter	ADCA0	Physical input channels	Total 12 ch	Total 21 ch	Total 25 ch	Total 36 ch
			(12 bit resolution: 8 ch + 10 bit resolution: 4 ch)	(12 bit resolution: 10 ch + 10 bit resolution: 11 ch)	(12 bit resolution: 11 ch + 10 bit resolution: 14 ch)	(12 bit resolution: 16 ch + 10 bit resolution: 20 ch)
		External multiplexer support for channel number extension	Provided			
		Channels with T&H	3			6
	ADCA1	Physical input channels	Not provided			
		External multiplexer support for channel number extension	Not provided			
		Channels with T&H	Not provided			
Timer	Timer Array Unit D (TAUD)		1 unit (16 bit resolution timers × 16 channels /unit)			
	Timer Array Unit B (TAUB)		Not provided		1 unit (16 bit resolution timers × 16 channels /unit)	
	Timer Array Unit J (TAUJ)		4 units (32 bit resolution timers × 4 channels /unit)			
	Operating System Timer (OSTM)		1 unit			
	Real-Time Clock (RTCA)		1 unit			
	Encoder Timer (ENCA)		1 unit			
	Window Watchdog Timer A (WDTA)		2 units			

Table 1C.1 Overview of Product

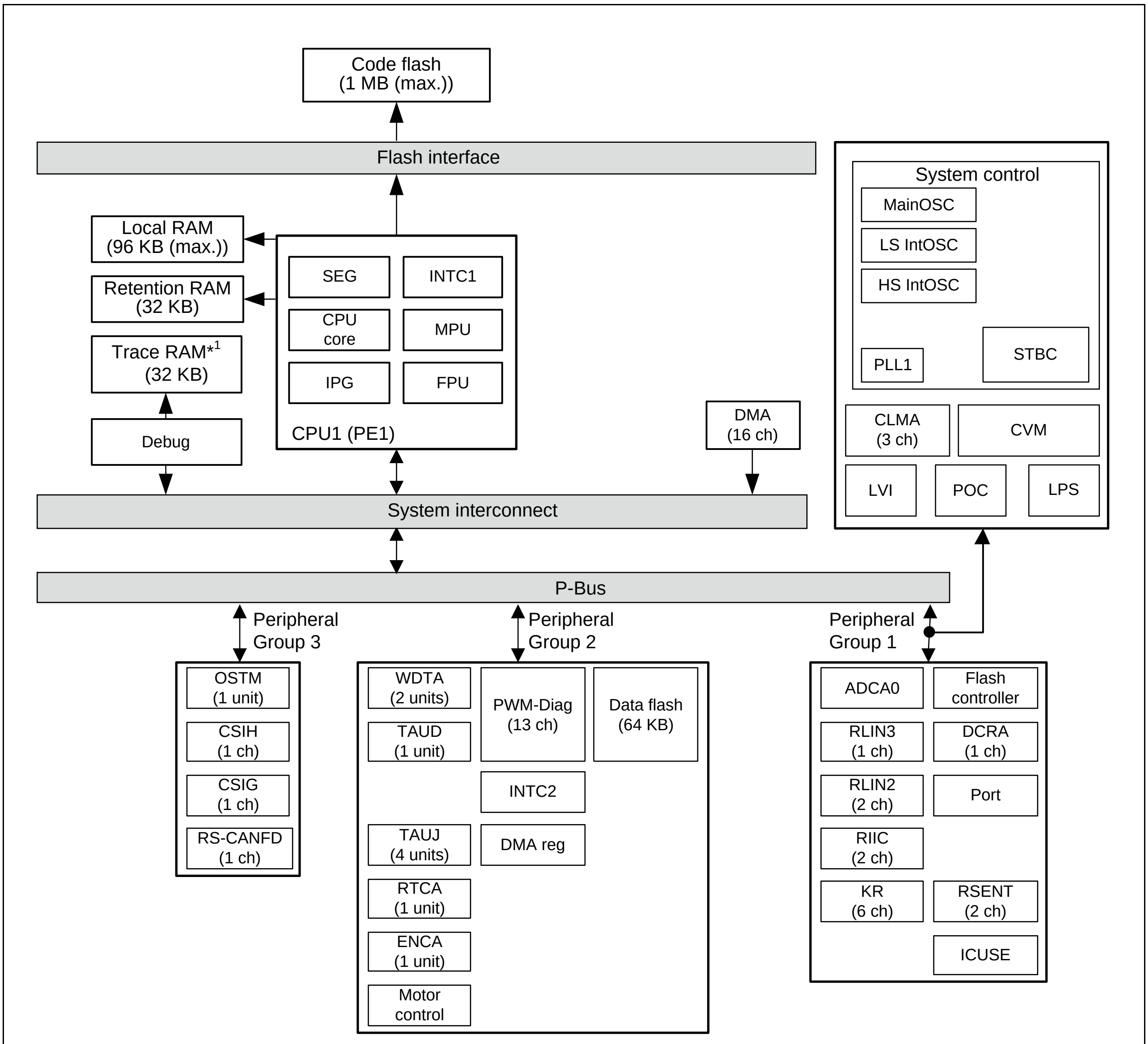
Product Name		RH850/F1KM-S1			
		48 Pins	64 Pins	80 Pins	100 Pins
Serial interfaces	Clocked Serial Interface G (CSIG)	1 channel			
	Clocked Serial Interface H (CSIH)	1 channel		3 channels	4 channels
	CAN Interface (RS-CANFD)	1 channel	3 channels		6 channels
	LIN/UART Interface (RLIN3)	1 channel	2 channels	3 channels	4 channels
	LIN Master Interface (RLIN2)	2 channels			3 channels
	I^{2}C Bus Interface (RIIC)	2 channels			
	Clock Extension Peripheral Interface (CXP1)	Not provided			
	Single Edge Nibble Transmission (RSENT)	2 channels			
External Interrupts	Maskable	8		12	13
	Non-maskable (NMI)	1			
Other functions	Clock Monitors (CLMA)	For PLL1, HS IntOSC, MainOSC			
	Data CRC (DCRA)	1 channel		4 channels	
	Low-Voltage Indicator (LVI)	Provided			
	Power-On Clear (POC)	Provided			
	Core Voltage Monitors (CVM)	Provided			
	Error Correction Coding (ECC)	For Code Flash, Data Flash, Local RAM, Retention RAM, CSIH, RS-CANFD			
	Low Power Sampler (LPS)	Provided			
	PWM Output/Diagnostic (PWM-Diag)	13 channels	24 channels		48 channels
	Motor Control	1 unit			
	Key Return (KR)	6 channels	8 channels		
	CLOCK OUTPUT (FOUT)	Provided			
	RESET OUTPUT ($\overline{\text{RESETOUT}}$)	Not Provided	Provided		
	Intelligent Cryptographic Unit E (ICUSE)	Provided			
	Secure WDT (SWDT)	Provided			
	On-Chip debug (OCD)	Provided			
	Boundary Scan	Provided			
Voltage supply	Internal supply	VPOC to 5.5 V			
	Input/output buffer supplies	VPOC to 5.5 V			
	A/D Converter supplies	3.0 V to 5.5 V			
Package		48-pin LQFP	64-pin LQFP	80-pin LQFP	100-pin LQFP

1C.3 RH850/F1KM Product Lineup

Table 1C.2 Product Lineup

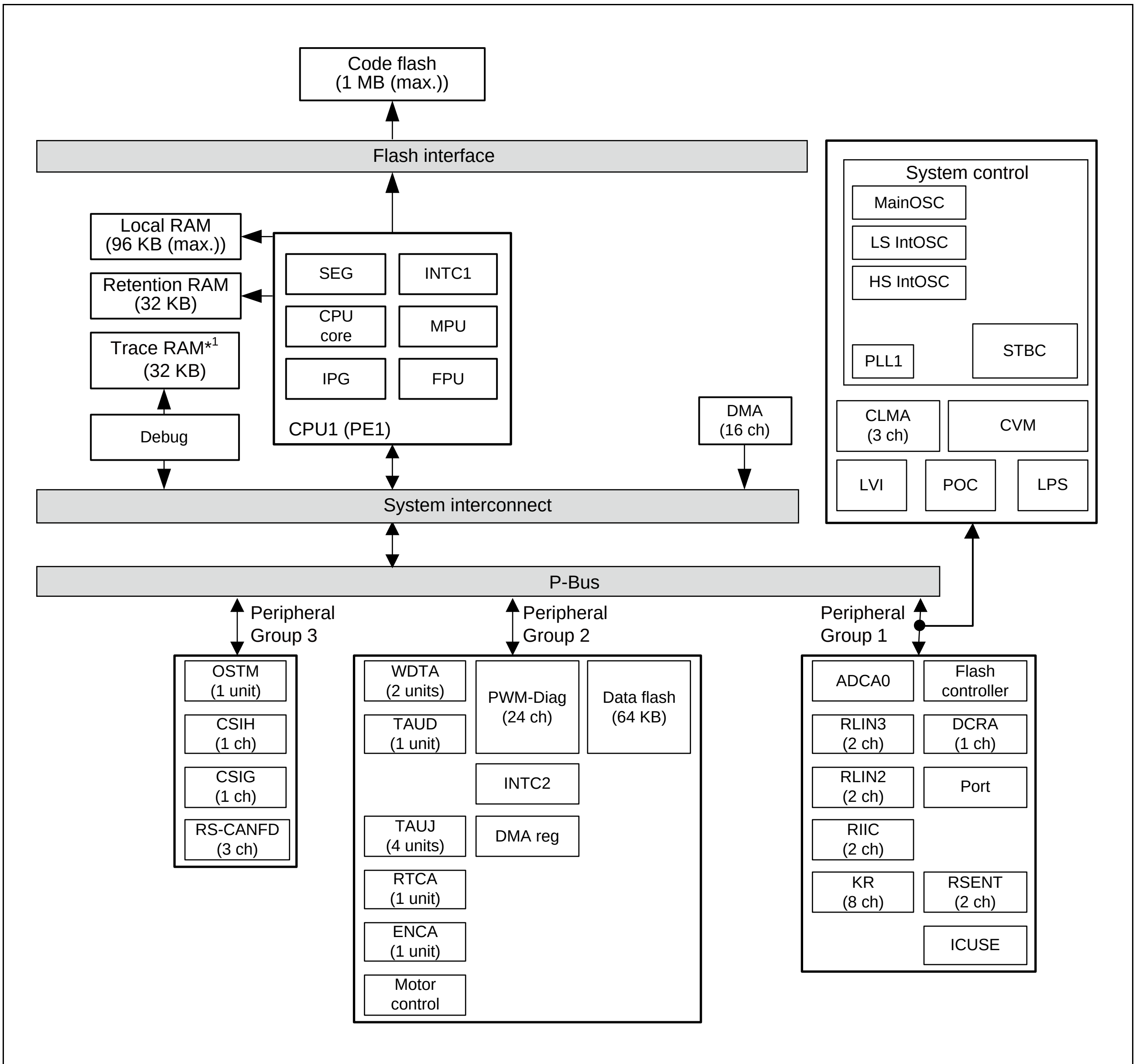
F1KM-S1		Memory					Part Name	
							Operating Temperature (Ta)	
Pin Count	CPU Frequency	Code Flash	Data Flash	Local RAM (LRAM)	Retention RAM (RRAM)	Trace RAM	–40°C to +105°C Package	–40°C to +125°C Package
100 pins	120 MHz max.	1024 KB	64 KB	96 KB	32 KB	32 KB	R7F7016843AFP-C LQFP	R7F7016844AFP-C LQFP
		768 KB		64 KB		Not available	R7F7016853AFP-C LQFP	R7F7016854AFP-C LQFP
		512 KB		32 KB		Not available	R7F7016863AFP-C LQFP	R7F7016864AFP-C LQFP
80 pins	120 MHz max.	1024 KB	64 KB	96 KB	32 KB	32 KB	R7F7016873AFP-C LQFP	R7F7016874AFP-C LQFP
		768 KB		64 KB		Not available	R7F7016883AFP-C LQFP	R7F7016884AFP-C LQFP
		512 KB		32 KB		Not available	R7F7016893AFP-C LQFP	R7F7016894AFP-C LQFP
64 pins	120 MHz max.	1024 KB	64 KB	96 KB	32 KB	32 KB	R7F7016903AFP-C LQFP	R7F7016904AFP-C LQFP
		768 KB		64 KB		Not available	R7F7016913AFP-C LQFP	R7F7016914AFP-C LQFP
		512 KB		32 KB		Not available	R7F7016923AFP-C LQFP	R7F7016924AFP-C LQFP
48 pins	120 MHz max.	1024 KB	64 KB	96 KB	32 KB	32 KB	R7F7016933AFP-C LQFP	R7F7016934AFP-C LQFP
		768 KB		64 KB		Not available	R7F7016943AFP-C LQFP	R7F7016944AFP-C LQFP
		512 KB		32 KB		Not available	R7F7016953AFP-C LQFP	R7F7016954AFP-C LQFP

1C.4 RH850/F1KM Product Block Diagrams



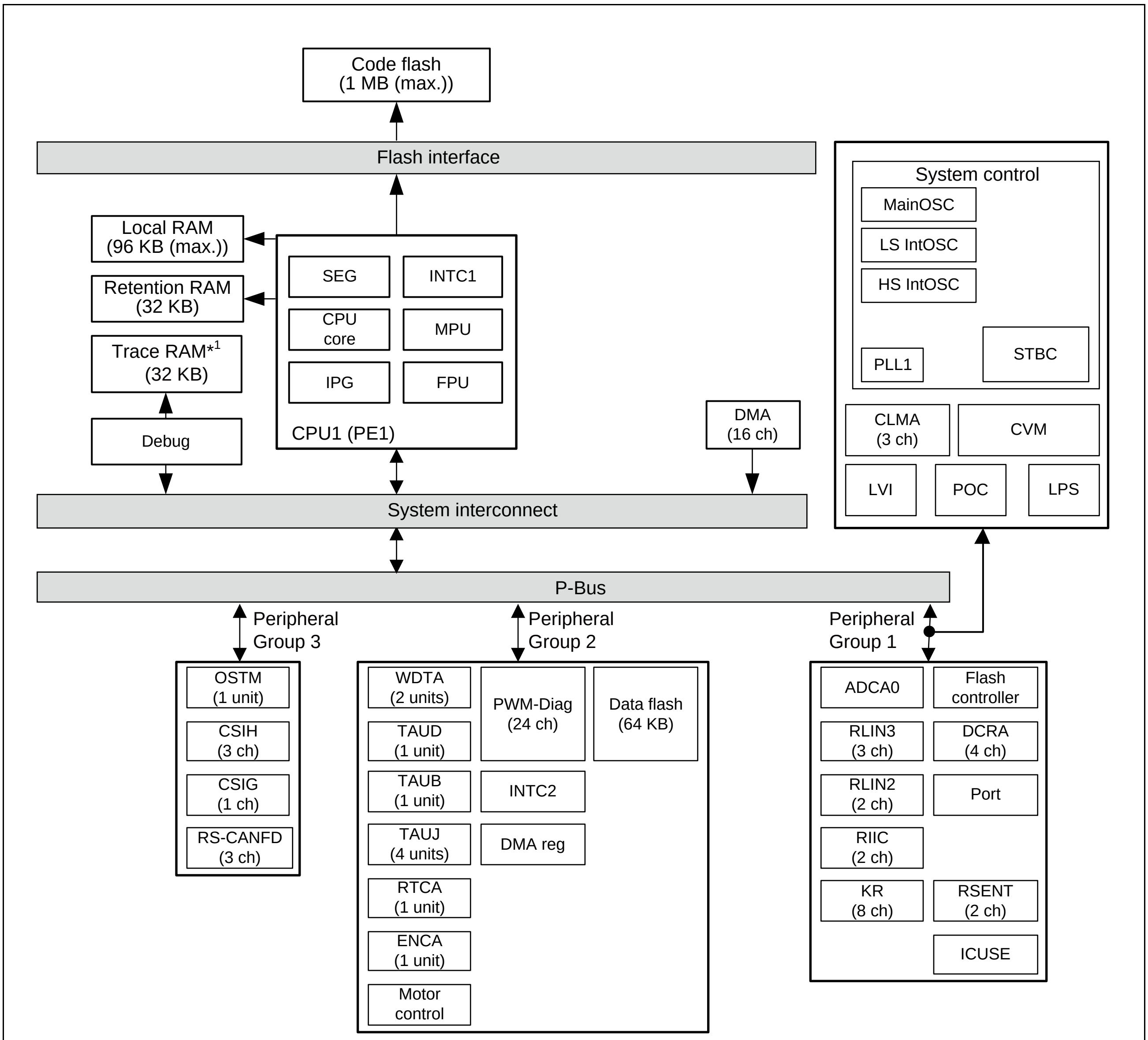
Note 1. The trace RAM is only supported by products with 1 MB code flash memory.

Figure 1C.1 Internal Block Diagram (RH850/F1KM-S1 48-Pin Version)



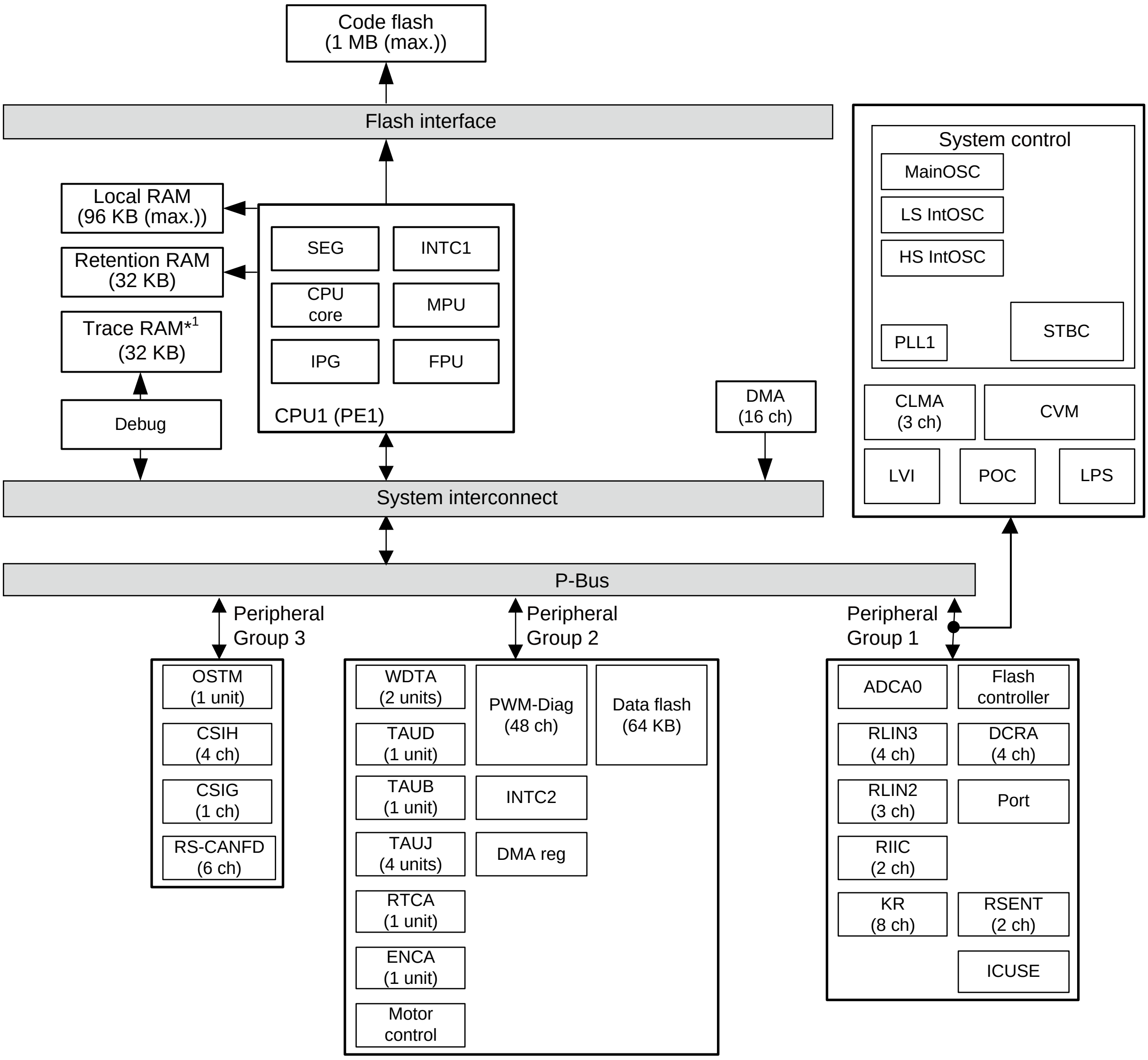
Note 1. The trace RAM is only supported by products with 1 MB code flash memory.

Figure 1C.2 Internal Block Diagram (RH850/F1KM-S1 64-Pin Version)



Note 1. The trace RAM is only supported by products with 1 MB code flash memory.

Figure 1C.3 Internal Block Diagram (RH850/F1KM-S1 80-Pin Version)



Note 1. The trace RAM is only supported by products with 1 MB code flash memory.

Figure 1C.4 Internal Block Diagram (RH850/F1KM-S1 100-Pin Version)

Section 2A Pin Function of RH850/F1KH-D8

This section describes the pin and port functions.

Section 2A.1, Pin Connection Diagram to **Section 2A.5, Recommended Connection of Unused Pins** describe the pin connections and respective pins.

Section 2A.6, Features of RH850/F1KH Port to **Section 2A.13, Description of Port Noise Filter & Edge/Level Detection** describe the general port functions.

2A.1 Pin Connection Diagram

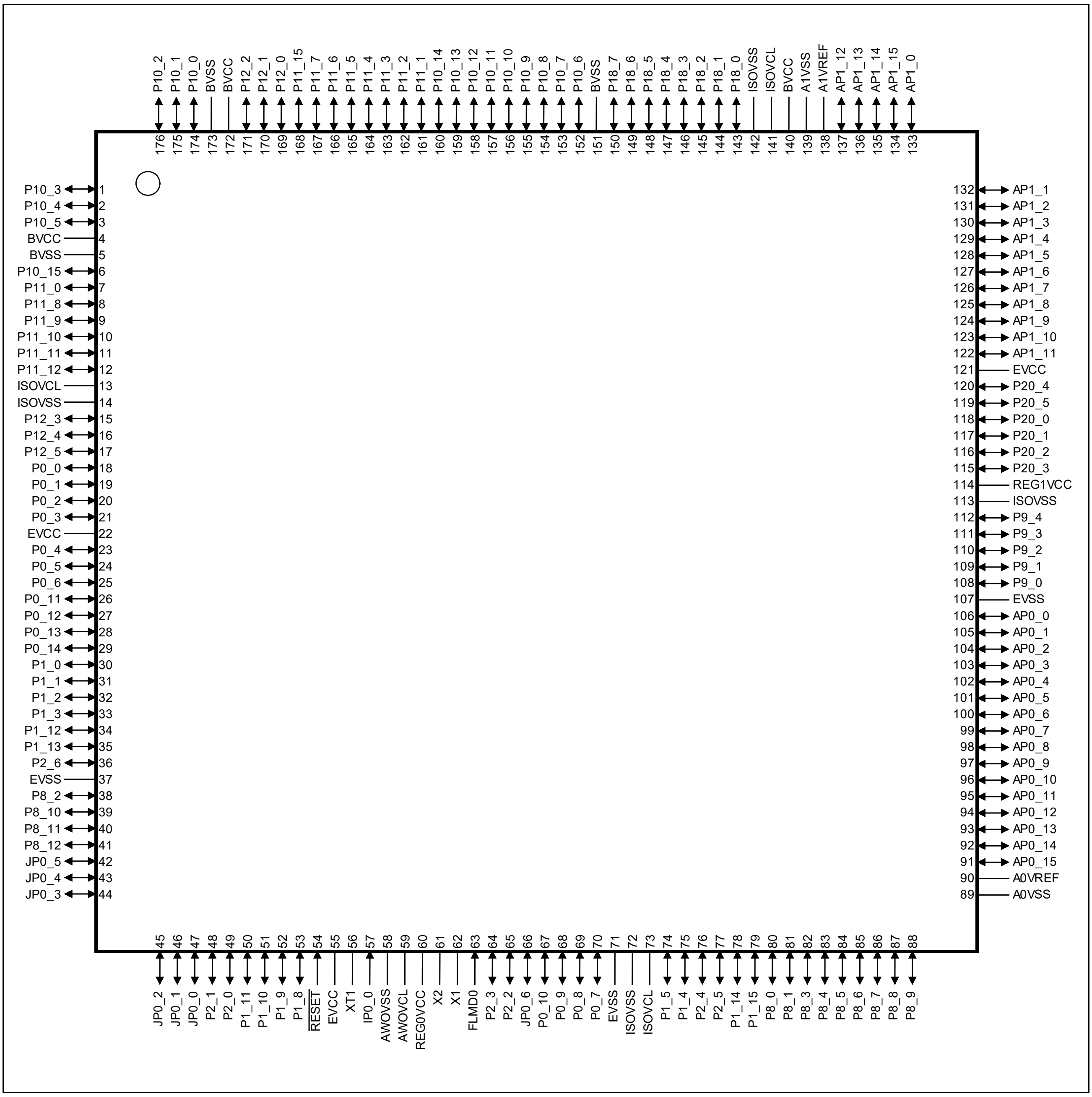
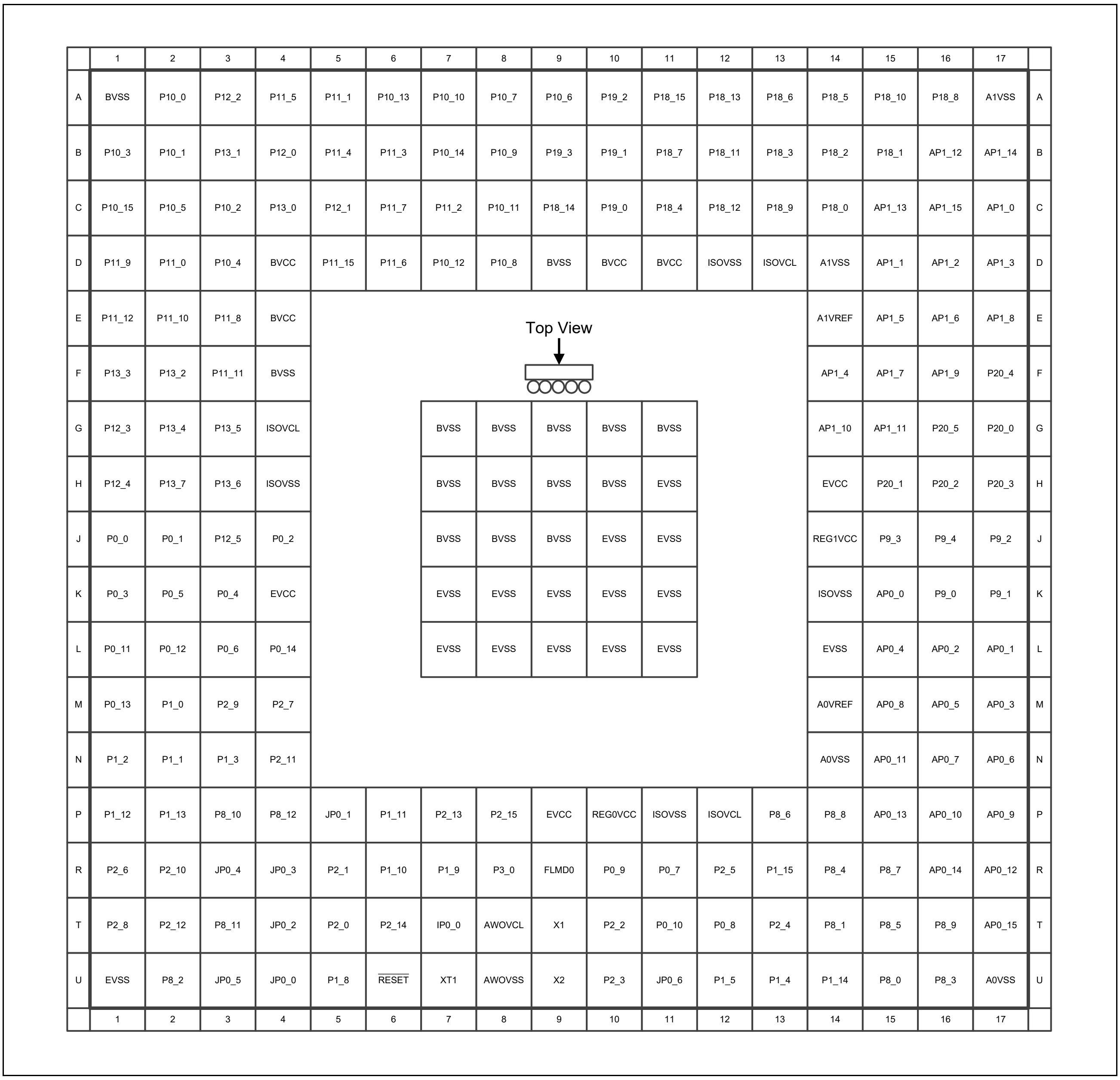


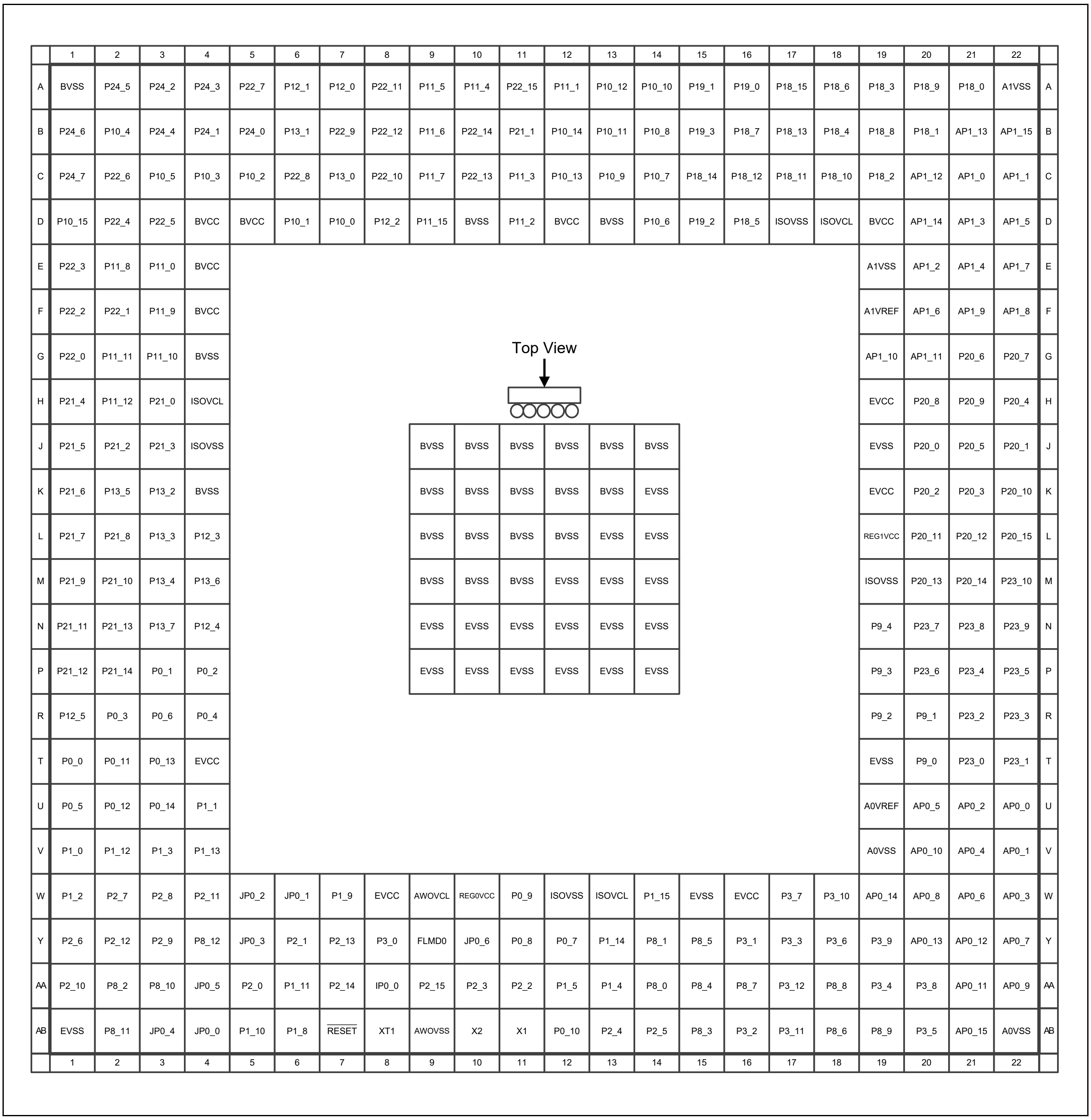
Figure 2A.1 Pin Connection Diagram (176-Pin LQFP)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	BVSS	P10_0	P12_2	P11_5	P11_1	P10_13	P10_10	P10_7	P10_6	P19_2	P18_15	P18_13	P18_6	P18_5	P18_10	P18_8	A1VSS	A
B	P10_3	P10_1	P13_1	P12_0	P11_4	P11_3	P10_14	P10_9	P19_3	P19_1	P18_7	P18_11	P18_3	P18_2	P18_1	AP1_12	AP1_14	B
C	P10_15	P10_5	P10_2	P13_0	P12_1	P11_7	P11_2	P10_11	P18_14	P19_0	P18_4	P18_12	P18_9	P18_0	AP1_13	AP1_15	AP1_0	C
D	P11_9	P11_0	P10_4	BVCC	P11_15	P11_6	P10_12	P10_8	BVSS	BVCC	BVCC	ISOVSS	ISOVCL	A1VSS	AP1_1	AP1_2	AP1_3	D
E	P11_12	P11_10	P11_8	BVCC										A1VREF	AP1_5	AP1_6	AP1_8	E
F	P13_3	P13_2	P11_11	BVSS										AP1_4	AP1_7	AP1_9	P20_4	F
G	P12_3	P13_4	P13_5	ISOVCL			BVSS	BVSS	BVSS	BVSS	BVSS			AP1_10	AP1_11	P20_5	P20_0	G
H	P12_4	P13_7	P13_6	ISOVSS			BVSS	BVSS	BVSS	BVSS	EVSS			EVCC	P20_1	P20_2	P20_3	H
J	P0_0	P0_1	P12_5	P0_2			BVSS	BVSS	BVSS	EVSS	EVSS			REG1VCC	P9_3	P9_4	P9_2	J
K	P0_3	P0_5	P0_4	EVCC			EVSS	EVSS	EVSS	EVSS	EVSS			ISOVSS	AP0_0	P9_0	P9_1	K
L	P0_11	P0_12	P0_6	P0_14			EVSS	EVSS	EVSS	EVSS	EVSS			EVSS	AP0_4	AP0_2	AP0_1	L
M	P0_13	P1_0	P2_9	P2_7										A0VREF	AP0_8	AP0_5	AP0_3	M
N	P1_2	P1_1	P1_3	P2_11										A0VSS	AP0_11	AP0_7	AP0_6	N
P	P1_12	P1_13	P8_10	P8_12	JP0_1	P1_11	P2_13	P2_15	EVCC	REG0VCC	ISOVSS	ISOVCL	P8_6	P8_8	AP0_13	AP0_10	AP0_9	P
R	P2_6	P2_10	JP0_4	JP0_3	P2_1	P1_10	P1_9	P3_0	FLMD0	P0_9	P0_7	P2_5	P1_15	P8_4	P8_7	AP0_14	AP0_12	R
T	P2_8	P2_12	P8_11	JP0_2	P2_0	P2_14	IP0_0	AWOVCL	X1	P2_2	P0_10	P0_8	P2_4	P8_1	P8_5	P8_9	AP0_15	T
U	EVSS	P8_2	JP0_5	JP0_0	P1_8	$\overline{\text{RESET}}$	XT1	AWOVSS	X2	P2_3	JP0_6	P1_5	P1_4	P1_14	P8_0	P8_3	A0VSS	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Figure 2A.2 Pin Connection Diagram (233-Pin FPBGA)

Figure 2A.3 Reserved



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
A	BVSS	P24_5	P24_2	P24_3	P22_7	P12_1	P12_0	P22_11	P11_5	P11_4	P22_15	P11_1	P10_12	P10_10	P19_1	P19_0	P18_15	P18_6	P18_3	P18_9	P18_0	A1VSS	A
B	P24_6	P10_4	P24_4	P24_1	P24_0	P13_1	P22_9	P22_12	P11_6	P22_14	P21_1	P10_14	P10_11	P10_8	P19_3	P18_7	P18_13	P18_4	P18_8	P18_1	AP1_13	AP1_15	B
C	P24_7	P22_6	P10_5	P10_3	P10_2	P22_8	P13_0	P22_10	P11_7	P22_13	P11_3	P10_13	P10_9	P10_7	P18_14	P18_12	P18_11	P18_10	P18_2	AP1_12	AP1_0	AP1_1	C
D	P10_15	P22_4	P22_5	BVCC	BVCC	P10_1	P10_0	P12_2	P11_15	BVSS	P11_2	BVCC	BVSS	P10_6	P19_2	P18_5	ISOVSS	ISOVCL	BVCC	AP1_14	AP1_3	AP1_5	D
E	P22_3	P11_8	P11_0	BVCC															A1VSS	AP1_2	AP1_4	AP1_7	E
F	P22_2	P22_1	P11_9	BVCC															A1VREF	AP1_6	AP1_9	AP1_8	F
G	P22_0	P11_11	P11_10	BVSS															AP1_10	AP1_11	P20_6	P20_7	G
H	P21_4	P11_12	P21_0	ISOVCL															EVCC	P20_8	P20_9	P20_4	H
J	P21_5	P21_2	P21_3	ISOVSS					BVSS	BVSS	BVSS	BVSS	BVSS	BVSS					EVSS	P20_0	P20_5	P20_1	J
K	P21_6	P13_5	P13_2	BVSS					BVSS	BVSS	BVSS	BVSS	BVSS	EVSS					EVCC	P20_2	P20_3	P20_10	K
L	P21_7	P21_8	P13_3	P12_3					BVSS	BVSS	BVSS	BVSS	EVSS	EVSS					REG1VCC	P20_11	P20_12	P20_15	L
M	P21_9	P21_10	P13_4	P13_6					BVSS	BVSS	BVSS	EVSS	EVSS	EVSS					ISOVSS	P20_13	P20_14	P23_10	M
N	P21_11	P21_13	P13_7	P12_4					EVSS	EVSS	EVSS	EVSS	EVSS	EVSS					P9_4	P23_7	P23_8	P23_9	N
P	P21_12	P21_14	P0_1	P0_2					EVSS	EVSS	EVSS	EVSS	EVSS	EVSS					P9_3	P23_6	P23_4	P23_5	P
R	P12_5	P0_3	P0_6	P0_4															P9_2	P9_1	P23_2	P23_3	R
T	P0_0	P0_11	P0_13	EVCC															EVSS	P9_0	P23_0	P23_1	T
U	P0_5	P0_12	P0_14	P1_1															A0VREF	AP0_5	AP0_2	AP0_0	U
V	P1_0	P1_12	P1_3	P1_13															A0VSS	AP0_10	AP0_4	AP0_1	V
W	P1_2	P2_7	P2_8	P2_11	JP0_2	JP0_1	P1_9	EVCC	AWOVCL	REG0VCC	P0_9	ISOVSS	ISOVCL	P1_15	EVSS	EVCC	P3_7	P3_10	AP0_14	AP0_8	AP0_6	AP0_3	W
Y	P2_6	P2_12	P2_9	P8_12	JP0_3	P2_1	P2_13	P3_0	FLMD0	JP0_6	P0_8	P0_7	P1_14	P8_1	P8_5	P3_1	P3_3	P3_6	P3_9	AP0_13	AP0_12	AP0_7	Y
AA	P2_10	P8_2	P8_10	JP0_5	P2_0	P1_11	P2_14	IP0_0	P2_15	P2_3	P2_2	P1_5	P1_4	P8_0	P8_4	P8_7	P3_12	P8_8	P3_4	P3_8	AP0_11	AP0_9	AA
AB	EVSS	P8_11	JP0_4	JP0_0	P1_10	P1_8	$\overline{\text{RESET}}$	XT1	AWOVSS	X2	X1	P0_10	P2_4	P2_5	P8_3	P3_2	P3_11	P8_6	P8_9	P3_5	AP0_15	A0VSS	AB
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	

Figure 2A.4 Pin Connection Diagram (324-Pin FPBGA)

Table 2A.1 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK / RLIN37RX / INTP17
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O
4	BVCC
5	BVSS
6	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
7	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
8	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
9	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
10	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
11	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
12	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
13	ISOVCL
14	ISOVSS
15	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
16	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
17	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
18	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
19	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
20	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
21	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
22	EVCC
23	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
24	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
25	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
26	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
27	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
28	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
29	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
30	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0 / $\overline{\text{CSIG4SSI}}$
31	P1_1 / INTP18 / RLIN33TX / CSIG4SC / TAUJ2I1 / TAUJ2O1
32	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2 / CSIG4SI
33	P1_3 / INTP19 / CAN3TX / DPIN23 / CSIG4SO / TAUJ2I3 / TAUJ2O3
34	P1_12 / CAN4RX / INTP4 / RLIN36TX
35	P1_13 / CAN4TX / RLIN36RX / INTP16
36	P2_6 / ADCA0SEL2 / CSIG4RYI / CSIG4RYO
37	EVSS
38	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
39	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
40	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
41	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
42	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT

Table 2A.1 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
43	JP0_4 / $\overline{\text{DCUTRST}}$
44	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
45	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
46	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
47	JP0_0 / INTP0 / TAUJ2I0 / TAUJ2O0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO
48	P2_1 / RLIN27TX / CAN6TX
49	P2_0 / RLIN27RX / CAN6RX / INTP6
50	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
51	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
52	P1_9 / DPIN20 / INTP21
53	P1_8
54	$\overline{\text{RESET}}$
55	EVCC
56	XT1
57	IP0_0 / XT2
58	AWOVSS
59	AWOVCL
60	REG0VCC
61	X2
62	X1
63	FLMD0
64	P2_3 / RLIN28TX / CSIH4CSS1
65	P2_2 / RLIN28RX / CSIH4CSS0
66	JP0_6 / $\overline{\text{EVTO}}$
67	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
68	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
69	P0_8 / INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
70	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
71	EVSS
72	ISOVSS
73	ISOVCL
74	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20 / CSIH4SC
75	P1_4 / RLIN35RX / INTP15 / DPIN18 / CSIH4SI
76	P2_4 / RLIN29RX / ADCA0SEL0 / CSIH4SO
77	P2_5 / RLIN29TX / $\overline{\text{CSIH4SSI}}$ / ADCA0SEL1
78	P1_14 / RLIN23RX / CAN7RX / INTP9 / CSIH4RYI / CSIH4RYO
79	P1_15 / RLIN23TX / CAN7TX
80	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / RIIC1SDA / SENT0RX / ADCA0I0S
81	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / RIIC1SCL / SENT0SPCO / ADCA0I1S
82	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
83	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
84	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
85	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
86	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S

Table 2A.1 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
87	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
88	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
89	A0VSS
90	A0VREF
91	AP0_15 / ADCA0I15
92	AP0_14 / ADCA0I14
93	AP0_13 / ADCA0I13
94	AP0_12 / ADCA0I12
95	AP0_11 / ADCA0I11
96	AP0_10 / ADCA0I10
97	AP0_9 / ADCA0I9
98	AP0_8 / ADCA0I8
99	AP0_7 / ADCA0I7
100	AP0_6 / ADCA0I6
101	AP0_5 / ADCA0I5
102	AP0_4 / ADCA0I4
103	AP0_3 / ADCA0I3
104	AP0_2 / ADCA0I2
105	AP0_1 / ADCA0I1
106	AP0_0 / ADCA0I0
107	EVSS
108	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA / ADCA0I2S
109	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL / ADCA0I3S
110	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
111	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S
112	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
113	ISOVSS
114	REG1VCC
115	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
116	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
117	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
118	P20_0 / RLIN26RX / PWGA64O / CAN6RX / INTP6 / CSIG3SI
119	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
120	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
121	EVCC
122	AP1_11 / ADCA1I11
123	AP1_10 / ADCA1I10
124	AP1_9 / ADCA1I9
125	AP1_8 / ADCA1I8
126	AP1_7 / ADCA1I7
127	AP1_6 / ADCA1I6
128	AP1_5 / ADCA1I5
129	AP1_4 / ADCA1I4
130	AP1_3 / ADCA1I3

Table 2A.1 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
131	AP1_2 / ADCA1I2
132	AP1_1 / ADCA1I1
133	AP1_0 / ADCA1I0
134	AP1_15 / ADCA1I15
135	AP1_14 / ADCA1I14
136	AP1_13 / ADCA1I13
137	AP1_12 / ADCA1I12
138	A1VREF
139	A1VSS
140	BVCC
141	ISOVCL
142	ISOVSS
143	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / TAUJ3I0 / TAUJ3O0 / ADCA1I0S
144	P18_1 / PWGA62O / ETNB0TXD0 / TAUJ3I1 / TAUJ3O1 / ADCA1I1S
145	P18_2 / PWGA63O / ETNB0TXD1 / TAUJ3I2 / TAUJ3O2 / ADCA1I2S
146	P18_3 / PWGA71O / ETNB0TXD2 / TAUJ3I3 / TAUJ3O3 / ADCA1I3S
147	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
148	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
149	P18_6 / ADCA1I6S
150	P18_7 / ETNB0TXCLK / ADCA1I7S
151	BVSS
152	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0 / RLIN24RX / MODE2
153	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
154	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / TAUJ3I2 / TAUJ3O2 / FLMD1
155	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
156	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
157	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
158	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
159	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
160	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
161	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
162	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3
163	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
164	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
165	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
166	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL

Table 2A.1 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
167	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
168	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$ / ETNB0RXERR / RLIN36TX
169	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
170	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
171	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
172	BVCC
173	BVSS
174	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1O3
175	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / TAUJ3O0 / MODE0
176	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
A1	BVSS
A2	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1O3
A3	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
A4	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
A5	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
A6	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
A7	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
A8	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
A9	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0 / RLIN24RX / MODE2
A10	P19_2 / ADCA1I18S
A11	P18_15 / ADCA1I15S
A12	P18_13 / ADCA1I13S
A13	P18_6 / ADCA1I6S
A14	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
A15	P18_10 / ADCA1I10S
A16	P18_8 / ADCA1I8S
A17	A1VSS
B1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK / RLIN37RX / INTP17
B2	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / TAUJ3O0 / MODE0
B3	P13_1 / MEMC0A20
B4	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
B5	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
B6	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
B7	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
B8	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
B9	P19_3 / ADCA1I19S
B10	P19_1 / ADCA1I17S
B11	P18_7 / ETNB0TXCLK / ADCA1I7S
B12	P18_11 / ADCA1I11S
B13	P18_3 / PWGA71O / ETNB0TXD2 / TAUJ3I3 / TAUJ3O3 / ADCA1I3S
B14	P18_2 / PWGA63O / ETNB0TXD1 / TAUJ3I2 / TAUJ3O2 / ADCA1I2S
B15	P18_1 / PWGA62O / ETNB0TXD0 / TAUJ3I1 / TAUJ3O1 / ADCA1I1S
B16	AP1_12 / ADCA1I12
B17	AP1_14 / ADCA1I14
C1	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
C2	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
C3	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1
C4	P13_0 / MEMC0A19
C5	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
C6	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
C7	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3
C8	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
C9	P18_14 / ADCA1I14S
C10	P19_0 / ADCA1I16S
C11	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
C12	P18_12 / ADCA1I12S
C13	P18_9 / ADCA1I9S
C14	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / TAUJ3I0 / TAUJ3O0 / ADCA1I0S
C15	AP1_13 / ADCA1I13
C16	AP1_15 / ADCA1I15
C17	AP1_0 / ADCA1I0
D1	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
D2	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
D3	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
D4	BVCC
D5	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$ / ETNB0RXERR / RLIN36TX
D6	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL
D7	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
D8	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / TAUJ3I2 / TAUJ3O2 / FLMD1
D9	BVSS
D10	BVCC
D11	BVCC
D12	ISOVSS
D13	ISOVCL
D14	A1VSS
D15	AP1_1 / ADCA1I1
D16	AP1_2 / ADCA1I2
D17	AP1_3 / ADCA1I3
E1	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
E2	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
E3	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
E4	BVCC
E14	A1VREF
E15	AP1_5 / ADCA1I5
E16	AP1_6 / ADCA1I6
E17	AP1_8 / ADCA1I8

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
F1	P13_3 / ETNB0RXERR
F2	P13_2 / ETNB0RXDV
F3	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
F4	BVSS
F14	AP1_4 / ADCA1I4
F15	AP1_7 / ADCA1I7
F16	AP1_9 / ADCA1I9
F17	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
G1	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
G2	P13_4
G3	P13_5 / MEMC0A21
G4	ISOVCL
G7	BVSS
G8	BVSS
G9	BVSS
G10	BVSS
G11	BVSS
G14	AP1_10 / ADCA1I10
G15	AP1_11 / ADCA1I11
G16	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
G17	P20_0 / RLIN26RX / PWGA64O / CAN6RX / INTP6 / CSIG3SI
H1	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
H2	P13_7 / PWGA73O
H3	P13_6 / MEMC0A22 / PWGA72O
H4	ISOVSS
H7	BVSS
H8	BVSS
H9	BVSS
H10	BVSS
H11	EVSS
H14	EVCC
H15	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
H16	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
H17	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
J1	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
J2	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
J3	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
J4	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
J7	BVSS
J8	BVSS
J9	BVSS
J10	EVSS
J11	EVSS
J14	REG1VCC
J15	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
J16	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
J17	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
K1	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
K2	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
K3	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
K4	EVCC
K7	EVSS
K8	EVSS
K9	EVSS
K10	EVSS
K11	EVSS
K14	ISOVSS
K15	AP0_0 / ADCA0I0
K16	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA / ADCA0I2S
K17	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL / ADCA0I3S
L1	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
L2	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
L3	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
L4	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
L7	EVSS
L8	EVSS
L9	EVSS
L10	EVSS
L11	EVSS
L14	EVSS
L15	AP0_4 / ADCA0I4
L16	AP0_2 / ADCA0I2
L17	AP0_1 / ADCA0I1
M1	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
M2	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0 / $\overline{\text{CSIG4SSI}}$
M3	P2_9 / PWGA77O
M4	P2_7 / RLIN210RX
M14	A0VREF
M15	AP0_8 / ADCA0I8
M16	AP0_5 / ADCA0I5
M17	AP0_3 / ADCA0I3
N1	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2 / CSIG4SI
N2	P1_1 / INTP18 / RLIN33TX / CSIG4SC / TAUJ2I1 / TAUJ2O1
N3	P1_3 / INTP19 / CAN3TX / DPIN23 / CSIG4SO / TAUJ2I3 / TAUJ2O3
N4	P2_11 / PWGA79O
N14	A0VSS
N15	AP0_11 / ADCA0I11
N16	AP0_7 / ADCA0I7

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
N17	AP0_6 / ADCA0I6
P1	P1_12 / CAN4RX / INTP4 / RLIN36TX
P2	P1_13 / CAN4TX / RLIN36RX / INTP16
P3	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
P4	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
P5	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
P6	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
P7	P2_13 / RLIN211TX
P8	P2_15 / PWGA75O
P9	EVCC
P10	REG0VCC
P11	ISOVSS
P12	ISOVCL
P13	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
P14	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
P15	AP0_13 / ADCA0I13
P16	AP0_10 / ADCA0I10
P17	AP0_9 / ADCA0I9
R1	P2_6 / ADCA0SEL2 / CSIG4RYI / CSIG4RYO
R2	P2_10 / PWGA78O
R3	JP0_4 / $\overline{\text{DCUTRST}}$
R4	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
R5	P2_1 / RLIN27TX / CAN6TX
R6	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
R7	P1_9 / DPIN20 / INTP21
R8	P3_0 / PWGA76O
R9	FLMD0
R10	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
R11	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
R12	P2_5 / RLIN29TX / $\overline{\text{CSIH4SSI}}$ / ADCA0SEL1
R13	P1_15 / RLIN23TX / CAN7TX
R14	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
R15	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
R16	AP0_14 / ADCA0I14
R17	AP0_12 / ADCA0I12
T1	P2_8 / RLIN210TX
T2	P2_12 / RLIN211RX
T3	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
T4	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
T5	P2_0 / RLIN27RX / CAN6RX / INTP6
T6	P2_14 / PWGA74O
T7	IP0_0 / XT2
T8	AWOVCL
T9	X1
T10	P2_2 / RLIN28RX / CSIH4CSS0

Table 2A.2 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
T11	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
T12	P0_8 / INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
T13	P2_4 / RLIN29RX / ADCA0SEL0 / CSIH4SO
T14	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / RIIC1SCL / SENT0SPCO / ADCA0I1S
T15	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
T16	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
T17	AP0_15 / ADCA0I15
U1	EVSS
U2	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
U3	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
U4	JP0_0 / INTP0 / TAUJ2I0 / TAUJ2O0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO
U5	P1_8
U6	$\overline{\text{RESET}}$
U7	XT1
U8	AWOVSS
U9	X2
U10	P2_3 / RLIN28TX / CSIH4CSS1
U11	JP0_6 / $\overline{\text{EVTO}}$
U12	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20 / CSIH4SC
U13	P1_4 / RLIN35RX / INTP15 / DPIN18 / CSIH4SI
U14	P1_14 / RLIN23RX / CAN7RX / INTP9 / CSIH4RYI / CSIH4RYO
U15	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / RIIC1SDA / SENT0RX / ADCA0I0S
U16	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
U17	A0VSS

Table 2A.3 Reserved

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
A1	BVSS
A2	P24_5 / CAN10RX / INTP20
A3	P24_2 / CAN9TX
A4	P24_3 / CAN9RX / INTP19
A5	P22_7 / MMCA0CMD
A6	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
A7	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
A8	P22_11 / MMCA0DAT2
A9	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
A10	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
A11	P22_15 / MMCA0DAT6
A12	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
A13	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
A14	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
A15	P19_1 / ADCA1I17S
A16	P19_0 / ADCA1I16S
A17	P18_15 / ADCA1I15S
A18	P18_6 / PWGA95O / ADCA1I6S
A19	P18_3 / PWGA71O / ETNB0TXD2 / TAUJ3I3 / TAUJ3O3 / ADCA1I3S
A20	P18_9 / ADCA1I9S
A21	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / TAUJ3I0 / TAUJ3O0 / ADCA1I0S
A22	A1VSS
B1	P24_6 / CAN11TX
B2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
B3	P24_4 / CAN10TX
B4	P24_1 / CAN8RX / INTP18
B5	P24_0 / CAN8TX
B6	P13_1 / MEMC0A20
B7	P22_9 / MMCA0DAT0
B8	P22_12 / MMCA0DAT3
B9	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL
B10	P22_14 / MMCA0DAT5
B11	P21_1 / MMCA0DAT7
B12	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
B13	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
B14	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / TAUJ3I2 / TAUJ3O2 / FLMD1
B15	P19_3 / ADCA1I19S
B16	P18_7 / ETNB0TXCLK / ADCA1I7S
B17	P18_13 / ADCA1I13S

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
B18	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
B19	P18_8 / ADCA1I8S
B20	P18_1 / PWGA62O / ETNB0TXD0 / TAUJ3I1 / TAUJ3O1 / ADCA1I1S
B21	AP1_13 / ADCA1I13
B22	AP1_15 / ADCA1I15
C1	P24_7 / CAN11RX / INTP21
C2	P22_6 / ETNB1TXCLK
C3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O
C4	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / CSIH1SSI / MEMC0CLK / RLIN37RX / INTP17
C5	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1
C6	P22_8 / MMCA0CLK
C7	P13_0 / MEMC0A19
C8	P22_10 / MMCA0DAT1
C9	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
C10	P22_13 / MMCA0DAT4
C11	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
C12	P10_13 / CSIH0SSI / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
C13	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
C14	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
C15	P18_14 / ADCA1I14S
C16	P18_12 / ADCA1I12S
C17	P18_11 / ADCA1I11S
C18	P18_10 / ADCA1I10S
C19	P18_2 / PWGA63O / ETNB0TXD1 / TAUJ3I2 / TAUJ3O2 / ADCA1I2S
C20	AP1_12 / ADCA1I12
C21	AP1_0 / ADCA1I0
C22	AP1_1 / ADCA1I1
D1	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / MEMC0RD
D2	P22_4 / ETNB1TXD0
D3	P22_5 / ETNB1TXEN
D4	BVCC
D5	BVCC
D6	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / TAUJ3O0 / MODE0
D7	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1O3
D8	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
D9	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / MEMC0ASTB / ETNB0RXERR / RLIN36TX
D10	BVSS
D11	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
D12	BVCC
D13	BVSS
D14	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0 / RLIN24RX / MODE2
D15	P19_2 / ADCA1I18S
D16	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
D17	ISOVSS
D18	ISOVCL
D19	BVCC
D20	AP1_14 / ADCA1I14
D21	AP1_3 / ADCA1I3
D22	AP1_5 / ADCA1I5
E1	P22_3 / ETNB1TXD1
E2	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
E3	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
E4	BVCC
E19	A1VSS
E20	AP1_2 / ADCA1I2
E21	AP1_4 / ADCA1I4
E22	AP1_7 / ADCA1I7
F1	P22_2 / ETNB1TXD2
F2	P22_1 / ETNB1TXD3
F3	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
F4	BVCC
F19	A1VREF
F20	AP1_6 / ADCA1I6
F21	AP1_9 / ADCA1I9
F22	AP1_8 / ADCA1I8
G1	P22_0 / ETNB1RXCLK
G2	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
G3	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
G4	BVSS
G19	AP1_10 / ADCA1I10
G20	AP1_11 / ADCA1I11
G21	P20_6 / PWGA88O
G22	P20_7 / PWGA89O
H1	P21_4 / ETNB1RXD0
H2	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
H3	P21_0 / ETNB1RXDV
H4	ISOVCL
H19	EVCC
H20	P20_8 / PWGA90O
H21	P20_9 / PWGA91O
H22	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
J1	P21_5 / ETNB1RXD3

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
J2	P21_2 / ETNB1RXD2
J3	P21_3 / ETNB1RXD1
J4	ISOVSS
J9	BVSS
J10	BVSS
J11	BVSS
J12	BVSS
J13	BVSS
J14	BVSS
J19	EVSS
J20	P20_0 / RLIN26RX / PWGA64O / CAN6RX / INTP6 / CSIG3SI
J21	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
J22	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
K1	P21_6 / ETNB1MDC
K2	P13_5 / MEMC0A21
K3	P13_2 / ETNB0RXDV
K4	BVSS
K9	BVSS
K10	BVSS
K11	BVSS
K12	BVSS
K13	BVSS
K14	EVSS
K19	EVCC
K20	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
K21	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
K22	P20_10 / PWGA92O
L1	P21_7 / ETNB1MDIO
L2	P21_8 / ETNB1RXERR
L3	P13_3 / ETNB0RXERR
L4	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
L9	BVSS
L10	BVSS
L11	BVSS
L12	BVSS
L13	EVSS
L14	EVSS
L19	REG1VCC
L20	P20_11 / PWGA93O
L21	P20_12 / PWGA94O
L22	P20_15 / RLIN214RX
M1	P21_9
M2	P21_10
M3	P13_4 / ETNB1LINK
M4	P13_6 / MEMC0A22 / PWGA72O

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
M9	BVSS
M10	BVSS
M11	BVSS
M12	EVSS
M13	EVSS
M14	EVSS
M19	ISOVSS
M20	P20_13 / RLIN215RX / PWGA95O
M21	P20_14 / RLIN215TX
M22	P23_10 / RLIN214TX
N1	P21_11 / RLIN213RX
N2	P21_13 / RLIN212RX
N3	P13_7 / MEMC0A23 / PWGA73O
N4	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
N9	EVSS
N10	EVSS
N11	EVSS
N12	EVSS
N13	EVSS
N14	EVSS
N19	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
N20	P23_7 / CSIG4SI
N21	P23_8 / CSIG4SC
N22	P23_9 / $\overline{\text{CSIG4SSI}}$
P1	P21_12 / RLIN213TX
P2	P21_14 / RLIN212TX
P3	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
P4	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
P9	EVSS
P10	EVSS
P11	EVSS
P12	EVSS
P13	EVSS
P14	EVSS
P19	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S
P20	P23_6 / CSIG4SO
P21	P23_4 / CSIH4RYI / CSIH4RYO
P22	P23_5 / CSIG4RYI / CSIG4RYO
R1	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
R2	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
R3	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
R4	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
R19	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
R20	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL / ADCA0I3S
R21	P23_2 / CSIH4SI
R22	P23_3 / CSIH4SC
T1	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
T2	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
T3	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
T4	EVCC
T19	EVSS
T20	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA / ADCA0I2S
T21	P23_0 / $\overline{\text{CSIH4SSI}}$
T22	P23_1 / CSIH4SO
U1	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
U2	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
U3	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
U4	P1_1 / INTP18 / RLIN33TX / CSIG4SC / TAUJ2I1 / TAUJ2O1
U19	A0VREF
U20	AP0_5 / ADCA0I5
U21	AP0_2 / ADCA0I2
U22	AP0_0 / ADCA0I0
V1	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0 / $\overline{\text{CSIG4SSI}}$
V2	P1_12 / CAN4RX / INTP4 / RLIN36TX
V3	P1_3 / INTP19 / CAN3TX / DPIN23 / CSIG4SO / TAUJ2I3 / TAUJ2O3
V4	P1_13 / CAN4TX / RLIN36RX / INTP16
V19	A0VSS
V20	AP0_10 / ADCA0I10
V21	AP0_4 / ADCA0I4
V22	AP0_1 / ADCA0I1
W1	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2 / CSIG4SI
W2	P2_7 / RLIN210RX
W3	P2_8 / RLIN210TX
W4	P2_11 / PWGA79O
W5	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
W6	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
W7	P1_9 / DPIN20 / INTP21
W8	EVCC
W9	AWOVCL
W10	REG0VCC
W11	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
W12	ISOVSS
W13	ISOVCL
W14	P1_15 / RLIN23TX / CAN7TX
W15	EVSS
W16	EVCC
W17	P3_7 / CAN10RX / INTP20 / PWGA86O

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
W18	P3_10 / CAN11TX
W19	AP0_14 / ADCA0I14
W20	AP0_8 / ADCA0I8
W21	AP0_6 / ADCA0I6
W22	AP0_3 / ADCA0I3
Y1	P2_6 / ADCA0SEL2 / CSIG4RYI / CSIG4RYO
Y2	P2_12 / RLIN211RX
Y3	P2_9 / PWGA77O
Y4	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
Y5	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
Y6	P2_1 / RLIN27TX / CAN6TX
Y7	P2_13 / RLIN211TX
Y8	P3_0 / PWGA76O
Y9	FLMD0
Y10	JP0_6 / $\overline{\text{EVTO}}$
Y11	P0_8 / INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
Y12	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
Y13	P1_14 / RLIN23RX / CAN7RX / INTP9 / CSIH4RYI / CSIH4RYO
Y14	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / RIIC1SCL / SENT0SPCO / ADCA0I1S
Y15	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
Y16	P3_1 / PWGA80O
Y17	P3_3 / CAN8RX / INTP18 / PWGA82O
Y18	P3_6 / PWGA85O / CAN9TX
Y19	P3_9 / CAN11RX / INTP21
Y20	AP0_13 / ADCA0I13
Y21	AP0_12 / ADCA0I12
Y22	AP0_7 / ADCA0I7
AA1	P2_10 / PWGA78O
AA2	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
AA3	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
AA4	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
AA5	P2_0 / RLIN27RX / CAN6RX / INTP6
AA6	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
AA7	P2_14 / PWGA74O
AA8	IP0_0 / XT2
AA9	P2_15 / PWGA75O
AA10	P2_3 / RLIN28TX / CSIH4CSS1
AA11	P2_2 / RLIN28RX / CSIH4CSS0
AA12	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20 / CSIH4SC
AA13	P1_4 / RLIN35RX / INTP15 / DPIN18 / CSIH4SI
AA14	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / RIIC1SDA / SENT0RX / ADCA0I0S
AA15	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
AA16	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
AA17	P3_12 / CSIH4CSS1

Table 2A.4 Pin Assignment 324-Pin FPBGA

Pin No.	Pin Name
AA18	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
AA19	P3_4 / PWGA83O / CAN8TX
AA20	P3_8 / PWGA87O / CAN10TX
AA21	AP0_11 / ADCA0I11
AA22	AP0_9 / ADCA0I9
AB1	EVSS
AB2	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
AB3	JP0_4 / $\overline{\text{DCUTRST}}$
AB4	JP0_0 / INTP0 / TAUJ2I0 / TAUJ2O0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO
AB5	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
AB6	P1_8
AB7	$\overline{\text{RESET}}$
AB8	XT1
AB9	AWOVSS
AB10	X2
AB11	X1
AB12	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
AB13	P2_4 / RLIN29RX / ADCA0SEL0 / CSIH4SO
AB14	P2_5 / RLIN29TX / $\overline{\text{CSIH4SSI}}$ / ADCA0SEL1
AB15	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
AB16	P3_2 / PWGA81O
AB17	P3_11 / CSIH4CSS0
AB18	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
AB19	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
AB20	P3_5 / CAN9RX / INTP19 / PWGA84O
AB21	AP0_15 / ADCA0I15
AB22	A0VSS

2A.2 Pin Description

Table 2A.5 Pin Functions

	No. of Pins						
Pin Name	176 Pins	233 Pins	272 Pins	324 Pins	IO	Pin Function	Unit
AnVREF	✓	✓	—	✓	—	ADCAn voltage supply and reference voltage	ADCAn
	n = 0, 1	n = 0, 1	—	n = 0, 1			
AnVSS	✓	✓	—	✓	—	ADCAn ground	
	n = 0, 1	n = 0, 1	—	n = 0, 1			
ADCA0Im	✓	✓	—	✓	I	ADCA0 input channel m with 12-bit resolution	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
ADCA1Im	✓	✓	—	✓	I	ADCA1 input channel m with 12-bit resolution	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
ADCA0ImS	✓	✓	—	✓	I	ADCA0 input channel m with 10-bit resolution	
	m = 0 to 11, 14 to 19	m = 0 to 11, 14 to 19	—	m = 0 to 11, 14 to 19			
ADCA1ImS	✓	✓	—	✓	I	ADCA1 input channel m with 10-bit resolution	
	m = 0 to 7	m = 0 to 19	—	m = 0 to 19			
ADCA0SELy	✓	✓	—	✓	O	Selection pin y for ADCA0 input and external MPX	
	y = 0 to 2	y = 0 to 2	—	y = 0 to 2			
ADCAnTRGy	✓	✓	—	✓	I	ADCAn external trigger pin y	
	n = 0, 1 y = 0 to 2	n = 0, 1 y = 0 to 2	—	n = 0, 1 y = 0 to 2			
AP0_m	✓	✓	—	✓	IO	Analog port 0_m	Port
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
AP1_m	✓	✓	—	✓	IO	Analog port 1_m	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
APO	✓	✓	—	✓	O	Port output signal for analog input	LPS0
AWOVCL	✓	✓	—	✓	—	Voltage regulator for Always-On area (AWO area) capacitor connection	Power
AWOVSS	✓	✓	—	✓	—	Internal logic for Always-On area (AWO area) ground	
BVCC	✓	✓	—	✓	—	Port buffer voltage supply	
BVSS	✓	✓	—	✓	—	Port buffer ground	
CANzRX	✓	✓	—	✓	I	CANz receive data input	RCFDCn
	z = 0 to 7	z = 0 to 7	—	z = 0 to 11			
CANzTX	✓	✓	—	✓	O	CANz transmit data output	
	z = 0 to 7	z = 0 to 7	—	z = 0 to 11			
CSCXFOUT	✓	✓	—	✓	O	Clock output	Clock
CSIGnRYI	✓	✓	—	✓	I	CSIGn ready (1) / busy (0) input signal	CSIGn
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIGnRYO	✓	✓	—	✓	O	CSIGn ready (1) / busy (0) output signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIGnSC	✓	✓	—	✓	IO	CSIGn serial clock signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIGnSI	✓	✓	—	✓	I	CSIGn serial data input	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIGnSO	✓	✓	—	✓	O	CSIGn serial data output	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
$\overline{\text{CSIGnSSI}}$	✓	✓	—	✓	I	CSIGn SS function control input signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnCSS0	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnCSS1	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 1	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			

Table 2A.5 Pin Functions

	No. of Pins						
Pin Name	176 Pins	233 Pins	272 Pins	324 Pins	IO	Pin Function	Unit
CSIHnCSS2	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 2	CSIHn
	n = 0 to 3	n = 0 to 3	—	n = 0 to 3			
CSIHnCSS3	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 3	
	n = 0 to 3	n = 0 to 3	—	n = 0 to 3			
CSIHnCSS4	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 4	
	n = 0 to 2	n = 0 to 2	—	n = 0 to 2			
CSIHnCSS5	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 5	
	n = 0 to 2	n = 0 to 2	—	n = 0 to 2			
CSIHnCSS6	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 6	
	n = 0	n = 0	—	n = 0			
CSIHnCSS7	✓	✓	—	✓	O	CSIHn serial peripheral chip select signal 7	
	n = 0	n = 0	—	n = 0			
CSIHnRYI	✓	✓	—	✓	I	CSIHn ready (1) / busy (0) input signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnRYO	✓	✓	—	✓	O	CSIHn ready (1) / busy (0) output signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnSC	✓	✓	—	✓	IO	CSIHn serial clock signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnSI	✓	✓	—	✓	I	CSIHn serial data input	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
CSIHnSO	✓	✓	—	✓	O	CSIHn serial data output	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
$\overline{\text{CSIHnSSI}}$	✓	✓	—	✓	I	CSIHn slave select input signal	
	n = 0 to 4	n = 0 to 4	—	n = 0 to 4			
$\overline{\text{DCURDY}}$	✓	✓	—	✓	O	Debug ready	OCD
DCUTCK	✓	✓	—	✓	I	Debug clock	
DCUTDI	✓	✓	—	✓	I	Debug data input	
DCUTDO	✓	✓	—	✓	O	Debug data output	
DCUTMS	✓	✓	—	✓	I	Debug mode select	
$\overline{\text{DCUTRST}}$	✓	✓	—	✓	I	Debug reset	
DPINm	✓	✓	—	✓	I	Digital port input m	LPS0
	m = 0 to 23	m = 0 to 23	—	m = 0 to 23			
DPO	✓	✓	—	✓	O	Port output signal for digital input	
ENCA0TINm	✓	✓	—	✓	I	ENCA0 capture trigger input m	ENCAn
	m = 0, 1	m = 0, 1	—	m = 0, 1			
ENCA0E0	✓	✓	—	✓	I	ENCA0 encoder input 0	
ENCA0E1	✓	✓	—	✓	I	ENCA0 encoder input 1	
ENCA0EC	✓	✓	—	✓	I	ENCA0 encoder clear input	
ETNBnLINK	✓	✓	—	✓	I	PHY link status	ETNBn
	n = 0	n = 0	—	n = 0, 1			
ETNBnMDC	✓	✓	—	✓	O	PHY management clock	
	n = 0	n = 0	—	n = 0, 1			
ETNBnMDIO	✓	✓	—	✓	IO	Management transmit / receive data signal	
	n = 0	n = 0	—	n = 0, 1			
ETNBnRXCLK	✓	✓	—	✓	I	MII receive clock	
	n = 0	n = 0	—	n = 0, 1			
ETNBnRXD[3:0]	✓	✓	—	✓	I	MII receive data input	
	n = 0	n = 0	—	n = 0, 1			
ETNBnRXDV	✓	✓	—	✓	I	MII receive data valid	
	n = 0	n = 0	—	n = 0, 1			

Table 2A.5 Pin Functions

	No. of Pins						
Pin Name	176 Pins	233 Pins	272 Pins	324 Pins	IO	Pin Function	Unit
ETNBnRXERR	✓	✓	—	✓	I	MII receive error	ETNBn
	n = 0	n = 0	—	n = 0, 1			
ETNBnTXCLK	✓	✓	—	✓	I	MII transmit clock	
	n = 0	n = 0	—	n = 0, 1			
ETNBnTXD[3:0]	✓	✓	—	✓	O	MII transmit data output	
	n = 0	n = 0	—	n = 0, 1			
ETNBnTXEN	✓	✓	—	✓	O	MII transmit data enable	
	n = 0	n = 0	—	n = 0, 1			
EVCC	✓	✓	—	✓	—	Port buffer voltage supply	Power
EVSS	✓	✓	—	✓	—	Port buffer ground	
$\overline{\text{EVTO}}$	✓	✓	—	✓	O	Event output	TEU_OUT
FLMD0	✓	✓	—	✓	I	Operating mode select pin 0	Mode
FLMD1	✓	✓	—	✓	I	Operating mode select pin 1	
FLXA0RXDA	✓	✓	—	✓	I	FLXA0 channel A receive data input	FLXAn
FLXA0RXDB	✓	✓	—	✓	I	FLXA0 channel B receive data input	
FLXA0STPWT	✓	✓	—	✓	I	FLXA0 stop watch trigger input	
FLXA0TXDA	✓	✓	—	✓	O	FLXA0 channel A transmit data output	
FLXA0TXDB	✓	✓	—	✓	O	FLXA0 channel B transmit data output	
FLXA0TXENA	✓	✓	—	✓	O	FLXA0 channel A transmit enable	
FLXA0TXENB	✓	✓	—	✓	O	FLXA0 channel B transmit enable	
FPDR	✓	✓	—	✓	I	Serial Communication Interface RXD	FLASH
FPDT	✓	✓	—	✓	O	Serial Communication Interface TXD	
FPCK	✓	✓	—	✓	I	Serial Communication Interface clock	
INTPm	✓	✓	—	✓	I	External interrupt input m	INTC
	m = 0 to 23	m = 0 to 23	—	m = 0 to 23			
IP0_0	✓	✓	—	✓	I	Input port 0_0	Port
ISOVCL	✓	✓	—	✓	—	Voltage regulator for Isolated area (ISO area) capacitor connection	Power
ISOVSS	✓	✓	—	✓	—	Internal logic for Isolated area (ISO area) ground	
JP0_m	✓	✓	—	✓	IO	JTAG port 0_m	JTAG
	m = 0 to 6	m = 0 to 6	—	m = 0 to 6			
KR0Im	✓	✓	—	✓	I	KR0 key input signal	KRn
	m = 0 to 7	m = 0 to 7	—	m = 0 to 7			
LPDCLK	✓	✓	—	✓	I	LPD clock input (4-pin mode)	LPD
LPDCLKOUT	✓	✓	—	✓	O	LPD clock output (4-pin mode)	
LPDI	✓	✓	—	✓	I	LPD data input (4-pin mode)	
LPDIO	✓	✓	—	✓	IO	LPD data input / output (1-pin mode)	
LPDO	✓	✓	—	✓	O	LPD data output (4-pin mode)	
MEMC0Am	✓	✓	—	✓	O	MEMC0 address m	MEMCn
	m = 16 to 22	m = 16 to 22	—	m = 16 to 23			
MEMC0ADm	✓	✓	—	✓	IO	MEMC0 address / data m	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
$\overline{\text{MEMC0ASTB}}$	✓	✓	—	✓	O	MEMC0 address strobe	
$\overline{\text{MEMC0BENm}}$	✓	✓	—	✓	O	MEMC0 byte enable m	
	m = 0, 1	m = 0, 1	—	m = 0, 1			
MEMC0CLK	✓	✓	—	✓	O	MEMC0 clock output	
$\overline{\text{MEMC0CSm}}$	✓	✓	—	✓	O	MEMC0 chip select m	
	m = 0 to 3	m = 0 to 3	—	m = 0 to 3			
$\overline{\text{MEMC0RD}}$	✓	✓	—	✓	O	MEMC0 read strobe	
$\overline{\text{MEMC0WAIT}}$	✓	✓	—	✓	I	MEMC0 wait input	
$\overline{\text{MEMC0WR}}$	✓	✓	—	✓	O	MEMC0 write strobe	

Table 2A.5 Pin Functions

Pin Name	No. of Pins				IO	Pin Function	Unit
	176 Pins	233 Pins	272 Pins	324 Pins			
MMCA0CLK	—	—	—	✓	O	MMCA Clock	MMCAn
MMCA0CMD	—	—	—	✓	IO	MMCA Command / Response	
MMCA0DATm	—	—	—	✓	IO	MMCA Data[7:0]	
				m = 0 to 7			
MODEm	✓	✓	—	✓	I	Sub operating mode select	Mode
	m = 0 to 2	m = 0 to 2	—	m = 0 to 2			
NMI	✓	✓	—	✓	I	External non-maskable interrupt input	INTC
P0_m	✓	✓	—	✓	IO	Port 0_m	Port
	m = 0 to 14	m = 0 to 14	—	m = 0 to 14			
P1_m	✓	✓	—	✓	IO	Port 1_m	
	m = 0 to 5, 8 to 15	m = 0 to 5, 8 to 15	—	m = 0 to 5, 8 to 15			
P2_m	✓	✓	—	✓	IO	Port 2_m	
	m = 0 to 6	m = 0 to 15	—	m = 0 to 15			
P3_m	—	✓	—	✓	IO	Port3_m	
		m = 0	—	m = 0 to 12			
P8_m	✓	✓	—	✓	IO	Port 8_m	
	m = 0 to 12	m = 0 to 12	—	m = 0 to 12			
P9_m	✓	✓	—	✓	IO	Port 9_m	
	m = 0 to 4	m = 0 to 4	—	m = 0 to 4			
P10_m	✓	✓	—	✓	IO	Port 10_m	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
P11_m	✓	✓	—	✓	IO	Port 11_m	
	m = 0 to 12, 15	m = 0 to 12, 15	—	m = 0 to 12, 15			
P12_m	✓	✓	—	✓	IO	Port 12_m	
	m = 0 to 5	m = 0 to 5	—	m = 0 to 5			
P13_m	—	✓	—	✓	IO	Port 13_m	
		m = 0 to 7	—	m = 0 to 7			
P18_m	✓	✓	—	✓	IO	Port 18_m	
	m = 0 to 7	m = 0 to 15	—	m = 0 to 15			
P19_m	—	✓	—	✓	IO	Port19_m	
		m = 0 to 3	—	m = 0 to 3			
P20_m	✓	✓	—	✓	IO	Port 20_m	
	m = 0 to 5	m = 0 to 5	—	m = 0 to 15			
P21_m	—	—	—	✓	IO	Port21_m	
			—	m = 0 to 14			
P22_m	—	—	—	✓	IO	Port22_m	
			—	m = 0 to 15			
P23_m	—	—	—	✓	IO	Port23_m	
				m = 0 to 10			
P24_m	—	—	—	✓	IO	Port24_m	
				m = 0 to 7			
PWGAnO	✓	✓	—	✓	O	PWGAn output signal	PWGAn
	n = 0 to 71	n = 0 to 79	—	n = 0 to 95			
REGnVCC	✓	✓	—	✓	—	Voltage regulators voltage supply	Power
	n = 0, 1	n = 0, 1	—	n = 0, 1	—		
$\overline{\text{RESET}}$	✓	✓	—	✓	I	External reset input	Reset
$\overline{\text{RESETOUT}}$	✓	✓	—	✓	O	Reset output	

Table 2A.5 Pin Functions

	No. of Pins						
Pin Name	176 Pins	233 Pins	272 Pins	324 Pins	IO	Pin Function	Unit
RIICnSCL	✓	✓	—	✓	IO	RIICn serial clock	RIICn
	n = 0, 1	n = 0, 1	—	n = 0, 1			
RIICnSDA	✓	✓	—	✓	IO	RIICn serial data	
	n = 0, 1	n = 0, 1	—	n = 0, 1			
RLIN2mRX	✓	✓	—	✓	I	RLIN2m receive data input	RLIN24n
	m = 0 to 9	m = 0 to 11	—	m = 0 to 15			
RLIN2mTX	✓	✓	—	✓	O	RLIN2m transmit data output	
	m = 0 to 9	m = 0 to 11	—	m = 0 to 15			
RLIN3nRX	✓	✓	—	✓	I	RLIN3n receive data input	RLIN3n
	n = 0 to 7	n = 0 to 7	—	n = 0 to 7			
RLIN3nTX	✓	✓	—	✓	O	RLIN3n transmit data output	
	n = 0 to 7	n = 0 to 7	—	n = 0 to 7			
RTCA0OUT	✓	✓	—	✓	O	RTCA0 1Hz output	RTCAn
SELDPk	✓	✓	—	✓	O	External multiplexer select signal output k for the digital port	LPS0
	k = 0 to 2	k = 0 to 2	—	k = 0 to 2			
SENTnRX	✓	✓	—	✓	I	SENT receive data input	RSENTn
	n = 0, 1	n = 0, 1	—	n = 0, 1			
SENTnSPCO	✓	✓	—	✓	O	SENT SPC Extension Output	
	n = 0, 1	n = 0, 1	—	n = 0, 1			
SFMA0CLK	✓	✓	—	✓	O	SFMA0 clock	SFMAn
SFMA0IOm	✓	✓	—	✓	IO	SFMA0 master data input / output	
	m = 0 to 3	m = 0 to 3	—	m = 0 to 3			
SFMA0SSL	✓	✓	—	✓	O	SFMA0 slave select	
TAPA0ESO	✓	✓	—	✓	I	Hi-Z control	TAPAn
TAPA0UN	✓	✓	—	✓	O	Motor control output U phase (negative)	
TAPA0UP	✓	✓	—	✓	O	Motor control output U phase (positive)	
TAPA0VN	✓	✓	—	✓	O	Motor control output V phase (negative)	
TAPA0VP	✓	✓	—	✓	O	Motor control output V phase (positive)	
TAPA0WN	✓	✓	—	✓	O	Motor control output W phase (negative)	
TAPA0WP	✓	✓	—	✓	O	Motor control output W phase (positive)	
TAUBnIm	✓	✓	—	✓	I	TAUBn channel input m	TAUBn
	n = 0, 1 m = 0 to 15	n = 0, 1 m = 0 to 15	—	n = 0, 1 m = 0 to 15			
TAUBnOm	✓	✓	—	✓	O	TAUBn channel output m	
	n = 0, 1 m = 0 to 15	n = 0, 1 m = 0 to 15	—	n = 0, 1 m = 0 to 15			
TAUD0Im	✓	✓	—	✓	I	TAUD0 channel input m	TAUDn
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
TAUD0Om	✓	✓	—	✓	O	TAUD0 channel output m	
	m = 0 to 15	m = 0 to 15	—	m = 0 to 15			
TAUJnIm	✓	✓	—	✓	I	TAUJn channel input m	TAUJn
	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	—	n = 0 to 3 m = 0 to 3			
TAUJnOm	✓	✓	—	✓	O	TAUJn channel output m	
	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	—	n = 0 to 3 m = 0 to 3			
X1, X2	✓	✓	—	✓	—	Main OSC connections	MOSC
XT1, XT2	✓	✓	—	✓	—	Sub OSC connections	SOSC

CAUTION

When pin functions for a peripheral module are allocated to multiple pins, use the pins from the same port group or nearby pins as the pins for a given channel.

- **(e.g.)** When RS-CANFD channel 0 is used:

CAN0TX	P0_0	P10_1
CAN0RX	P0_1	P10_0

Use one of the following pin combinations:

- P0_0 and P0_1, or
- P10_0 and P10_1.

The combinations of P0_0 and P10_0, and P0_1 and P10_1 are not allowed.

- **(e.g.)** When CSIH4 is used:

CSIH4SC	P1_5	P23_3
CSIH4SO	P2_4	P23_1
CSIH4SI	P1_4	P23_2

Use one of the following pin combinations:

- P1_5, P2_4 and P1_4, or
- P23_3, 23_1 and P23_2.

The pin combinations of the following are not allowed:

- P1_5, P2_4 and P23_2
- P1_5, P23_1 and P1_4
- P23_3, P2_4 and P23_2
- P1_5, P23_1 and P23_2
- P23_3, P2_4 and P1_4
- P23_3, P23_1 and P1_4.

2A.3 Pin Functions During and After Reset

Table 2A.6 Pin Functions During and After Reset

Pins	During Reset	After Reset
JP0_0	High impedance	JP0_0: Input Serial programming mode: FPDR, FPDT (1 wire UART) FPDR (2 wire UART) Nexus I/F: DCUTDI input LPD (4 pins): LPDI input LPD (1 pin): LPDIO input/output
JP0_1	High impedance	JP0_1: Input Serial programming mode: FPDT Nexus I/F: DCUTDO output LPD (4 pins): LPDO output LPD (1 pin): High impedance
JP0_2	High impedance	JP0_2: Input Serial programming mode: FPCK Nexus I/F: DCUTCK input LPD (4 pins): LPDCLK input LPD (1 pin): High impedance
JP0_3	High impedance	JP0_3: Input Serial programming mode: High impedance Nexus I/F: DCUTMS input LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_4	Input[*3,*5]	JP0_4: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCUTRST}}$ input[*1] LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_5	High impedance	JP0_5: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCURDY}}$ output LPD (4 pins): LPDCLKOUT output LPD (1 pin): High impedance
JP0_6	High impedance	JP0_6: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{EVTO}}$ output LPD (4 pins): High impedance LPD (1 pin): High impedance
P8_6	Output[*2,*4]	Output (OPBT0.RESETOUTEN = 1)[*2] High impedance (OPBT0.RESETOUTEN = 0)[*2,*4]
P0 to P3, P8 to P13, P18 to P24 (except P8_6, P10_1, P10_2, P10_6 and P10_8)	High impedance	High impedance
P10_1	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE0 input (FLMD0 = 1, FLMD1 = 1)
P10_2	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE1 input (FLMD0 = 1, FLMD1 = 1)

Table 2A.6 Pin Functions During and After Reset

Pins	During Reset	After Reset
P10_6	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 1) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 0) MODE2 input (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 1)
P10_8	High impedance	High impedance (FLMD0 = 0) FLMD1 input (FLMD0 = 1)
FLMD0	Input	Input
$\overline{\text{RESET}}$	Input	Input
AP0, AP1	High impedance	High impedance

Note 1. When Nexus is enabled and no external device is connected, the level of the pin must always be fixed to low level.

Note 2. $\overline{\text{RESETOUT}}$ is output. For details, see **Section 2A.11, Port (Special I/O) Function Overview**.

Note 3. When the power is turned on or when $\overline{\text{RESET}}$ is low level, JP0_4 pin should be driven low level.

Note 4. If OPBT0.RESETOUTEN = 0, P8_6 pin status has a possibility to become unstable (less than 15 μs) at the transition moment to reset status by internal reset factors.

Note 5. When $\overline{\text{RESET}}$ is low level, on-chip pull-down resistor is connected to JP0_4.

2A.4 Port State in Standby Mode

For the port state in standby mode, see **Section 14.1.4, I/O Buffer Control**.

2A.5 Recommended Connection of Unused Pins

If the pins are not used, it is recommended to connect them as shown below.

Table 2A.7 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
A0VREF, A1VREF	Connected to EVCC or BVCC
A0VSS, A1VSS	Connected to EVSS or BVSS
$\overline{\text{RESET}}$	Connected to EVCC or BVCC via a resistor
XT1	Connected to REGnVCC or AWOVSS via a resistor*2 (bit 0 of IPIBC0 = 1) Connected to AWOVSS (bit 0 of IPIBC0 = 0)
X1	Connected to AWOVSS via a resistor
X2	Open
IP0_0	Connected to REGnVCC or AWOVSS via a resistor*2 (bit 0 of IPIBC0 = 1) Open (bit 0 of IPIBC0 = 0)
JP0 (excluding JP0_4) P0 P1 P2 P3 P8 (excluding P8_6) P9 P20 P23	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P8_6	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS via a resistor
P10_1, P10_2, P10_6, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open

Table 2A.7 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
P10 (excluding P10_1, P10_2, P10_6, P10_8) P11 P12 P13 P18 P19 P21 P22 P24	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to BVCC or BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
AP1	Input: Open (when the PIBCn_m bit is 0) Connected to A1VREF or A1VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC via a resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Connected to EVCC via a resistor $\overline{\text{DCUTRST}}$ (JP0_4): Connected to EVSS via a resistor*[1] $\overline{\text{DCURDY}}$ /LPDCLKOUT (JP0_5): Open $\overline{\text{EVTO}}$ (JP0_6): Open

Note 1. For in case when a debugging interface is used, this pin should be connected to EVCC through resistor depending on the development tool made by a third party.

Note 2. XT1 = IP0_0 (XT2) = REGnVCC or AWOVSS should be set.
XT1 is connected to IP0_0 (XT2) through an internal resistor. Therefore, it is necessary to maintain equal voltage level in order not to make a current path.

2A.6 Features of RH850/F1KH Port

2A.6.1 Port Group

The RH850/F1KH provides the following port groups, indicated by the numbers in the table below.

Table 2A.8 Port Groups in RH850/F1KH-D8

No. of Pins	Port Group	RH850/F1KH-D8
176 pins	Number	14
	Name	P0 to P2, P8 to P12, P18, P20, JP0, AP0,AP1, IP0
233 pins	Number	17
	Name	P0 to P3, P8 to P13, P18 to P20, JP0, AP0, AP1, IP0
324 pins	Number	21
	Name	P0 to P3, P8 to P13, P18 to P24, JP0, AP0, AP1, IP0

2A.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n”. For example, the port mode control register of the Pn pin is PMCn (n = 0 to 3, 8 to 13, 18 to 24).

2A.6.3 Register Base Addresses

Port and JTAG port base addresses are listed in the following table.

Port and JTAG port register addresses are given as offsets from the base addresses.

Table 2A.9 Register Base Addresses

Base Address Name	Base Address
<PORTn_base>	FFC1 0000_H
<JPORT0_base>	FFC2 0000_H

2A.6.4 Clock Supply

The clock supply to ports is shown in the following table.

Table 2A.10 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Port	Register access clock	CPUCLK_UL

2A.7 Port Functions

The microcontroller has various pins for input/output functions, known as ports. The ports are organized in port groups.

The RH850/F1KH also has several control registers to enable pins to be used as other than general-purpose input/output pins.

For a description of the terms pin, port, and port group, see **Section 2A.7.2, Terms**.

2A.7.1 Functional Overview

- All the port settings can be specified individually.
- The maximum number of bits (pins) in a port is 16.
- The output level of any pin can be set independently without affecting the other pins in the same port.
- Input buffers are enabled through registers settings.
- Pin level is read by dedicated port-pin-read register (PPR)
- All possible port functions are shown in the tables listed below.
  Table 2A.39, JTAG Port 0 (JP0), **Table 2A.41, Port 0 (P0)**, **Table 2A.43, Port 1 (P1)**, **Table 2A.45, Port 2 (P2)**, **Table 2A.47, Port 3 (P3)**, **Table 2A.49, Port 8 (P8)**, **Table 2A.51, Port 9 (P9)**, **Table 2A.53, Port 10 (P10)**, **Table 2A.55, Port 11 (P11)**, **Table 2A.57, Port 12 (P12)**, **Table 2A.59, Port 13 (P13)**, **Table 2A.61, Port 18 (P18)**, **Table 2A.63, Port 19 (P19)**, **Table 2A.65, Port 20 (P20)**, **Table 2A.67, Port 21 (P21)**, **Table 2A.69, Port 22 (P22)**, **Table 2A.71, Port 23 (P23)**, **Table 2A.73, Port 24 (P24)**, **Table 2A.75, Analog Port 0 (AP0)**, **Table 2A.77, Analog Port 1 (AP1)**, **Table 2A.79, Input Port 0 (IP0)**, and **Section 2A.9.2, Pin Function Configuration**.

CAUTION

Some input or output functions may be assigned to more than one port. Only activate a given function on a single pin. Do not activate a function on multiple pins at the same time. This also applies in cases where multiple peripheral functions are assigned to a single multiplexed function and only one of these functions is used.

[Example]

INTP0 is assigned to the following pins on this device. However, the INTP0 function should not be activated on more than one pin. After activating the function on one pin, do not activate it on another.

- JP0_0 (1st input alternative function)
- P0_1 (2nd, 3rd input alternative function)
- P10_0 (2nd input alternative function)

In the above case, when the 1st input alternative function (INTP0) of JP0_0 is selected, using the 2nd input alternative function (CAN0RX/INTP0) of P0_1 only for the CAN signal is also prohibited.

2A.7.2 Terms

The following terms are used in this section:

Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

Port mode and ports

A pin in port mode works as a general-purpose input/output pin. It is then called “port”.

The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions, such as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that two different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function of the pin at that time.

2A.7.2.1 JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging.

JTAG port group registers and bit names are prefixed by a “J”. For example, JP0 denotes JTAG port group 0, and JPM0.JPM0_m denotes the JPM0_m port mode bit of the JPM0 port mode register.

NOTE

In this section, the descriptions about all ports and their registers other than PFCAEn and PIPCn apply to the JTAG port unless otherwise specified.

2A.7.3 Overview of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCn_m bit = 0)
  A pin in port mode operates as a general-purpose input/output pin. The I/O mode is selected by setting the PMn.PMn_m bit.
- Software I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 0)
  In this mode, the pins operate as alternative functions. The I/O mode is selected by setting the PMn.PMn_m bit.
- Direct I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 1)
  In this mode, the pins operate as alternative functions. Unlike the software I/O control alternative mode, however, the I/O mode is directly controlled by the alternative function.

An overview of the register settings is given in the tables below.

Table 2A.11 Pin Function Configuration (Overview)

	Bit			
Mode	PMCn_m	PMn_m	PIPCn_m	I/O
Port mode	0	0	X	O
		1*1		I
Software I/O control alternative mode	1	0	0	O
		1	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

Note 1. The input buffer must be enabled (PIBCn_m bit = 1).

- Software I/O control alternative mode (PIPCn.PIPCn_m bit = 0)
  - Output (PMn_m bit = 0): Alternative output mode 1 to Alternative output mode 7
  - Input (PMn_m bit = 1): Alternative input mode 1 to Alternative input mode 7
- Direct I/O control alternative mode (PIPCn.PIPCn_m bit = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 7 and Alternative input mode 1 to Alternative input mode 7 is directly selected by the alternative function.

Table 2A.12 Alternative Mode Selection Overview (PMCn.PMCn_m Bit = 1)

Mode	Register					I/O
	PIPC*1	PM*1	PFCAE	PFCE	PFC	
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)		1				I
Alternative output mode 2 (ALT-OUT2)		0			1	O
Alternative input mode 2 (ALT-IN2)		1				I
Alternative output mode 3 (ALT-OUT3)		0		1	0	O
Alternative input mode 3 (ALT-IN3)		1				I
Alternative output mode 4 (ALT-OUT4)		0			1	O
Alternative input mode 4 (ALT-IN4)		1				I
Alternative output mode 5 (ALT-OUT5)		0	1	0	0	O
Alternative input mode 5 (ALT-IN5)		1				I
Alternative output mode 6 (ALT-OUT6)		0			1	O
Alternative input mode 6 (ALT-IN6)		1				I
Alternative output mode 7 (ALT-OUT7)		0		1	0	O
Alternative input mode 7 (ALT-IN7)		1				I
Other than the above	Setting prohibited					

Note 1. If PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

If a pin is in alternative mode (PMCn.PMCn_m bit = 1), one of up to seven alternative functions can be selected for that pin by using the PFCn, PFCEn, and PFCAEn registers.

2A.7.4 Pin Data Input/Output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.

2A.7.4.1 Output Data

In the port mode (PMCn.PMCn_m = 0), the value of the Pn.Pn_m is output from the Pn_m pin.

2A.7.4.2 Input Data

When the PPRn register is read, either the value of the Pn_m pin, the value of the Pn.Pn_m, or the value output by the alternative function is returned.

Which value is returned depends on the pin mode and setting of several control bits.

The different PPRn read modes are shown in the table below.

Table 2A.13 PPRn_m Read Values

PMC n_m	PM n_m	PIBC n_m	PIPC n_m	PODC n_m	Mode	PPRn_m Read Value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m bit
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m bit^{*1}
				1	Port open-drain output	
1	1	X	0	X	Software I/O control alternative input	Pn_m pin
	0			0	Software I/O control alternative push-pull output	Output signal from the alternative function^{*1}
				1	Software I/O control alternative open-drain output	
	X		1	0	Direct I/O control alternative input or push-pull output	I/O port in alternative mode: ● Input: Pn_m pin ● Output: Output signal from the alternative function^{*1}
				1	Direct I/O control alternative input or open-drain output	

Note 1. When PBDCn_m = 1, the level of the Pn_m pin is returned by the PPRn_m bit.

The control registers in the above table have the following effects:

- PMCn.PMCn_m bit
  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).
- PMn.PMn_m bit
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) when the port mode (PMCn_m = 0) and software I/O control alternative mode (PMCn_m = 1, PIPCn_m = 0) have been selected.
- PIBCn.PIBCn_m bit
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). If the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit; otherwise the Pn_m pin level is returned.
- PIPCn.PIPCn_m bit
  This bit selects software I/O control alternative mode or direct I/O control alternative mode.
- PODCn.PODCn_m bit
  This bit selects push-pull output (PODCn_m = 0) or open-drain output (PODCn_m = 1).

- PBDCn.PBDCn_m bit
  In output mode, when this bit is set to 1, the pin enters the bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

CAUTION

When using Pn_m as an alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read at the PPRn.PPRn_m bit by enabling bidirectional mode (PBDCn.PBDCn_m = 1).
Note, however, that the level of the Pn_m pin will be input to the alternative function that the Pn_m pin is being used as.

2A.7.4.3 Writing to the Pn Register

The data to be output via port Pn_m in port mode (PMCn.PMCn_m bit = 0) is held in port register Pn.

Pn data can be overwritten in two ways:

- By writing data directly to the Pn register.
  In this case, new data can be written directly to the Pn register.
- By performing an indirect bitwise operation (a “set”, “reset”, or “not” operation) on the Pn register.
  An indirect bitwise operation (“set”, “reset”, or “not”) can be performed on the Pn register by using the following two registers:
  - Port Set/Reset register PSRn
    If the PSRn.PSRn (m + 16) bit = 1, the value of the Pn.Pn_m bit is determined by the value of the PSRn.PSRn_m bit.
    In other words, the Pn_m bit can be set or reset without writing directly to the Pn register.
  - Port NOT register PNOTn
    By setting PNOTn.PNOTn_m to 1, the Pn.Pn_m bit can be inverted without writing directly to the Pn register.

An indirect bitwise operation on the Pn register (“set”, “reset”, or “not”) has no effect on the bits that do not need to be updated, allowing you to overwrite only the bit or bits that need to be overwritten.

2A.8 Schematic View of Port Control

The following figure is a schematic view of the port control functions.

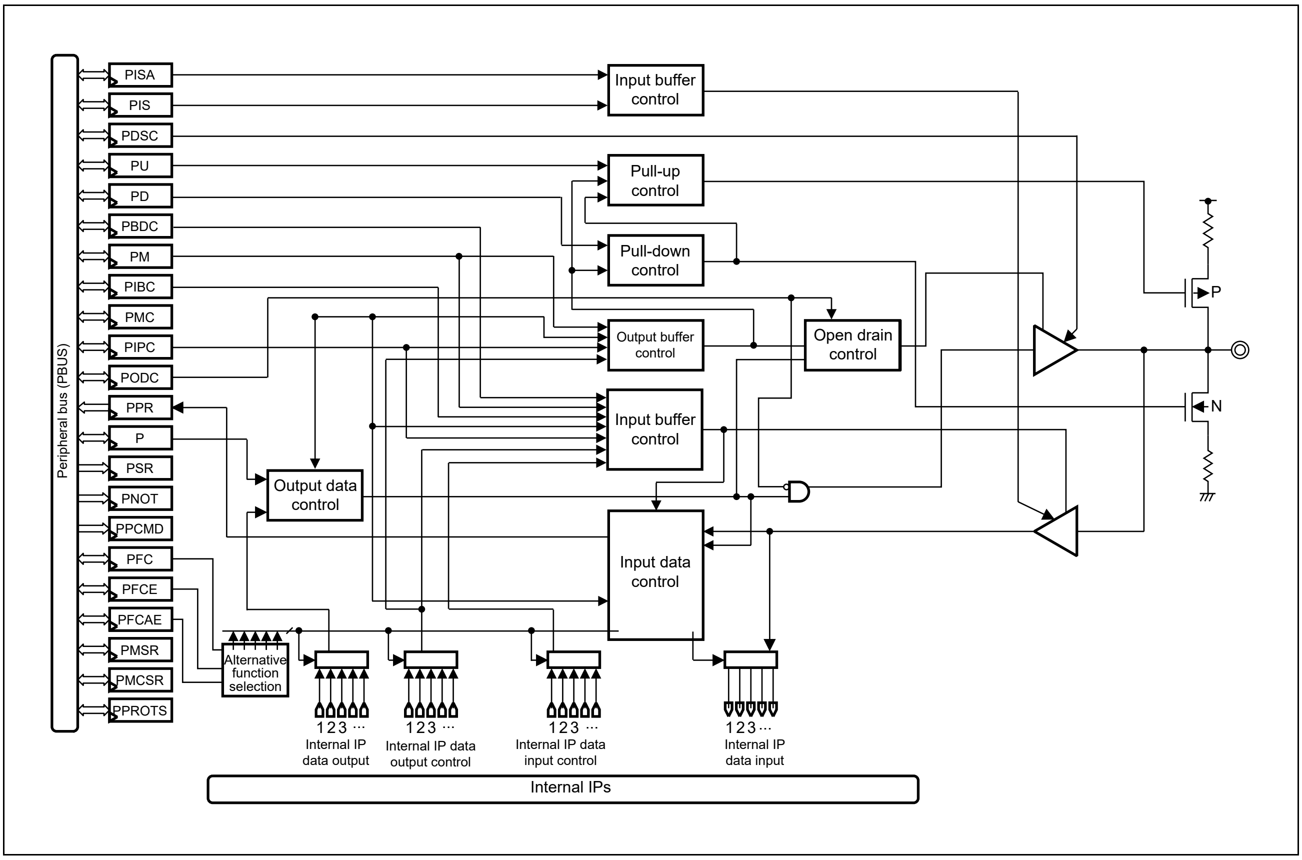


Figure 2A.5 Schematic View of Port Control

CAUTION

Use documented alternative functions only. The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.9 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then describes all registers in detail. The configuration registers are grouped as follows:

- **Section 2A.9.2, Pin Function Configuration**
- **Section 2A.9.3, Pin Data Input/Output**
- **Section 2A.9.4, Configuration of Electrical Characteristics**

2A.9.1 Overview

The following registers are used for setting the individual pins of the port groups.

For details on <PORTn_base> and <JPORT0_base>, see **Section 2A.6.3, Register Base Addresses**.

Table 2A.14 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin function configuration			
PORT	Port mode control register	PMCn	<PORTn_base> + 0400_{H} + n × 4
JTAG		JPMC0	<JPORT0_base> + 0040_{H}
PORT	Port mode control set/reset register	PMCSRn	<PORTn_base> + 0900_{H} + n × 4
JTAG		JPMCSR0	<JPORT0_base> + 0090_{H}
PORT	Port IP control register	PIPCn	<PORTn_base> + 4200_{H} + n × 4
PORT	Port mode register	PMn	<PORTn_base> + 0300_{H} + n × 4
		APMn	<PORTn_base> + $03C8_{H}$ + n × 4
JTAG		JPM0	<JPORT0_base> + 0030_{H}
PORT	Port mode set/reset register	PMSRn	<PORTn_base> + 0800_{H} + n × 4
		APMSRn	<PORTn_base> + $08C8_{H}$ + n × 4
JTAG		JPMSR0	<JPORT0_base> + 0080_{H}
PORT	Port input buffer control register	PIBCn	<PORTn_base> + 4000_{H} + n × 4
		APIBCn	<PORTn_base> + $40C8_{H}$ + n × 4
JTAG		JPIBC0	<JPORT0_base> + 0400_{H}
PORT		IPIBC0	<PORTn_base> + $40F0_{H}$
PORT	Port function control register	PFCn	<PORTn_base> + 0500_{H} + n × 4
JTAG		JPFC0	<JPORT0_base> + 0050_{H}
PORT	Port function control expansion register	PFCEn	<PORTn_base> + 0600_{H} + n × 4
JTAG		JPFCE0	<JPORT0_base> + 0060_{H}
PORT	Port function control additional expansion register	PFCAEn	<PORTn_base> + $0A00_{H}$ + n × 4
Pin data input/output			
PORT	Port bidirection control register	PBDCn	<PORTn_base> + 4100_{H} + n × 4
		APBDCn	<PORTn_base> + $41C8_{H}$ + n × 4
JTAG		JPBDC0	<JPORT0_base> + 0410_{H}
PORT	Port pin read register	PPRn	<PORTn_base> + 0200_{H} + n × 4
		APPRn	<PORTn_base> + $02C8_{H}$ + n × 4
JTAG		JPPR0	<JPORT0_base> + 0020_{H}
PORT		IPPR0	<PORTn_base> + $02F0_{H}$

Table 2A.14 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin data input/output			
PORT	Port register	Pn	<PORTn_base> + 0000$_H$ + n × 4
		APn	<PORTn_base> + 00C8$_H$ + n × 4
JTAG		JP0	<JPORT0_base> + 0000$_H$
PORT	Port NOT register	PNOTn	<PORTn_base> + 0700$_H$ + n × 4
		APNOTn	<PORTn_base> + 07C8$_H$ + n × 4
JTAG		JPNOT0	<JPORT0_base> + 0070$_H$
PORT	Port set/reset register	PSRn	<PORTn_base> + 0100$_H$ + n × 4
		APSRn	<PORTn_base> + 01C8$_H$ + n × 4
JTAG		JPSR0	<JPORT0_base> + 0010$_H$
Configuration of electrical characteristics			
PORT	Pull-up option register	PUn	<PORTn_base> + 4300$_H$ + n × 4
JTAG		JPU0	<JPORT0_base> + 0430$_H$
PORT	Pull-down option register	PDn	<PORTn_base> + 4400$_H$ + n × 4
JTAG		JPD0	<JPORT0_base> + 0440$_H$
PORT	Port drive strength control register	PDSCn	<PORTn_base> + 4600$_H$ + n × 4
JTAG		JPDSC0	<JPORT0_base> + 0460$_H$
PORT	Port open drain control register	PODCn	<PORTn_base> + 4500$_H$ + n × 4
JTAG		JPODC0	<JPORT0_base> + 0450$_H$
PORT	Port input buffer selection register	PISn	<PORTn_base> + 4700$_H$ + n × 4
JTAG		JPIS0	<JPORT0_base> + 0470$_H$
PORT	Port input buffer selection advanced register	PISAn	<PORTn_base> + 4A00$_H$ + n × 4
JTAG		JPISA0	<JPORT0_base> + 04A0$_H$
Port register protection			
PORT	Port protection command register	PPCMDn	<PORTn_base> + 4C00$_H$ + n × 4
JTAG		JPPCMD0	<JPORT0_base> + 04C0$_H$
PORT	Port protection status register	PPROTSn	<PORTn_base> + 4B00$_H$ + n × 4
JTAG		JPPROTS0	<JPORT0_base> + 04B0$_H$

Index n

In **Table 2A.14, Port Group Configuration Registers**, the index “n” in register symbols denotes the actual indices of the individual port groups. For example, PMCn generically indicates a port mode control register for port group n (Pn). The values for n differ according to the number of pins on the device in the way shown in **Table 2A.15, Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols**.

Table 2A.15 Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols

Number of Pins on the Device	Port Groups	Values for “n”
176 pins	P0, P1, P2, P8, P9, P10, P11, P12, P18, P20	0,1,2,8,9,10,11,12,18,20
	AP0, AP1	0, 1
233 pins	P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20	0,1,2,3,8,9,10,11,12,13,18,19,20
	AP0, AP1	0, 1
324 pins	P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24	0,1,2,3,8,9,10,11,12,13,18,19,20,21, 22,23,24
	AP0, AP1	0, 1

JTAG port registers

JTAG port registers are not explicitly described in the following register descriptions.

All descriptions (except for those of the PFCAEn register and PIPCn register) apply to JTAG port registers. Note, however, that the JTAG port register base address differs from that of regular ports.

Value after reset

The values after reset depend on the ports. For the values after reset, see the register descriptions in the following pages.

2A.9.2 Pin Function Configuration

2A.9.2.1 PMCn / JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access: PMCn: This register can be read or written in 16-bit units.
JPMC0: This register can be read or written in 8-bit units.

Address: PMCn: <PORTn_base> + 0400_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20, 21, 22, 23, 24)
JPMC0: <JPORT0_base> + 0040_H[*1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC n_15	PMC n_14	PMC n_13	PMC n_12	PMC n_11	PMC n_10	PMC n_9	PMC n_8	PMC n_7	PMC n_6	PMC n_5	PMC n_4	PMC n_3	PMC n_2	PMC n_1	PMC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.16 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

CAUTIONS

1. I/O is not controlled by only setting alternative mode (PMCn.PMCn_m bit = 1). If the alternative function requires direct I/O control, also set the PIPCn.PIPCn_m bit to 1.
2. If a port is to be used as an input pin in alternative mode, the signals from some pins will pass through a noise filter. These pins may require the setting of the FCLA0CTLm_<*name*>, DNFA<*name*>CTL and the DNFA<*name*>EN register. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**, and **Section 2A.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].

2A.9.2.2 PMCSRn / JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value of PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access: PMCSRn: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of register PMCn.

JPMCSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JPMC0.

Address: PMCSRn: <PORTn_base> + 0900_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20, 21, 22, 23, 24)

JPMCSR0: <JPORT0_base> + 0090_H*1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn_31	PMCSRn_30	PMCSRn_29	PMCSRn_28	PMCSRn_27	PMCSRn_26	PMCSRn_25	PMCSRn_24	PMCSRn_23	PMCSRn_22	PMCSRn_21	PMCSRn_20	PMCSRn_19	PMCSRn_18	PMCSRn_17	PMCSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn_15	PMCSRn_14	PMCSRn_13	PMCSRn_12	PMCSRn_11	PMCSRn_10	PMCSRn_9	PMCSRn_8	PMCSRn_7	PMCSRn_6	PMCSRn_5	PMCSRn_4	PMCSRn_3	PMCSRn_2	PMCSRn_1	PMCSRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.17 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMCSRn_m (PMCSRn_[15:0]) is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m. 1: PMCn_m is PMCSRn_m. Example: If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_[15:0]	Data bits that specify the value of PMCn_m if PMCSRn_m of the corresponding upper bit (PMCSRn_[31:16]) is 1. 0: PMCn_m is 0. 1: PMCn_m is 1.

NOTE

The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].

2A.9.2.3 PIPCn — Port IP Control Register

This register specifies whether the I/O direction of the Pn_m pin is controlled by the port mode register PMn.PMn_m or by an alternative function.

If the Pn_m pin is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn_m setting.

Regarding the alternative functions for which the PIPC register must be set, see **Section 2A.11, Port (Special I/O) Function Overview**.

Access: This register can be read or written in 16-bit units.

Address: PIPCn: <PORTn_base> + 4200_H + n × 4 (n = 0, 1, 2, 10, 11, 12, 20, 21, 22, 23)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPC n_15	PIPC n_14	PIPC n_13	PIPC n_12	PIPC n_11	PIPC n_10	PIPC n_9	PIPC n_8	PIPC n_7	PIPC n_6	PIPC n_5	PIPC n_4	PIPC n_3	PIPC n_2	PIPC n_1	PIPC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, and **Table 2A.72, Control Registers (P23)**.

Table 2A.18 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

2A.9.2.4 PMn / APMn / JPM0 — Port Mode Register

This register specifies whether the individual pins of the port group n are in input mode or in output mode.

Access: PMn, APMn: These registers can be read or written in 16-bit units.

JPM0: This register can be read or written in 8-bit units.

Address: PMn: <PORTn_base> + 0300$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APMn: <PORTn_base> + 03C8$_H$ + n × 4 (n = 0, 1)

JPM0: <JPORT0_base> + 0030$_H$*1

Value after reset: FFFF$_H$*2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	1	1	1	1	1	1	1	1	1	1*3	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1)**.

Note 2. The PM8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8 register is FFBF$_H$.
When the OPBT0.RESETOUTEN = 0, the PM8 register is FFFF$_H$.

Note 3. The PM8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PM8_6 bit is 1.

Table 2A.19 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

NOTES

1. To use a port in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).
2. By default, PMn.PMn_m specifies the I/O direction in port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m=1), since PIPCn.PIPCn_m = 0 after reset.
3. The control bits of the analog port register (APMn) are APMn_[15:0].
4. The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].

2A.9.2.5 PMSRn / APMSRn / JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write data to the PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn_m is set by the corresponding bit in the lower 16 bits of PMSRn.

Access:	PMSRn, APMSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of registers PMn and APMn. JPMSR0: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Bits 15 to 8 are read as FF_H. Reading bits 7 to 0 returns the value of register JPM0.
Address:	PMSRn: <PORTn_base> + 0800_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24) APMSRn: <PORTn_base> + $08C8_H$ + n × 4 (n = 0, 1) JPMSR0: <JPORT0_base> + 0080_H[*1]
Value after reset:	0000 $FFFF_H$[*2]

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSR n_31	PMSR n_30	PMSR n_29	PMSR n_28	PMSR n_27	PMSR n_26	PMSR n_25	PMSR n_24	PMSR n_23	PMSR n_22	PMSR n_21	PMSR n_20	PMSR n_19	PMSR n_18	PMSR n_17	PMSR n_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSR n_15	PMSR n_14	PMSR n_13	PMSR n_12	PMSR n_11	PMSR n_10	PMSR n_9	PMSR n_8	PMSR n_7	PMSR n_6	PMSR n_5	PMSR n_4	PMSR n_3	PMSR n_2	PMSR n_1	PMSR n_0
Value after reset	1	1	1	1	1	1	1	1	1	1[*3]	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1)**.

Note 2. The PMSR8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8 register is 0000 $FFBF_H$.
When the OPBT0.RESETOUTEN = 0, the PMSR8 register is 0000 $FFFF_H$.

Note 3. The PMSR8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PMSR8_6 bit is 1.

Table 2A.20 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMSRn_m (PMSRn_[15:0]) is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m. **Example:** If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Data bits that specify the value of PMn_m if PMSRn_m of the corresponding upper bit (PMSRn_[31:16]) is 1. 0: PMn_m is 0. 1: PMn_m is 1.

NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[31:0].
2. The control bits of the analog port mode set/reset register (APMSRn) are APMSRn_[31:0].

2A.9.2.6 PIBCn / APIBCn / JPIBC0 / IPIBC0 — Port Input Buffer Control Register

In input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables the port pin's input buffer.

Access: PIBCn, APIBCn, IPIBC0: These registers can be read or written in 16-bit units.

JPIBC0: This register can be read or written in 8-bit units.

Address: PIBCn: <PORTn_base> + 4000_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APIBCn: <PORTn_base> + $40C8_H$+ n × 4 (n = 0, 1)

JPIBC0: <JPORT0_base> + 0400_H

IPIBC0: <PORTn_base> + $40F0_H{}^{*1}$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBC n_15	PIBC n_14	PIBC n_13	PIBC n_12	PIBC n_11	PIBC n_10	PIBC n_9	PIBC n_8	PIBC n_7	PIBC n_6	PIBC n_5	PIBC n_4	PIBC n_3	PIBC n_2	PIBC n_1	PIBC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, **Table 2A.78, Control Registers (AP1)**, and **Table 2A.80, Control Registers (IP0)**.

Table 2A.21 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled

NOTES

1. When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].

CAUTION

Settings in this register are overruled in bidirectional mode (PBDCn.PBDCn_m = 1).

2A.9.2.7 PFCn / JPFC0 — Port Function Control Register

This register, together with register PFCEn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is selected by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCn: This register can be read or written in 16-bit units.
JPFC0: This register can be read or written in 8-bit units.

Address: PFCn: <PORTn_base> + 0500_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20, 24)
JPFC0: <JPORT0_base> + 0050_H*[1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFC n_15	PFC n_14	PFC n_13	PFC n_12	PFC n_11	PFC n_10	PFC n_9	PFC n_8	PFC n_7	PFC n_6	PFC n_5	PFC n_4	PFC n_3	PFC n_2	PFC n_1	PFC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.66, Control Registers (P20)**, and **Table 2A.74, Control Registers (P24).**

Table 2A.22 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2A.25, Setting Alternative Functions**.

NOTE

The control bits of the JTAG port function control register (JPFC0) are JPFC0_[7:0].

2A.9.2.8 PFCEn / JPFCE0 — Port Function Control Expansion Register

This register, together with register PFCn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access:	PFCEn: This register can be read or written in 16-bit units. JPFCE0: This register can be read or written in 8-bit units.
Address:	PFCEn: <PORTn_base> + 0600_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20) JPFCE0: <JPORT0_base> + 0060_H*1
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn_11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.62, Control Registers (P18)**, and **Table 2A.66, Control Registers (P20)**.

Table 2A.23 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2A.25, Setting Alternative Functions**.

2A.9.2.9 PFCAEn — Port Function Control Additional Expansion Register

This register selects the alternative peripheral functions together with PFCEn, PFCn registers.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCAEn: This register can be read or written in 16-bit units.

Address: PFCAEn: <PORTn_base> + $0A00_H$ + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 20)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAE n_15	PFCAE n_14	PFCAE n_13	PFCAE n_12	PFCAE n_11	PFCAE n_10	PFCAE n_9	PFCAE n_8	PFCAE n_7	PFCAE n_6	PFCAE n_5	PFCAE n_4	PFCAE n_3	PFCAE n_2	PFCAE n_1	PFCAE n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, and **Table 2A.66, Control Registers (P20)**.

Table 2A.24 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2A.25, Setting Alternative Functions**.

Table 2A.25 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1
			0	Alternative output mode 1
		1	1	Alternative input mode 2
			0	Alternative output mode 2
	1	0	1	Alternative input mode 3
			0	Alternative output mode 3
		1	1	Alternative input mode 4
			0	Alternative output mode 4
1	0	0	1	Alternative input mode 5
			0	Alternative output mode 5
		1	1	Alternative input mode 6
			0	Alternative output mode 6
	1	0	1	Alternative input mode 7
			0	Alternative output mode 7
		1	X	Setting prohibited

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m, or PFCAEn_m bit, set the PMCn_m bit to “1”.
- With this product, the I/O of some functions is assigned to two or more pins, but a specific pin function can only be set to one pin at a time. Setting the same pin function to two or more pins at the same time is prohibited. For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as a pin function other than b.

NOTE

For more details on the assignment of each function, see **Section 2A.10.1.2, Control Registers** to **Section 2A.10.15.2, Control Registers**.

2A.9.3 Pin Data Input/Output

2A.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

This register enables the input buffer in output mode and sets the port to bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

Access:	PBDCn, APBDCn: These registers can be read or written in 16-bit units. JPBDC0: This register can be read or written in 8-bit units.
Address:	PBDCn: <PORTn_base> + 4100_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24) APBDCn: <PORTn_base> + $41C8_H$ + n × 4 (n = 0, 1) JPBDC0: <JPORT0_base> + 0410_H^{*1}
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1)**.

Table 2A.26 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin. 0: Bidirectional mode disabled 1: Bidirectional mode enabled

CAUTION

- When the Pn_m port is used for the alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read from PPRn.PPRn_m by enabling the bidirectional mode (PBDCn.PBDCn_m = 1).
- However, output of that alternative output function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn_m, PFCEn.PFCEn_m, and PFCAEn.PFCAEn_m). If the alternative input function in question is being used by another pin, the alternative input function is not guaranteed.

NOTE

The control bits of the JTAG port bidirection control register (JPBDC0) are JPBDC0_[7:0].

2A.9.3.2 PPRn / APPRn / JPPR0 / IPPR0 — Port Pin Read Register

This register reflects the actual level of the Pn_m pin, whether it is the value of the Pn.Pn_m bit or the level of an alternative output function.

Access: PPRn, APPRn, IPPR0: These registers are read-only registers that can be read in 16-bit units.

JPPR0: This register is a read-only register that can be read in 8-bit units.

Address: PPRn: <PORTn_base> + 0200$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APPRn: <PORTn_base> + 02C8$_H$ + n × 4 (n = 0, 1)

JPPR0: <JPORT0_base> + 0020$_H$

IPPR0: <PORTn_base> + 02F0$_H$*1

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPR n_15	PPR n_14	PPR n_13	PPR n_12	PPR n_11	PPR n_10	PPR n_9	PPR n_8	PPR n_7	PPR n_6	PPR n_5	PPR n_4	PPR n_3	PPR n_2	PPR n_1	PPR n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, **Table 2A.78, Control Registers (AP1)**, and **Table 2A.80, Control Registers (IP0)**.

Table 2A.27 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	The Pn_m Pin, Pn.Pn_m value or alternative function output.

NOTES

1. For the read values of the PPRn register, see **Section 2A.7.4, Pin Data Input/Output**.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].

2A.9.3.3 Pn / APn / JP0 — Port Register

This register holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access: Pn, APn: These registers can be read or written in 16-bit units.

JP0: This register can be read or written in 8-bit units.

Address: Pn: <PORTn_base> + 0000_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APn: <PORTn_base> + $00C8_H$ + n × 4 (n = 0, 1)

JP0: <JPORT0_base> + 0000_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1)**.

Table 2A.28 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

NOTE

The control bits of the JTAG port register (JP0) are JP0_[7:0].

2A.9.3.4 PNOTn / APNOTn / JPNOT0 — Port NOT Register

This register allows the Pn_m bit of the port register Pn to be inverted without directly writing to Pn.

Access: PNOTn, APNOTn: These registers are write-only registers that can be written in 16-bit units. When read, 0000_H is returned.

JPNOT0: This register is a write-only register that can be written in 8-bit units. When read, 00_H is returned.

Address: PNOTn: <PORTn_base> + 0700_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APNOTn: <PORTn_base> + $07C8_H$ + n × 4 (n = 0, 1)

JPNOT0: <JPORT0_base> + 0070_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn_15	PNOTn_14	PNOTn_13	PNOTn_12	PNOTn_11	PNOTn_10	PNOTn_9	PNOTn_8	PNOTn_7	PNOTn_6	PNOTn_5	PNOTn_4	PNOTn_3	PNOTn_2	PNOTn_1	PNOTn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1).**

Table 2A.29 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted. 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted ($\overline{\text{Pn\_m}}$ → Pn_m)

NOTE

The control bits of the JTAG port NOT register are JPNOT0_[7:0].

2A.9.3.5 PSRn / APSRn / JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to the Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn_m is set by the corresponding bit in the lower 16 bits of PSRn.

Access: PSRn, APSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of registers Pn and APn.

JPSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JP0.

Address: PSRn: <PORTn_base> + 0100_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

APSRn: <PORTn_base> + $01C8_H$ + n × 4 (n = 0, 1)

JPSR0: <JPORT0_base> + 0010_H*1

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, **Table 2A.74, Control Registers (P24)**, **Table 2A.76, Control Registers (AP0)**, and **Table 2A.78, Control Registers (AP1)**.

Table 2A.30 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m (PSRn_[15:0]) is written to Pn_m. 0: Pn_m is not affected by PSRn_m 1: Pn_m is PSRn_m Example: If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit (PSRn_[31:16]) PSRn_m is 1. 0: Pn_m = 0 1: Pn_m = 1

NOTE

The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].

2A.9.4 Configuration of Electrical Characteristics

2A.9.4.1 PUn / JPU0 — Pull-Up Option Register

This register specifies whether an internal pull-up resistor is connected to an input pin.

Access: PUn: This register can be read or written in 16-bit units.

JPU0: This register can be read or written in 8-bit units.

Address: PUn: <PORTn_base> + 4300_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11,12, 13, 18, 19, 20, 21, 22, 23, 24)

JPU0: <JPORT0_base> + 0430_H[*1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.31 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an internal pull-up resistor is connected to the corresponding pin. 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The pull-up resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].

2A.9.4.2 PDn / JPD0 — Pull-Down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin.

Access:	PDn: This register can be read or written in 16-bit units.
	JPD0: This register can be read or written in 8-bit units.
Address:	PDn: <PORTn_base> + 4400$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)
	JPD0: <JPORT0_base> + 0440$_H$^{*1}
Value after reset:	0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.32 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin. 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].

2A.9.4.3 PDSCn / JPDSC0 — Port Drive Strength Control Register

This register specifies the output driver strength of the port pin. This function selects the fast mode (high drive strength) or slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMDn register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**. Regarding the alternative functions for which the PDSC register needs to be set, see **Section 2A.11.3.3, Output Buffer Control (PDSC)**.

Access: PDSCn, JPDSC0: These registers can be read or written in 32-bit units.

Address: PDSCn: <PORTn_base> + 4600_H + n × 4 (n = 0, 1, 2, 3, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)
JPDSC0: <JPORT0_base> + 0460_H[*1]

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSC n_15	PDSC n_14	PDSC n_13	PDSC n_12	PDSC n_11	PDSC n_10	PDSC n_9	PDSC n_8	PDSC n_7	PDSC n_6	PDSC n_5	PDSC n_4	PDSC n_3	PDSC n_2	PDSC n_1	PDSC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.33 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Lower drive strength (when the frequency output from the pin is 10 MHz or below) 1: High drive strength (when the frequency output from the pin is 40 MHz or less).

2A.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access:	PODCn, JPODC0: These registers can be read or written in 32-bit units.
Address:	PODCn: <PORTn_base> + 4500_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24) JPODC0: <JPORT0_base> + 0450_H*1
Value after reset:	0000 0000_H*2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0*3	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Note 2. The PODC8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8 register is 0000 0040_H.
When the OPBT0.RESETOUTEN = 0, the PODC8 register is 0000 0000_H.

Note 3. The PODC8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8_6 bit is 1.
When the OPBT0.RESETOUTEN = 0, the PODC8_6 bit is 0.

Table 2A.34 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

NOTE

The control bits of the JTAG port open drain control register (JPODC0) are JPODC0_[31:0].

2A.9.4.5 PISn/JPIS0 — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

Access: PISn: This register can be read or written in 16-bit units.

JPIS0: This register can be read or written in 8-bit units.

Address: PISn: <PORTn_base> + 4700_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24)

JPIS0: <JPORT0_base> + 0470_H^{*1}

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIS n_15	PIS n_14	PIS n_13	PIS n_12	PIS n_11	PIS n_10	PIS n_9	PIS n_8	PIS n_7	PIS n_6	PIS n_5	PIS n_4	PIS n_3	PIS n_2	PIS n_1	PIS n_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.44, Control Registers (P1)**, **Table 2A.46, Control Registers (P2)**, **Table 2A.48, Control Registers (P3)**, **Table 2A.50, Control Registers (P8)**, **Table 2A.52, Control Registers (P9)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.64, Control Registers (P19)**, **Table 2A.66, Control Registers (P20)**, **Table 2A.68, Control Registers (P21)**, **Table 2A.70, Control Registers (P22)**, **Table 2A.72, Control Registers (P23)**, and **Table 2A.74, Control Registers (P24)**.

Table 2A.35 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer Characteristics: 0: Type 1 (SHMT1) 1: Type 2 (SHMT4)

NOTES

1. Details of the definition of type 1 and type 2 are given in **Section 2A.11.3.2, Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)**. For details, also see **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for input buffer characteristics.
2. The control bits of the JTAG port input buffer selection register (JPIS0) are JPIS0_[7:0].

2A.9.4.6 PISAn / JPISA0 — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics.

Access: PISAn: This register can be read or written in 16-bit units.
JPISA0: This register can be read or written in 8-bit units.

Address: PISAn: <PORTn_base> + $4A00_H$ + n × 4 (n = 10, 11, 12, 13, 18, 21, 22)
JPISA0: <JPORT0_base> + $04A0_H$[*1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISAn_15	PISAn_14	PISAn_13	PISAn_12	PISAn_11	PISAn_10	PISAn_9	PISAn_8	PISAn_7	PISAn_6	PISAn_5	PISAn_4	PISAn_3	PISAn_2	PISAn_1	PISAn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2A.10, Port (General I/O) Function Overview**: **Table 2A.40, Control Registers (JP0)**, **Table 2A.42, Control Registers (P0)**, **Table 2A.54, Control Registers (P10)**, **Table 2A.56, Control Registers (P11)**, **Table 2A.58, Control Registers (P12)**, **Table 2A.60, Control Registers (P13)**, **Table 2A.62, Control Registers (P18)**, **Table 2A.68, Control Registers (P21)**, and **Table 2A.70, Control Registers (P22)**.

Table 2A.36 PISAn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISA_[15:0]	Specifies the input buffer characteristics: 0: Type 2 (SHMT4) 1: Type 5 (TTL)

Table 2A.37 Port Input Selection Advanced Register Contents

PISAn	PISn	Function
0	0	Type 1 input buffer is selected (SHMT1)
	1	Type 2 input buffer is selected (SHMT4)
1	X	Type 5 input buffer is selected (TTL)

NOTE

Details of the definition of type 2 and type 5 are given in **Section 2A.11.3.2, Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)**. For details, also see **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for input buffer characteristics.

2A.9.5 Port Register Protection

RH850/F1KH has Port Protection Command Register (PPCMDn) and Port Protection Status Register (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 5, Write-Protected Registers**.

2A.9.6 Flowchart Examples for Port Settings

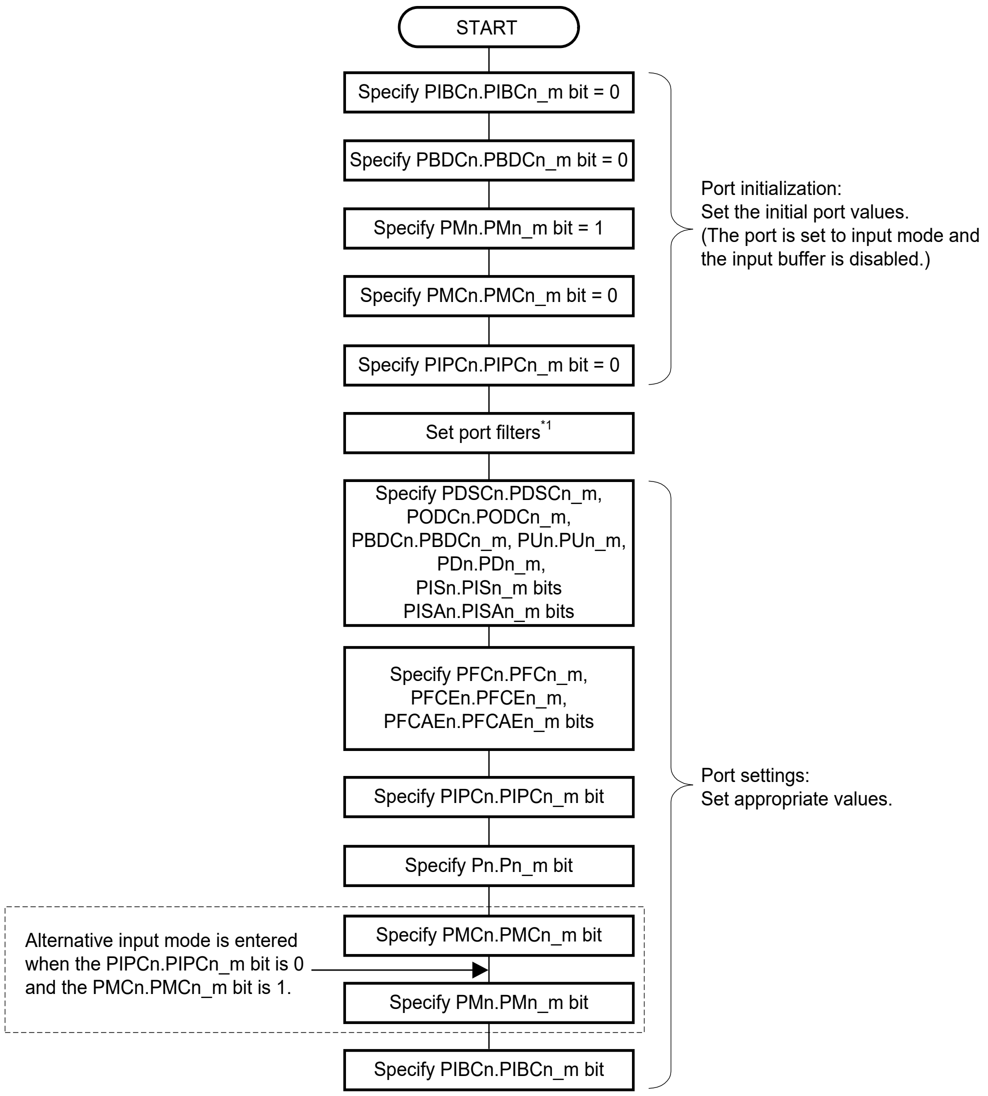
Examples of the port settings are shown in the flowchart below.

CAUTION

If the port is set to the PIPCn.PIPCn_m = 0 and alternative output mode, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCn_m bit is set to 1 and when the PMn.PMn_m bit is set to 0. If an interrupt-related signal is specified as an alternate function of the port, the mode will temporarily become the alternative input mode, so either disable the interrupt in question, or specify that the interrupt is ignored.

2A.9.6.1 Batch Setting

An example of specifying batch port settings is shown in the flowchart below.



Note 1. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-6 and INTP9-23 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2A.6 Example of Port Settings (When Specified in Batch)

2A.9.6.2 Individual Settings

An example of specifying individual port settings is shown in the flowchart below.

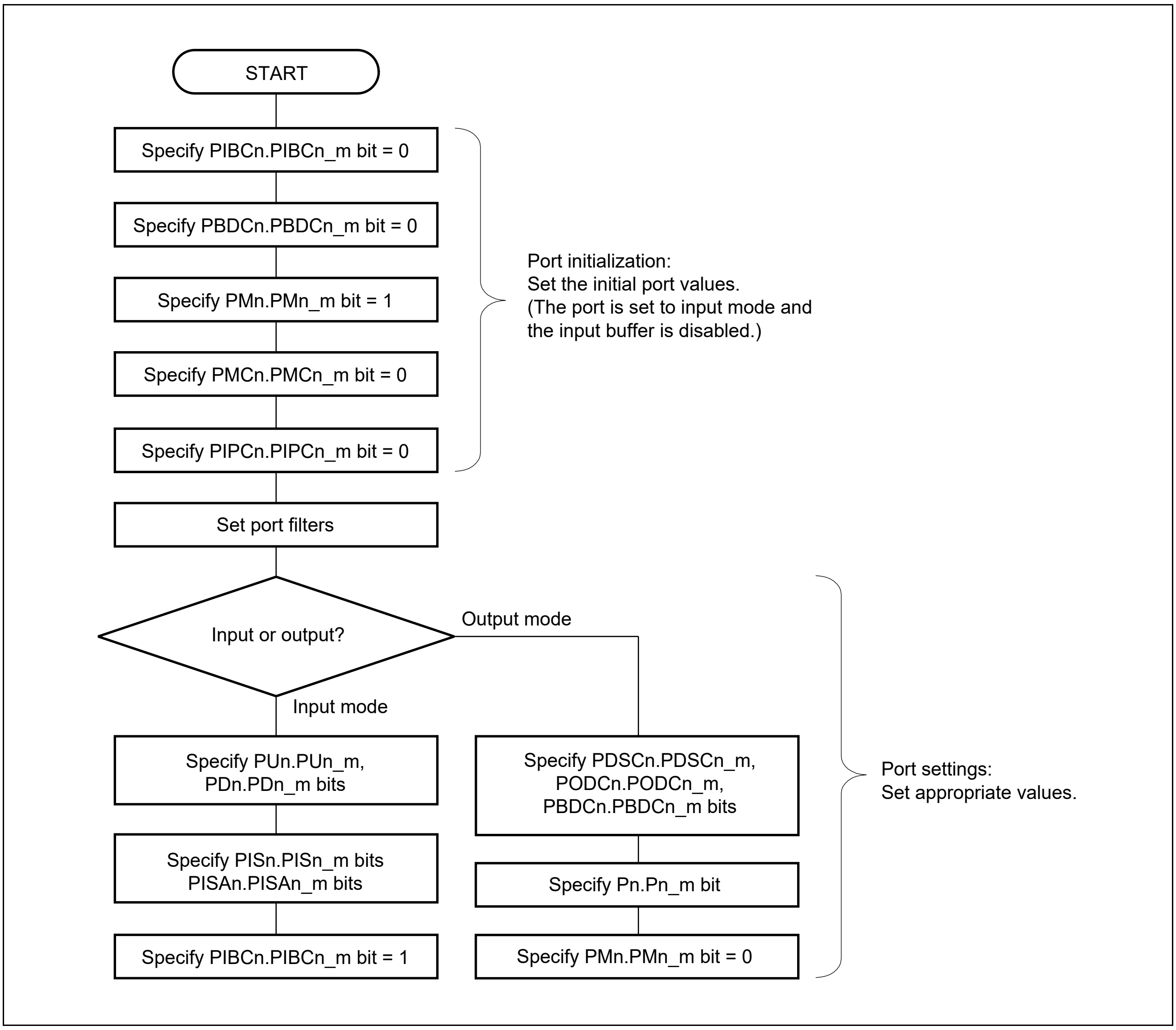


Figure 2A.7 Example of Port Settings (in Port Mode)

(1) With IP Control

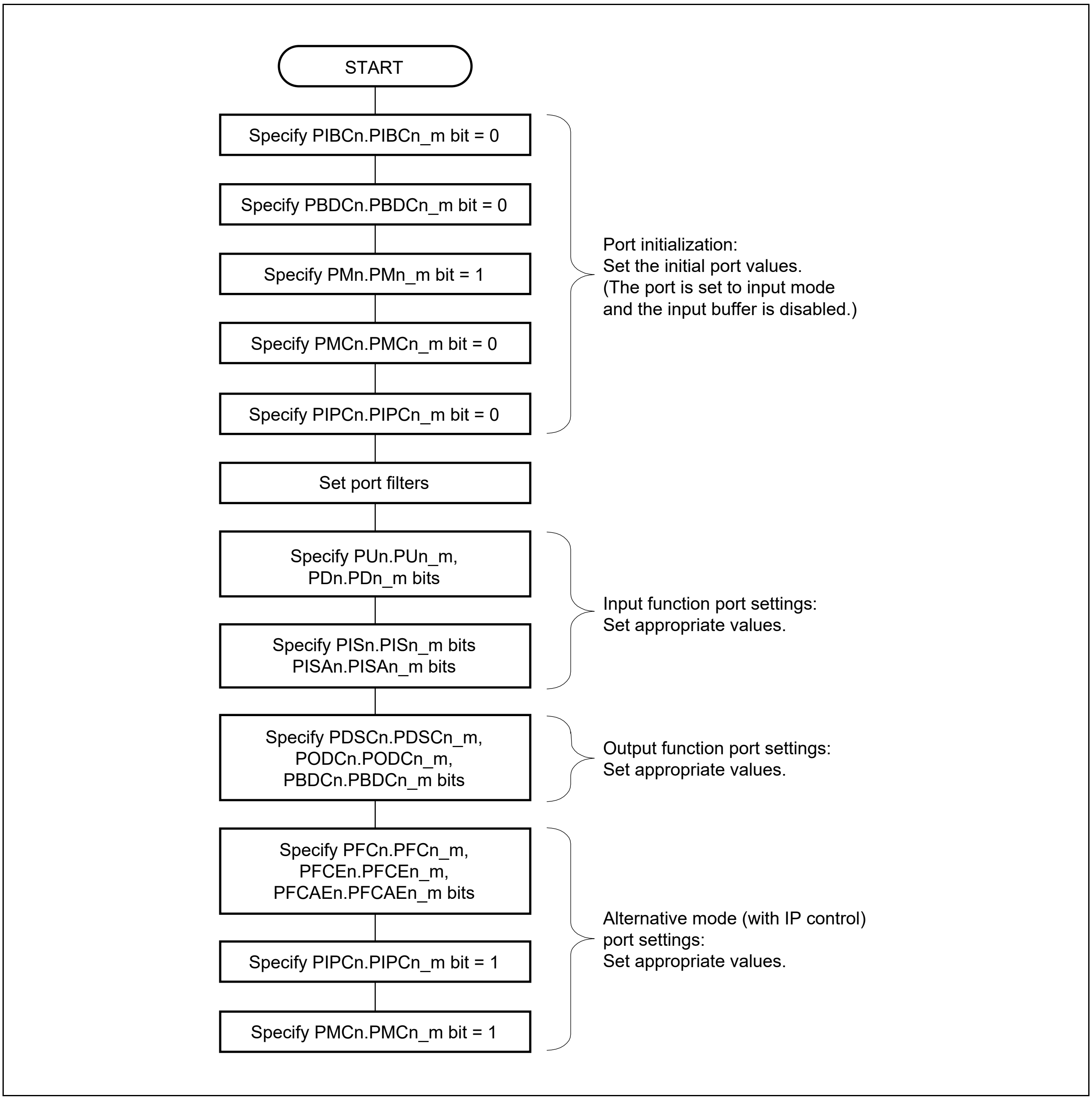
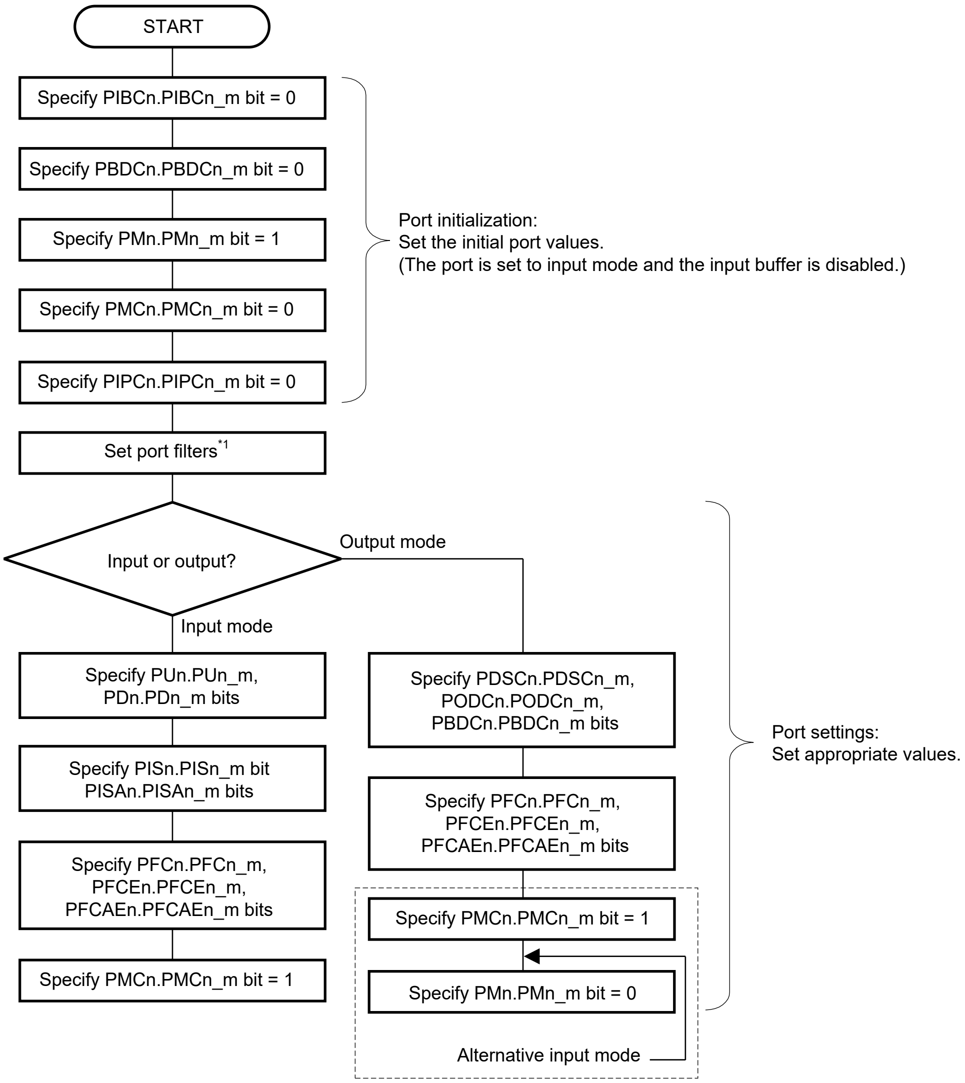


Figure 2A.8 Example of Port Settings (in Alternative Mode)

(2) Without IP Control



Note 1. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-6 and INTP9-23 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2A.9 Example of Port Settings (in Alternative Mode)

2A.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functions and all the functions assigned to the ports. See the following pages for details.

In addition, whether the port mode is alternative mode or not can be selected by PMCn register setting. When PMCn.PMCn_m = 1, alternative functions are selected by the PFCn, PFCEn, and PFCAEn registers.

Table 2A.38 Port Function

Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	Device			
						176 Pins	233 Pins	272 Pins	324 Pins
JTAG Port 0	JP0_0 - 6	7 bits	In/Out	AWO	JTAG, LPD	✓	✓	—	✓
Port 0	P0_0 - 14	15 bits	In/Out	AWO		✓	✓	—	✓
Port 1	P1_0 - 5, 8-15	14 bits	In/Out	AWO		✓	✓	—	✓
Port 2	P2_0 - 6	7 bits	In/Out	AWO		✓	—	—	—
	P2_0 - 15	16 bits				—	✓	—	✓
Port 3	P3_0	1 bit	In/Out	AWO		—	✓	—	—
	P3_0 – 10	11 bits				—	—	—	—
	P3_0 – 12	13 bits				—	—	—	✓
Port 8	P8_0 - 12	13 bits	In/Out	AWO	ADCA0 (10-bit resolution) $\overline{\text{RESETOUT}}$	✓	✓	—	✓
Port 9	P9_0 - 4	5 bits	In/Out	ISO	ADCA0 (10-bit resolution)	✓	✓	—	✓
Port 10	P10_0 - 15	16 bits	In/Out	ISO		✓	✓	—	✓
Port 11	P11_0 – 12, 15	14 bits	In/Out	ISO		✓	✓	—	✓
Port 12	P12_0 - 5	6 bits	In/Out	ISO		✓	✓	—	✓
Port 13	P13_0 - 7	8 bits	In/Out	ISO		—	✓	—	✓
Port 18	P18_0 - 7	8 bits	In/Out	ISO	ADCA1 (10-bit resolution)	✓	—	—	—
	P18_0 - 15	16 bits				—	✓	—	✓
Port 19	P19_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	✓	—	✓
Port 20	P20_0 - 5	6 bits	In/Out	ISO		✓	✓	—	—
	P20_0 - 14	15 bits				—	—	—	—
	P20_0 - 15	16 bits				—	—	—	✓
Port 21	P21_0 – 4	5 bits	In/Out	ISO		—	—	—	—
	P21_0 – 14	15 bits				—	—	—	✓
Port 22	P22_0 - 15	16 bits	In/Out	ISO		—	—	—	✓
Port 23	P23_0 - 10	11 bits	In/Out	ISO		—	—	—	✓
Port 24	P24_0 - 7	8 bits	In/Out	ISO		—	—	—	✓
Analog Port 0	AP0_0 - 15	16 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	✓	✓	—	✓
Analog Port 1	AP1_0 - 15	16 bits	In/Out	ISO	ADCA1 (12/10-bit resolution)	✓	✓	—	✓
Input Port 0	IP0_0	1 bit	In	AWO	SOSC (XT2 pin)	✓	✓	—	✓

2A.10.1 JTAG Port 0 (JP0)

2A.10.1.1 Alternative Function

The following alternative functions are available when JTAG port 0 is configured as a general-purpose I/O port by setting OPJTAG[1:0] on the corresponding option byte to 00_B.

Table 2A.39 JTAG Port 0 (JP0)

Port Mode (JPMC0_m = 0)	Alternative Mode (JPMC0_m = 1)										ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
JP0_0[*1]	INTP0		TAUJ2I0	TAUJ2O0	FPDR	FPDT						DCUTDI/LPDI/LPDIO	47	U4	—	AB4
JP0_1	INTP1		TAUJ0I0	TAUJ0O0		FPDT						DCUTDO/LPDO	46	P5	—	W6
JP0_2	INTP2		TAUJ0I1	TAUJ0O1	FPCK							DCUTCK/LPDCLK	45	T4	—	W5
JP0_3	INTP3	CSCXFOUT	TAUJ0I2	TAUJ0O2								DCUTMS	44	R4	—	Y5
JP0_4												$\overline{\text{DCUTRST}}$	43	R3	—	AB3
JP0_5	NMI	RTCA0OUT	TAUJ0I3	TAUJ0O3								$\overline{\text{DCURDY}}$/LPDCLKOUT	42	U3	—	AA4
JP0_6												$\overline{\text{EVTO}}$	66	U11	—	Y10

Note 1. In LPD (1 pin) mode, the JP0_0 output buffer state is Open-drain.

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2A.10.1.2 Control Registers

Table 2A.40 Control Registers (JP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
JP0	JTAG port register 0	8	6-0	R/W	0000$_H$	00$_H$	✓	✓	—	✓
JPSR0	JTAG port set/reset register 0	32	22-16, 6-0	R/W	0010$_H$	0000 0000$_H$	✓	✓	—	✓
JPPR0	JTAG port pin read register 0	8	6-0	R	0020$_H$	00$_H$	✓	✓	—	✓
JPM0	JTAG port mode register 0	8	6-0	R/W	0030$_H$	FF$_H$	✓	✓	—	✓
JPMC0	JTAG port mode control register 0	8	5, 3-0	R/W	0040$_H$	00$_H$	✓	✓	—	✓
JPFC0	JTAG port function control register 0	8	5, 3-0	R/W	0050$_H$	00$_H$	✓	✓	—	✓
JPFCE0	JTAG port function control expansion register 0	8	2-0	R/W	0060$_H$	00$_H$	✓	✓	—	✓
JPNOT0	JTAG port NOT register 0	8	6-0	W	0070$_H$	00$_H$	✓	✓	—	✓
JPMSR0	JTAG port mode set/reset register 0	32	22-16, 6-0	R/W	0080$_H$	0000 FFFF$_H$	✓	✓	—	✓
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	R/W	0090$_H$	0000 0000$_H$	✓	✓	—	✓
JPIBC0	JTAG port input buffer control register 0	8	6-0	R/W	0400$_H$	00$_H$	✓	✓	—	✓
JPBDC0	JTAG port bidirection control register 0	8	6-0	R/W	0410$_H$	00$_H$	✓	✓	—	✓
JPU0	Pull-up option register 0	8	6-0	R/W	0430$_H$	00$_H$	✓	✓	—	✓
JPD0	Pull-down option register 0	8	6-0	R/W	0440$_H$	00$_H$	✓	✓	—	✓
JPODC0	JTAG port open drain control register 0	32	6-0	R/W	0450$_H$	0000 0000$_H$	✓	✓	—	✓
JPDSC0	JTAG port drive strength control register 0	32	6, 5, 3-1	R/W	0460$_H$	0000 0000$_H$	✓	✓	—	✓
JPIS0	JTAG port input buffer selection register 0	8	6, 5, 3-0	R/W	0470$_H$	FF$_H$	✓	✓	—	✓
JPISA0	JTAG port input buffer selection advanced register 0	8	3, 2, 0	R/W	04A0$_H$	00$_H$	✓	✓	—	✓
JPPROTS0	JTAG port protection status register 0	32	0	R	04B0$_H$	0000 0000$_H$	✓	✓	—	✓
JPPCMD0	JTAG port protection command register 0	32	7-0	W	04C0$_H$	xxxx xx00$_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.2 Port 0 (P0)

2A.10.2.1 Alternative Function

Table 2A.41 Port 0 (P0)

Port Mode (PMC0_m = 0)	Alternative Mode (PMC0_m =1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P0_0	TAUD0I2	TAUD0O2	RLIN20RX	CAN0TX		PWGA10O	$\overline{\text{CSIH0SSI}}$	DPO	TAUJ2I1	TAUJ2O1							18	J1	—	T1
P0_1	TAUD0I4	TAUD0O4	CAN0RX/ INTP0	RLIN20TX	INTP0	PWGA11O	CSIH0SI	APO	TAUJ2I2	TAUJ2O2			CAN0RX				19	J2	—	P3
P0_2	TAUD0I6	TAUD0O6	CAN1RX/ INTP1	RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO	TAUJ2I3	TAUJ2O3	CAN1RX				20	J4	—	P4
P0_3	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10	CAN1TX	DPIN1	PWGA13O		CSIH0SO	INTP10		TAUJ1I0	TAUJ1O0	RLIN30RX				21	K1	—	R2
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O	CSIH1SI	SELDP0	DPIN8		TAUB0I12	TAUB0O12			RLIN31RX				23	K3	—	R4
P0_5	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO			TAUB0I14	TAUB0O14			CAN2RX				24	K2	—	U1
P0_6	INTP2		DPIN10	SELDP2	CSIH1SC					PWGA35O							25	L3	—	R3
P0_7	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	CAN3RX/ INTP3				CAN3RX				70	R11	—	Y12
P0_8	INTP16	RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX							69	T12	—	Y11
P0_9	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	CAN4RX/ INTP4				CAN4RX				68	R10	—	W11
P0_10	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6		CAN4TX							67	T11	—	AB12
P0_11	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8	RLIN26RX	PWGA34O									26	L1	—	T2
P0_12	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI	RLIN26TX									27	L2	—	U2
P0_13	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	CAN5RX/ INTP5		RLIN32RX		CAN5RX				28	M1	—	T3
P0_14	INTP17	RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX							29	L4	—	U3

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.2.2 Control Registers

Table 2A.42 Control Registers (P0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P0	Port register 0	16	14-0	R/W	0000_H	0000_H	✓	✓	—	✓
PSR0	Port set/reset register 0	32	30-16, 14-0	R/W	0100_H	$0000\ 0000_H$	✓	✓	—	✓
PPR0	Port pin read register 0	16	14-0	R	0200_H	0000_H	✓	✓	—	✓
PM0	Port mode register 0	16	14-0	R/W	0300_H	$FFFF_H$	✓	✓	—	✓
PMC0	Port mode control register 0	16	14-0	R/W	0400_H	0000_H	✓	✓	—	✓
PFC0	Port function control register 0	16	14-0	R/W	0500_H	0000_H	✓	✓	—	✓
PFCE0	Port function control expansion register 0	16	14-0	R/W	0600_H	0000_H	✓	✓	—	✓
PNOT0	Port NOT register 0	16	14-0	W	0700_H	0000_H	✓	✓	—	✓
PMSR0	Port mode set/reset register 0	32	30-16, 14-0	R/W	0800_H	$0000\ FFFF_H$	✓	✓	—	✓
PMCSR0	Port mode control set/reset register 0	32	30-16, 14-0	R/W	0900_H	$0000\ 0000_H$	✓	✓	—	✓
PFCAE0	Port function control additional expansion register 0	16	14, 13, 10-0	R/W	$0A00_H$	0000_H	✓	✓	—	✓
PIBC0	Port input buffer control register 0	16	14-0	R/W	4000_H	0000_H	✓	✓	—	✓
PBDC0	Port bidirection control register 0	16	14-0	R/W	4100_H	0000_H	✓	✓	—	✓
PIPC0	Port IP control register 0	16	14, 13, 6, 5, 3, 2	R/W	4200_H	0000_H	✓	✓	—	✓
PU0	Pull-up option register 0	16	14-0	R/W	4300_H	0000_H	✓	✓	—	✓
PD0	Pull-down option register 0	16	14-0	R/W	4400_H	0000_H	✓	✓	—	✓
PODC0	Port open drain control register 0	32	14-0	R/W	4500_H	$0000\ 0000_H$	✓	✓	—	✓
PDSC0	Port drive strength control register 0	32	14-0	R/W	4600_H	$0000\ 0000_H$	✓	✓	—	✓
PIS0	Port input buffer selection register 0	16	14-0	R/W	4700_H	$FFFF_H$	✓	✓	—	✓
PPROTS0	Port protection status register 0	32	0	R	$4B00_H$	$0000\ 0000_H$	✓	✓	—	✓
PPCMD0	Port protection command register 0	32	7-0	W	$4C00_H$	$XXXX\ XX00_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.3 Port 1 (P1)

2A.10.3.1 Alternative Function

Table 2A.43 Port 1 (P1)

Port Mode (PMC1_m = 0)	Alternative Mode (PMC1_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P1_0	RLIN33RX/ INTP13		INTP13				TAUJ2I0	TAUJ2O0	$\overline{\text{CSIG4SSI}}$				RLIN33RX				30	M2	—	V1
P1_1	INTP18	RLIN33TX			CSIG4SC		TAUJ2I1	TAUJ2O1									31	N2	—	U4
P1_2	CAN3RX/ INTP3		INTP3		DPIN19		TAUJ2I2	TAUJ2O2	CSIG4SI				CAN3RX				32	N1	—	W1
P1_3	INTP19	CAN3TX	DPIN23			CSIG4SO	TAUJ2I3	TAUJ2O3									33	N3	—	V3
P1_4	RLIN35RX/ INTP15		INTP15		DPIN18		CSIH4SI						RLIN35RX				75	U13	—	AA13
P1_5	ADCA1TRG0	RLIN35TX	DPIN17		INTP20		CSIH4SC										74	U12	—	AA12
P1_8																	53	U5	—	AB6
P1_9			DPIN20		INTP21												52	R7	—	W7
P1_10	RLIN24RX		DPIN21		INTP22		ADCA1TRG1										51	R6	—	AB5
P1_11	ADCA1TRG2	RLIN24TX	DPIN22		INTP14												50	P6	—	AA6
P1_12	CAN4RX/ INTP4		INTP4	RLIN36TX									CAN4RX				34	P1	—	V2
P1_13		CAN4TX	RLIN36RX/ INTP16										RLIN36RX				35	P2	—	V4
P1_14	RLIN23RX		CAN7RX/ INTP9		CSIH4RYI	CSIH4RYO							CAN7RX				78	U14	—	Y13
P1_15		RLIN23TX		CAN7TX													79	R13	—	W14

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.3.2 Control Registers

Table 2A.44 Control Registers (P1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P1	Port register 1	16	15-8, 5-0	R/W	0004H	0000H	✓	✓	—	✓
PSR1	Port set/reset register 1	32	31-24, 21-16, 15-8, 5-0	R/W	0104H	0000 0000H	✓	✓	—	✓
PPR1	Port pin read register 1	16	15-8, 5-0	R	0204H	0000H	✓	✓	—	✓
PM1	Port mode register 1	16	15-8, 5-0	R/W	0304H	FFFFH	✓	✓	—	✓
PMC1	Port mode control register 1	16	15-9, 5-0	R/W	0404H	0000H	✓	✓	—	✓
PFC1	Port function control register 1	16	15-9, 5-0	R/W	0504H	0000H	✓	✓	—	✓
PFCE1	Port function control expansion register 1	16	14-9, 5-0	R/W	0604H	0000H	✓	✓	—	✓
PNOT1	Port NOT register 1	16	15-8, 5-0	W	0704H	0000H	✓	✓	—	✓
PMSR1	Port mode set/reset register 1	32	31-24, 21-16, 15-8, 5-0	R/W	0804H	0000 FFFFH	✓	✓	—	✓
PMCSR1	Port mode control set/reset register 1	32	31-25, 21-16, 15-9, 5-0	R/W	0904H	0000 0000H	✓	✓	—	✓
PFCAE1	Port function control additional expansion register 1	16	14-12, 4, 2, 0	R/W	0A04H	0000H	✓	✓	—	✓
PIBC1	Port input buffer control register 1	16	15-8, 5-0	R/W	4004H	0000H	✓	✓	—	✓
PBDC1	Port bidirection control register 1	16	15-8, 5-0	R/W	4104H	0000H	✓	✓	—	✓
PIPC1	Port IP control register 1	16	5,3,1	R/W	4204H	0000H	✓	✓	—	✓
PU1	Pull-up option register 1	16	15-8, 5-0	R/W	4304H	0000H	✓	✓	—	✓
PD1	Pull-down option register 1	16	15-8, 5-0	R/W	4404H	0000H	✓	✓	—	✓
PODC1	Port open drain control register 1	32	15-8, 5-0	R/W	4504H	0000 0000H	✓	✓	—	✓
PDSC1	Port drive strength control register 1	32	15-8, 5-0	R/W	4604H	0000 0000H	✓	✓	—	✓
PIS1	Port input buffer selection register 1	16	15-8, 5-0	R/W	4704H	FFFFH	✓	✓	—	✓
PPROTS1	Port protection status register 1	32	0	R	4B04H	0000 0000H	✓	✓	—	✓
PPCMD1	Port protection command register 1	32	7-0	W	4C04H	xxxx xx00H	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.4 Port 2 (P2)

2A.10.4.1 Alternative Function

Table 2A.45 Port 2 (P2)

Port Mode (PMC2_m =0)	Alternative Mode (PMC2_m =1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P2_0	RLIN27RX		CAN6RX/ INTP6										CAN6RX				49	T5	—	AA5
P2_1		RLIN27TX		CAN6TX													48	R5	—	Y6
P2_2	RLIN28RX	CSIH4CSS0															65	T10	—	AA11
P2_3		RLIN28TX		CSIH4CSS1													64	U10	—	AA10
P2_4	RLIN29RX			ADCA0SEL0		CSIH4SO											76	T13	—	AB13
P2_5		RLIN29TX	$\overline{\text{CSIH4SSI}}$	ADCA0SEL1													77	R12	—	AB14
P2_6		ADCA0SEL2	CSIG4RYI	CSIG4RYO													36	R1	—	Y1
P2_7	RLIN210RX																—	M4	—	W2
P2_8		RLIN210TX															—	T1	—	W3
P2_9		PWGA77O															—	M3	—	Y3
P2_10		PWGA78O															—	R2	—	AA1
P2_11		PWGA79O															—	N4	—	W4
P2_12	RLIN211RX																—	T2	—	Y2
P2_13		RLIN211TX															—	P7	—	Y7
P2_14		PWGA74O															—	T6	—	AA7
P2_15		PWGA75O															—	P8	—	AA9

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.4.2 Control Registers

Table 2A.46 Control Registers (P2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P2	Port register 2	16	6-0	R/W	0008$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PSR2	Port set/reset register 2	32	22-16, 6-0	R/W	0108$_H$	0000 0000$_H$	✓	—	—	—
			31-16, 15-0				—	✓	—	✓
PPR2	Port pin read register 2	16	6-0	R	0208$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PM2	Port mode register 2	16	6-0	R/W	0308$_H$	FFFF$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PMC2	Port mode control register 2	16	6-0	R/W	0408$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PFC2	Port function control register 2	16	6-3, 1, 0	R/W	0508$_H$	0000$_H$	✓	✓	—	✓
PFCE2	Port function control expansion register 2	16	4, 0	R/W	0608$_H$	0000$_H$	✓	✓	—	✓
PNOT2	Port NOT register 2	16	6-0	W	0708$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PMSR2	Port mode set/reset register 2	32	22-16, 6-0	R/W	0808$_H$	0000 FFFF$_H$	✓	—	—	—
			31-16, 15-0				—	✓	—	✓
PMCSR2	Port mode control set/reset register 2	32	22-16, 6-0	R/W	0908$_H$	0000 0000$_H$	✓	—	—	—
			31-16, 15-0				—	✓	—	✓
PFCAE2	Port function control additional expansion register 2	16	0	R/W	0A08$_H$	0000$_H$	✓	✓	—	✓
PIBC2	Port input buffer control register 2	16	6-0	R/W	4008$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PBDC2	Port bidirection control register 2	16	6-0	R/W	4108$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PIPC2	Port IP control register 2	16	4	R/W	4208$_H$	0000$_H$	✓	✓	—	✓
PU2	Pull-up option register 2	16	6-0	R/W	4308$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PD2	Pull-down option register 2	16	6-0	R/W	4408$_H$	0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PODC2	Port open drain control register 2	32	6-0	R/W	4508$_H$	0000 0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PDSC2	Port drive strength control register 2	32	6-0	R/W	4608$_H$	0000 0000$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PIS2	Port input buffer selection register 2	16	6-0	R/W	4708$_H$	FFFF$_H$	✓	—	—	—
			15-0				—	✓	—	✓
PPROTS2	Port protection status register 2	32	0	R	4B08$_H$	0000 0000$_H$	✓	✓	—	✓
PPCMD2	Port protection command register 2	32	7-0	W	4C08$_H$	xxxx xx00$_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.5 Port 3 (P3)

2A.10.5.1 Alternative Function

Table 2A.47 Port 3 (P3)

Port Mode (PMC3_m =0)	Alternative Mode (PMC3_m =1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P3_0		PWGA76O															—	R8	—	Y8
P3_1		PWGA80O															—	—	—	Y16
P3_2		PWGA81O															—	—	—	AB16
P3_3	CAN8RX/ INTP18	PWGA82O	CAN8RX														—	—	—	Y17
P3_4		PWGA83O		CAN8TX													—	—	—	AA19
P3_5	CAN9RX/ INTP19	PWGA84O	CAN9RX														—	—	—	AB20
P3_6		PWGA85O		CAN9TX													—	—	—	Y18
P3_7	CAN10RX/ INTP20	PWGA86O	CAN10RX														—	—	—	W17
P3_8		PWGA87O		CAN10TX													—	—	—	AA20
P3_9	CAN11RX/ INTP21		CAN11RX														—	—	—	Y19
P3_10		CAN11TX															—	—	—	W18
P3_11		CSIH4CSS0															—	—	—	AB17
P3_12		CSIH4CSS1															—	—	—	AA17

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.5.2 Control Registers

Table 2A.48 Control Registers (P3)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W*^{1}	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P3	Port register 3	16	0	R/W	000C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PSR3	Port set/reset register 3	32	16, 0	R/W	010C$_H$	0000 0000$_H$	—	✓	—	—
			26-16, 10-0				—	—	—	—
			28-16, 12-0				—	—	—	✓
PPR3	Port pin read register 3	16	0	R	020C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PM3	Port mode register 3	16	0	R/W	030C$_H$	FFFF$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PMC3	Port mode control register 3	16	0	R/W	040C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PFC3	Port function control register 3	16	9-3	R/W	050C$_H$	0000$_H$	—	—	—	✓
PNOT3	Port NOT register 3	16	0	W	070C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PMSR3	Port mode set/reset register 3	32	16, 0	R/W	080C$_H$	0000 FFFF$_H$	—	✓	—	—
			26-16, 10-0				—	—	—	—
			28-16, 12-0				—	—	—	✓
PMCSR3	Port mode control set/reset register 3	32	16, 0	R/W	090C$_H$	0000 0000$_H$	—	✓	—	—
			26-16, 10-0				—	—	—	—
			28-16, 12-0				—	—	—	✓
PIBC3	Port input buffer control register 3	16	0	R/W	400C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PBDC3	Port bidirection control register 3	16	0	R/W	410C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PU3	Pull-up option register 3	16	0	R/W	430C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PD3	Pull-down option register 3	16	0	R/W	440C$_H$	0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PODC3	Port open drain control register 3	32	0	R/W	450C$_H$	0000 0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PDSC3	Port drive strength control register 3	32	0	R/W	460C$_H$	0000 0000$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PIS3	Port input buffer selection register 3	16	0	R/W	470C$_H$	FFFF$_H$	—	✓	—	—
			10-0				—	—	—	—
			12-0				—	—	—	✓
PPROTS3	Port protection status register 3	32	0	R	4B0C$_H$	0000 0000$_H$	—	✓	—	✓
PPCMD3	Port protection command register 3	32	7-0	W	4C0C$_H$	0000 0000$_H$	—	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.6 Port 8 (P8)

2A.10.6.1 Alternative Function

Table 2A.49 Port 8 (P8)

Port Mode (PMC8_m = 0)	Alternative Mode (PMC8_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P8_0	TAUJ0I0	TAUJ0O0	DPIN2	PWGA14O	INTP4	CSIH0CSS0	CAN6RX/ INTP6		RIIC1SDA		SENT0RX		CAN6RX		ADCA0I0S		80	U15	—	AA14
P8_1	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5	CSIH1CSS3		CAN6TX	RIIC1SCL			SENT0SPCO			ADCA0I1S		81	T14	—	Y14
P8_2	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O		RLIN37TX							ADCA0I4S		38	U2	—	AA2
P8_3	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1	INTP7	PWGA23O		CAN7TX							ADCA0I5S		82	U16	—	AB15
P8_4	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8	PWGA36O	CAN7RX/ INTP9						CAN7RX		ADCA0I6S		83	R14	—	AA15
P8_5	TAUJ0I3	TAUJ0O3	NMI	CSIH0CSS3	INTP9	PWGA37O									ADCA0I7S		84	T15	—	Y15
P8_6	NMI	CSIH0CSS4		PWGA38O		RTCA0OUT									ADCA0I8S	$\overline{\text{RESETOUT}}$	85	P13	—	AB18
P8_7		CSIH3CSS0		PWGA39O		ADCA0SEL0		RTCA0OUT							ADCA0I14S		86	R15	—	AA16
P8_8		CSIH3CSS1		PWGA40O		ADCA0SEL1	RLIN34RX/ INTP14						RLIN34RX		ADCA0I15S		87	P14	—	AA18
P8_9		CSIH3CSS2		PWGA41O		ADCA0SEL2		RLIN34TX							ADCA0I16S		88	T16	—	AB19
P8_10		CSIH3CSS3	DPIN14	PWGA42O			RLIN37RX/ INTP17						RLIN37RX		ADCA0I17S		39	P3	—	AA3
P8_11	TAUJ1I2	TAUJ1O2	DPIN15	PWGA43O		CSIH1CSS4	RLIN25RX								ADCA0I18S		40	T3	—	AB2
P8_12	TAUJ1I3	TAUJ1O3	DPIN16	PWGA44O		CSIH1CSS5	INTP23	RLIN25TX							ADCA0I19S		41	P4	—	Y4

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2A.50, Control Registers (P8)**.
3. When the $\overline{\text{RESETOUT}}$ function is selected for the P8_6 pin, the output on the pin is at the low level during a reset and after release from the reset state. For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

2A.10.6.2 Control Registers

Table 2A.50 Control Registers (P8)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P8	Port register 8	16	12-0	R/W	0020$_H$	0000$_H$	✓	✓	—	✓
PSR8	Port set/reset register 8	32	28-16, 12-0	R/W	0120$_H$	0000 0000$_H$	✓	✓	—	✓
PPR8	Port pin read register 8	16	12-0	R	0220$_H$	0000$_H$	✓	✓	—	✓
PM8	Port mode register 8	16	12-0	R/W	0320$_H$	FFBF$_H$	✓	✓	—	✓
PMC8	Port mode control register 8	16	12-0	R/W	0420$_H$	0000$_H$	✓	✓	—	✓
PFC8	Port function control register 8	16	12-0	R/W	0520$_H$	0000$_H$	✓	✓	—	✓
PFCE8	Port function control expansion register 8	16	12-0	R/W	0620$_H$	0000$_H$	✓	✓	—	✓
PNOT8	Port NOT register 8	16	12-0	W	0720$_H$	0000$_H$	✓	✓	—	✓
PMSR8	Port mode set/reset register 8	32	28-16, 12-0	R/W	0820$_H$	0000 FFBF$_H$	✓	✓	—	✓
PMCSR8	Port mode control set/reset register 8	32	28-16, 12-0	R/W	0920$_H$	0000 0000$_H$	✓	✓	—	✓
PFCAE8	Port function control additional expansion register 8	16	10, 8, 4, 1, 0	R/W	0A20$_H$	0000$_H$	✓	✓	—	✓
PIBC8	Port input buffer control register 8	16	12-0	R/W	4020$_H$	0000$_H$	✓	✓	—	✓
PBDC8	Port bidirection control register 8	16	12-0	R/W	4120$_H$	0000$_H$	✓	✓	—	✓
PU8	Pull-up option register 8	16	12-0	R/W	4320$_H$	0000$_H$	✓	✓	—	✓
PD8	Pull-down option register 8	16	12-0	R/W	4420$_H$	0000$_H$	✓	✓	—	✓
PODC8	Port open drain control register 8	32	12-0	R/W	4520$_H$	0000 0040$_H$	✓	✓	—	✓
PIS8	Port input buffer selection register 8	32	12-0	R/W	4720$_H$	FFFF$_H$	✓	✓	—	✓
PPROTS8	Port protection status register 8	32	0	R	4B20$_H$	0000 0000$_H$	✓	✓	—	✓
PPCMD8	Port protection command register 8	32	7-0	W	4C20$_H$	xxxx xx00$_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

CAUTION

P8_6 drives a low level after any kind of reset release, until it is later configured differently by register settings. For details, see **Section 2A.11.1.1, P8_6: $\overline{\textbf{RESETOUT}}$** .

2A.10.7 Port 9 (P9)

2A.10.7.1 Alternative Function

Table 2A.51 Port 9 (P9)

Port Mode (PMC9_m = 0)	Alternative Mode (PMC9_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P9_0	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0	CSIH2CSS0	KR0I4		TAUJ1I1	TAUJ1O1	SENT1RX		RIIC1SDA		ADCA0I2S		108	K16	—	T20
P9_1	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5	CSIH2CSS1			TAUJ1I2	TAUJ1O2		SENT1SPCO	RIIC1SCL		ADCA0I3S		109	K17	—	R20
P9_2	KR0I6	PWGA20O	TAPA0ESO	CSIH2CSS2											ADCA0I9S		110	J17	—	R19
P9_3	KR0I7	PWGA21O		CSIH2CSS3	TAUJ1I1	TAUJ1O1	INTP16								ADCA0I10S		111	J15	—	P19
P9_4		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0	INTP17								ADCA0I11S		112	J16	—	N19

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2A.52, Control Registers (P9)**.

2A.10.7.2 Control Registers

Table 2A.52 Control Registers (P9)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W*1	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P9	Port register 9	16	4-0	R/W	0024_H	0000_H	✓	✓	—	✓
PSR9	Port set/reset register 9	32	20-16, 4-0	R/W	0124_H	0000 0000_H	✓	✓	—	✓
PPR9	Port pin read register 9	16	4-0	R	0224_H	0000_H	✓	✓	—	✓
PM9	Port mode register 9	16	4-0	R/W	0324_H	$FFFF_H$	✓	✓	—	✓
PMC9	Port mode control register 9	16	4-0	R/W	0424_H	0000_H	✓	✓	—	✓
PFC9	Port function control register 9	16	4-0	R/W	0524_H	0000_H	✓	✓	—	✓
PFCE9	Port function control expansion register 9	16	4, 3, 1, 0	R/W	0624_H	0000_H	✓	✓	—	✓
PNOT9	Port NOT register 9	16	4-0	W	0724_H	0000_H	✓	✓	—	✓
PMSR9	Port mode set/reset register 9	32	20-16, 4-0	R/W	0824_H	0000 $FFFF_H$	✓	✓	—	✓
PMCSR9	Port mode control set/reset register 9	32	20-16, 4-0	R/W	0924_H	0000 0000_H	✓	✓	—	✓
PFCAE9	Port function control additional expansion register 9	16	1, 0	R/W	$0A24_H$	0000_H	✓	✓	—	✓
PIBC9	Port input buffer control register 9	16	4-0	R/W	4024_H	0000_H	✓	✓	—	✓
PBDC9	Port bidirection control register 9	16	4-0	R/W	4124_H	0000_H	✓	✓	—	✓
PU9	Pull-up option register 9	16	4-0	R/W	4324_H	0000_H	✓	✓	—	✓
PD9	Pull-down option register 9	16	4-0	R/W	4424_H	0000_H	✓	✓	—	✓
PODC9	Port open drain control register 9	32	4-0	R/W	4524_H	0000 0000_H	✓	✓	—	✓
PIS9	Port input buffer selection register 9	16	4-0	R/W	4724_H	$FFFF_H$	✓	✓	—	✓
PPROTS9	Port protection status register 9	32	0	R	$4B24_H$	0000 0000_H	✓	✓	—	✓
PPCMD9	Port protection command register 9	32	7-0	W	$4C24_H$	xxxx $xx00_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.8 Port 10 (P10)

2A.10.8.1 Alternative Function

Table 2A.53 Port 10 (P10)

Port Mode (PMC10_ m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative							
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output			176 Pins	233 Pins	272 Pins	324 Pins
P10_0	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O	TAUJ1I3	TAPA0UP	CSIH1SI	MEMC0A19	ETNB0RXCLK	TAUJ1O3	CAN0RX				174	A2	—	D7
P10_1	TAUD0I3	TAUD0O3	INTP18	CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN	CSIH1SC		ETNB0RXD0	MEMC0A20		TAUJ3O0		MODE0	175	B2	—	D6
P10_2	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO	ETNB0RXD1	MEMC0A21		RLIN37TX		MODE1	176	C3	—	C5
P10_3	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$	MEMC0CLK	RLIN37RX/ INTP17		RLIN37RX				1	B1	—	C4
P10_4	TAUD0I9	TAUD0O9	RLIN21RX	CAN6TX	KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$	PWGA53O	ETNB0RXD2	MEMC0A22					2	D3	—	B2
P10_5	TAUD0I11	TAUD0O11	CAN6RX/ INTP6	RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO	ETNB0RXD3	PWGA54O	CAN6RX				3	C2	—	C3
P10_6	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX /INTP1		MEMC0AD0		RLIN24RX		CAN1RX			MODE2	152	A9	—	D14
P10_7	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX	MEMC0AD1			RLIN24TX	TAUJ3I1	TAUJ3O1			153	A8	—	C14
P10_8	TAUD0I10	TAUD0O10	CSIG0SI	FLXA0TXDB	ENCA0EC	PWGA5O	MEMC0AD2						TAUJ3I2	TAUJ3O2		FLMD1	154	D8	—	B14
P10_9	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO	MEMC0AD3		FLXA0RXDB		RLIN30RX				155	B8	—	C13
P10_10	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1	MEMC0AD4				TAUJ3I3	TAUJ3O3			156	A7	—	A14
P10_11		PWGA16O	RLIN31RX/ INTP11	FLXA0TXENA		CSIH1CSS0	TAUB0I1	TAUB0O1	MEMC0AD5				RLIN31RX				157	C8	—	B13
P10_12		PWGA17O	FLXA0STPWT	RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3	MEMC0AD6								158	D7	—	A13
P10_13	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12	FLXA0TXENB			TAUB0I5	TAUB0O5	MEMC0AD7			CAN7TX	RLIN32RX				159	A6	—	C12
P10_14	ADCA1TRG0	PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7	MEMC0AD8		CAN7RX/ INTP9		CAN7RX				160	B7	—	B12
P10_15	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9		$\overline{\text{MEMC0RD}}$							6	C1	—	D1

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.8.2 Control Registers

Table 2A.54 Control Registers (P10)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P10	Port register 10	16	15-0	R/W	0028_H	0000_H	✓	✓	—	✓
PSR10	Port set/reset register 10	32	31-16, 15-0	R/W	0128_H	$0000\ 0000_H$	✓	✓	—	✓
PPR10	Port pin read register 10	16	15-0	R	0228_H	0000_H	✓	✓	—	✓
PM10	Port mode register 10	16	15-0	R/W	0328_H	$FFFF_H$	✓	✓	—	✓
PMC10	Port mode control register 10	16	15-0	R/W	0428_H	0000_H	✓	✓	—	✓
PFC10	Port function control register 10	16	15-0	R/W	0528_H	0000_H	✓	✓	—	✓
PFCE10	Port function control expansion register 10	16	15-0	R/W	0628_H	0000_H	✓	✓	—	✓
PNOT10	Port NOT register 10	16	15-0	W	0728_H	0000_H	✓	✓	—	✓
PMSR10	Port mode set/reset register 10	32	31-16, 15-0	R/W	0828_H	$0000\ FFFF_H$	✓	✓	—	✓
PMCSR10	Port mode control set/reset register 10	32	31-16, 15-0	R/W	0928_H	$0000\ 0000_H$	✓	✓	—	✓
PFCAE10	Port function control additional expansion register 10	16	15-0	R/W	$0A28_H$	0000_H	✓	✓	—	✓
PIBC10	Port input buffer control register 10	16	15-0	R/W	4028_H	0000_H	✓	✓	—	✓
PBDC10	Port bidirection control register 10	16	15-0	R/W	4128_H	0000_H	✓	✓	—	✓
PIPC10	Port IP control register 10	16	14-0	R/W	4228_H	0000_H	✓	✓	—	✓
PU10	Pull-up option register 10	16	15-0	R/W	4328_H	0000_H	✓	✓	—	✓
PD10	Pull-down option register 10	16	15-0	R/W	4428_H	0000_H	✓	✓	—	✓
PODC10	Port open drain control register 10	32	15-0	R/W	4528_H	$0000\ 0000_H$	✓	✓	—	✓
PDSC10	Port drive strength control register 10	32	15-0	R/W	4628_H	$0000\ 0000_H$	✓	✓	—	✓
PIS10	Port input buffer selection register 10	16	15-0	R/W	4728_H	$FFFF_H$	✓	✓	—	✓
PISA10	Port input buffer selection advanced register 10	16	5, 4, 2-0	R/W	$4A28_H$	0000_H	✓	✓	—	✓
PPROTS10	Port protection status register 10	32	0	R	$4B28_H$	$0000\ 0000_H$	✓	✓	—	✓
PPCMD10	Port protection command register 10	32	7-0	W	$4C28_H$	$xxxx\ xx00_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.9 Port 11 (P11)

2A.10.9.1 Alternative Function

Table 2A.55 Port 11 (P11)

Port mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P11_0	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11		$\overline{\text{MEMC0WR}}$							7	D2	—	E3
P11_1	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7	INTP20	PWGA26O	TAUB0I13	TAUB0O13	MEMC0AD9								161	A5	—	A12
P11_2		CSIH2SO	RLIN32RX/INTP12	RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15	MEMC0AD10		SFMA0IO3		RLIN32RX				162	C7	—	D11
P11_3	CSIH2SC		CAN3RX/INTP3	PWGA28O	TAUB1I1	TAUB1O1	MEMC0AD11			RLIN32TX	SFMA0IO2		CAN3RX				163	B6	—	C11
P11_4	CSIH2SI	CAN3TX	INTP21	PWGA29O	TAUB1I3	TAUB1O3	MEMC0AD12				SFMA0IO1						164	B5	—	A10
P11_5	CAN5RX/INTP5	RLIN33TX		PWGA30O	CSIH3SI		TAUB1I5	TAUB1O5	MEMC0AD13		SFMA0IO0		CAN5RX				165	A4	—	A9
P11_6	RLIN33RX/INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO	TAUB1I7	TAUB1O7	MEMC0AD14			SFMA0SSL	RLIN33RX				166	D6	—	B9
P11_7	INTP5			PWGA32O	CSIH3SC		TAUB1I9	TAUB1O9	MEMC0AD15			SFMA0CLK					167	C6	—	C9
P11_8	$\overline{\text{CSIG1SSI}}$	RLIN35TX		PWGA48O	TAUB1I11	TAUB1O11		$\overline{\text{MEMC0CS0}}$									8	E3	—	E2
P11_9		CSIG1SO	RLIN35RX/INTP15	PWGA49O	TAUB1I13	TAUB1O13		$\overline{\text{MEMC0CS1}}$					RLIN35RX				9	D1	—	F3
P11_10	CSIG1SC			PWGA50O	TAUB1I15	TAUB1O15		$\overline{\text{MEMC0CS2}}$									10	E2	—	G3
P11_11	CSIG1SI	RLIN25TX		PWGA51O	TAUB1I0	TAUB1O0		$\overline{\text{MEMC0CS3}}$	ETNB0RXDV								11	F3	—	G2
P11_12	RLIN25RX			PWGA52O	TAUB1I2	TAUB1O2	$\overline{\text{MEMC0WAIT}}$										12	E1	—	H2
P11_15	CAN2RX/INTP2	CSIH2CSS4		PWGA55O	TAUB1I8	TAUB1O8		$\overline{\text{MEMC0ASTB}}$	ETNB0RXERR	RLIN36TX			CAN2RX				168	D5	—	D9

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.9.2 Control Registers

Table 2A.56 Control Registers (P11)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W^{*1}			176 Pins	233 Pins	272 Pins	324 Pins
P11	Port register 11	16	15, 12-0	R/W	002C$_H$	0000$_H$	✓	✓	—	✓
PSR11	Port set/reset register 11	32	31, 28-16, 15, 12-0	R/W	012C$_H$	0000 0000$_H$	✓	✓	—	✓
PPR11	Port pin read register 11	16	15, 12-0	R	022C$_H$	0000$_H$	✓	✓	—	✓
PM11	Port mode register 11	16	15, 12-0	R/W	032C$_H$	FFFF$_H$	✓	✓	—	✓
PMC11	Port mode control register 11	16	15, 12-0	R/W	042C$_H$	0000$_H$	✓	✓	—	✓
PFC11	Port function control register 11	16	15, 12-0	R/W	052C$_H$	0000$_H$	✓	✓	—	✓
PFCE11	Port function control expansion register 11	16	15, 12-0	R/W	062C$_H$	0000$_H$	✓	✓	—	✓
PNOT11	Port NOT register 11	16	15, 12-0	W	072C$_H$	0000$_H$	✓	✓	—	✓
PMSR11	Port mode set/reset register 11	32	31, 28-16, 15, 12-0	R/W	082C$_H$	0000 FFFF$_H$	✓	✓	—	✓
PMCSR11	Port mode control set/reset register 11	32	31, 28-16, 15, 12-0	R/W	092C$_H$	0000 0000$_H$	✓	✓	—	✓
PFCAE11	Port function control additional expansion register 11	16	15, 11, 9, 7-0	R/W	0A2C$_H$	0000$_H$	✓	✓	—	✓
PIBC11	Port input buffer control register 11	16	15, 12-0	R/W	402C$_H$	0000$_H$	✓	✓	—	✓
PBDC11	Port bidirection control register 11	16	15, 12-0	R/W	412C$_H$	0000$_H$	✓	✓	—	✓
PIPC11	Port IP control register 11	16	10, 9, 7-1	R/W	422C$_H$	0000$_H$	✓	✓	—	✓
PU11	Pull-up option register 11	16	15, 12-0	R/W	432C$_H$	0000$_H$	✓	✓	—	✓
PD11	Pull-down option register 11	16	15, 12-0	R/W	442C$_H$	0000$_H$	✓	✓	—	✓
PODC11	Port open drain control register 11	32	15, 12-0	R/W	452C$_H$	0000 0000$_H$	✓	✓	—	✓
PDSC11	Port drive strength control register 11	32	15, 12-0	R/W	462C$_H$	0000 0000$_H$	✓	✓	—	✓
PIS11	Port input buffer selection register 11	16	15, 12-0	R/W	472C$_H$	FFFF$_H$	✓	✓	—	✓
PISA11	Port input buffer selection advanced register 11	16	15, 12-10	R/W	4A2C$_H$	0000$_H$	✓	✓	—	✓
PPROTS11	Port protection status register 11	32	0	R	4B2C$_H$	0000 0000$_H$	✓	✓	—	✓
PPCMD11	Port protection command register 11	32	7-0	W	4C2C$_H$	xxxx xx00$_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.10 Port 12 (P12)

2A.10.10.1 Alternative Function

Table 2A.57 Port 12 (P12)

Port Mode (PMC12_m =0)	Alternative Mode (PMC12_m =1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P12_0		CAN2TX		PWGA56O	TAUB1I10	TAUB1O10	$\overline{\text{CSIG2SSI}}$	MEMC0A16	RLIN36RX/ INTP16				RLIN36RX				169	B4	—	A7
P12_1	RLIN34RX/ INTP14	CSIH2CSS5		PWGA57O	TAUB1I12	TAUB1O12		MEMC0A17					RLIN34RX				170	C5	—	A6
P12_2	INTP19	RLIN34TX		PWGA58O	TAUB1I14	TAUB1O14		MEMC0A18	CSIG2RYI	CSIG2RYO							171	A3	—	D8
P12_3	RLIN27RX			PWGA68O	CSIG2SI	$\overline{\text{MEMC0BEN0}}$	TAUB1I6	TAUB1O6									15	G1	—	L4
P12_4		RLIN27TX		PWGA69O	CSIG2SC		ETNB0MDIO			$\overline{\text{MEMC0BEN1}}$							16	H1	—	N4
P12_5		PWGA70O		ETNB0MDC		CSIG2SO	TAUB1I4	TAUB1O4									17	J3	—	R1

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.10.2 Control Registers

Table 2A.58 Control Registers (P12)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P12	Port register 12	16	5-0	R/W	0030_H	0000_H	✓	✓	—	✓
PSR12	Port set/reset register 12	32	21-16, 5-0	R/W	0130_H	$0000\ 0000_H$	✓	✓	—	✓
PPR12	Port pin read register 12	16	5-0	R	0230_H	0000_H	✓	✓	—	✓
PM12	Port mode register 12	16	5-0	R/W	0330_H	$FFFF_H$	✓	✓	—	✓
PMC12	Port mode control register 12	16	5-0	R/W	0430_H	0000_H	✓	✓	—	✓
PFC12	Port function control register 12	16	5-0	R/W	0530_H	0000_H	✓	✓	—	✓
PFCE12	Port function control expansion register 12	16	5-0	R/W	0630_H	0000_H	✓	✓	—	✓
PNOT12	Port NOT register 12	16	5-0	W	0730_H	0000_H	✓	✓	—	✓
PMSR12	Port mode set/reset register 12	32	21-16, 5-0	R/W	0830_H	$0000\ FFFF_H$	✓	✓	—	✓
PMCSR12	Port mode control set/reset register 12	32	21-16, 5-0	R/W	0930_H	$0000\ 0000_H$	✓	✓	—	✓
PFCAE12	Port function control expansion register 12	16	4, 2-0	R/W	$0A30_H$	0000_H	✓	✓	—	✓
PIBC12	Port input buffer control register 12	16	5-0	R/W	4030_H	0000_H	✓	✓	—	✓
PBDC12	Port bidirection control register 12	16	5-0	R/W	4130_H	0000_H	✓	✓	—	✓
PIPC12	Port IP control register 12	16	5, 4	R/W	4230_H	0000_H	✓	✓	—	✓
PU12	Pull-up option register 12	16	5-0	R/W	4330_H	0000_H	✓	✓	—	✓
PD12	Pull-down option register 12	16	5-0	R/W	4430_H	0000_H	✓	✓	—	✓
PODC12	Port open drain control register 12	32	5-0	R/W	4530_H	$0000\ 0000_H$	✓	✓	—	✓
PDSC12	Port drive strength control register 12	32	5-0	R/W	4630_H	$0000\ 0000_H$	✓	✓	—	✓
PIS12	Port input buffer selection register 12	16	5-0	R/W	4730_H	$FFFF_H$	✓	✓	—	✓
PISA12	Port input buffer selection advanced register 12	16	4	R/W	$4A30_H$	0000_H	✓	✓	—	✓
PPROTS12	Port protection status register 12	32	0	R	$4B30_H$	$0000\ 0000_H$	✓	✓	—	✓
PPCMD12	Port protection command register 12	32	7-0	W	$4C30_H$	$xxxx\ xx00_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.11 Port 13 (P13)

2A.10.11.1 Alternative Function

Table 2A.59 Port 13 (P13)

Port Mode (PMC13_m =0)	Alternative Mode (PMC13_m =1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P13_0		MEMC0A19															—	C4	—	C7
P13_1		MEMC0A20															—	B3	—	B6
P13_2	ETNB0RXDV																—	F2	—	K3
P13_3	ETNB0RXERR																—	F1	—	L3
P13_4																	—	G2	—	—
	ETNB1LINK																—	—	—	M3
P13_5		MEMC0A21															—	G3	—	K2
P13_6		MEMC0A22		PWGA72O													—	H3	—	M4
P13_7				PWGA73O													—	H2	—	—
		MEMC0A23		PWGA73O													—	—	—	N3

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.11.2 Control Registers

Table 2A.60 Control Registers (P13)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W*1	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P13	Port register 13	16	7-0	R/W	0034$_H$	0000$_H$	—	✓	—	✓
PSR13	Port set/reset register 13	32	23-16, 7-0	R/W	0134$_H$	0000 0000$_H$	—	✓	—	✓
PPR13	Port pin read register 13	16	7-0	R	0234$_H$	0000$_H$	—	✓	—	✓
PM13	Port mode register 13	16	7-0	R/W	0334$_H$	FFFF$_H$	—	✓	—	✓
PMC13	Port mode control register 13	16	7-5, 3-0	R/W	0434$_H$	0000$_H$	—	✓	—	—
			7-0				—	—	—	✓
PFC13	Port function control register 13	16	7, 6	R/W	0534$_H$	0000$_H$	—	✓	—	✓
PNOT13	Port NOT register 13	16	7-0	W	0734$_H$	0000$_H$	—	✓	—	✓
PMSR13	Port mode set/reset register 13	32	23-16, 7-0	R/W	0834$_H$	0000 FFFF$_H$	—	✓	—	✓
PMCSR13	Port mode control set/reset register 13	32	23-21, 19-16, 7-5, 3-0	R/W	0934$_H$	0000 0000$_H$	—	✓	—	—
			23-16, 7-0				—	—	—	✓
PIBC13	Port input buffer control register 13	16	7-0	R/W	4034$_H$	0000$_H$	—	✓	—	✓
PBDC13	Port bidirection control register 13	16	7-0	R/W	4134$_H$	0000$_H$	—	✓	—	✓
PU13	Pull-up option register 13	16	7-0	R/W	4334$_H$	0000$_H$	—	✓	—	✓
PD13	Pull-down option register 13	16	7-0	R/W	4434$_H$	0000$_H$	—	✓	—	✓
PODC13	Port open drain control register 13	32	7-0	R/W	4534$_H$	0000 0000$_H$	—	✓	—	✓
PDSC13	Port drive strength control register 13	32	7-0	R/W	4634$_H$	0000 0000$_H$	—	✓	—	✓
PIS13	Port input buffer selection register 13	16	7-0	R/W	4734$_H$	FFFF$_H$	—	✓	—	✓
PISA13	Port input buffer selection advanced register 13	16	5, 3, 2	R/W	4A34$_H$	0000$_H$	—	✓	—	—
			5-2				—	—	—	✓
PPROTS13	Port protection status register 13	32	0	R	4B34$_H$	0000 0000$_H$	—	✓	—	✓
PPCMD13	Port protection command register 13	32	7-0	W	4C34$_H$	0000 0000$_H$	—	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.12 Port 18 (P18)

2A.10.12.1 Alternative Function

Table 2A.61 Port 18 (P18)

Port Mode (PMC18_m = 0)	Alternative Mode (PMC18_m = 1)										ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P18_0	CSIG1RYI	CSIG1RYO	ETNB0LINK	PWGA61O			TAUJ3I0	TAUJ3O0			ADCA1I0S		143	C14	—	A21
P18_1		PWGA62O		ETNB0TXD0			TAUJ3I1	TAUJ3O1			ADCA1I1S		144	B15	—	B20
P18_2		PWGA63O		ETNB0TXD1			TAUJ3I2	TAUJ3O2			ADCA1I2S		145	B14	—	C19
P18_3		PWGA71O		ETNB0TXD2			TAUJ3I3	TAUJ3O3			ADCA1I3S		146	B13	—	A19
P18_4		CSIH1CSS4		ETNB0TXD3							ADCA1I4S		147	C11	—	B18
P18_5		CSIH1CSS5		ETNB0TXEN							ADCA1I5S		148	A14	—	D16
P18_6											ADCA1I6S		149	A13	—	—
		PWGA95O									ADCA1I6S		—	—	—	A18
P18_7	ETNB0TXCLK										ADCA1I7S		150	B11	—	B16
P18_8											ADCA1I8S		—	A16	—	B19
P18_9											ADCA1I9S		—	C13	—	A20
P18_10											ADCA1I10S		—	A15	—	C18
P18_11											ADCA1I11S		—	B12	—	C17
P18_12											ADCA1I12S		—	C12	—	C16
P18_13											ADCA1I13S		—	A12	—	B17
P18_14											ADCA1I14S		—	C9	—	C15
P18_15											ADCA1I15S		—	A11	—	A17

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2A.62, Control Registers (P18)**.

2A.10.12.2 Control Registers

Table 2A.62 Control Registers (P18)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W^{*1}	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P18	Port register 18	16	7-0	R/W	0048_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PSR18	Port set/reset register 18	32	23-16, 7-0	R/W	0148_H	$0000\ 0000_H$	✓	—	—	—
			31-16, 15-0				—	✓	—	✓
PPR18	Port pin read register 18	16	7-0	R	0248_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PM18	Port mode register 18	16	7-0	R/W	0348_H	$FFFF_H$	✓	—	—	—
			15-0				—	✓	—	✓
PMC18	Port mode control register 18	16	7, 5-0	R/W	0448_H	0000_H	✓	✓	—	—
			7-0				—	—	—	✓
PFC18	Port function control register 18	16	5-0	R/W	0548_H	0000_H	✓	✓	—	✓
PFCE18	Port function control expansion register 18	16	3-0	R/W	0648_H	0000_H	✓	✓	—	✓
PNOT18	Port NOT register 18	16	7-0	W	0748_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PMSR18	Port mode set/reset register 18	32	23-16, 7-0	R/W	0848_H	$0000\ FFFF_H$	✓	—	—	—
			31-16, 15-0				—	✓	—	✓
PMCSR18	Port mode control set/reset register 18	32	23, 21-16, 7, 5-0	R/W	0948_H	$0000\ 0000_H$	✓	✓	—	—
			23-16, 7-0				—	—	—	✓
PIBC18	Port input buffer control register 18	16	7-0	R/W	4048_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PBDC18	Port bidirection control register 18	16	7-0	R/W	4148_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PU18	Pull-up option register 18	16	7-0	R/W	4348_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PD18	Pull-down option register 18	16	7-0	R/W	4448_H	0000_H	✓	—	—	—
			15-0				—	✓	—	✓
PODC18	Port open drain control register 18	32	7-0	R/W	4548_H	$0000\ 0000_H$	✓	—	—	—
			15-0				—	✓	—	✓
PDSC18	Port drive strength control register 18	32	7-0	R/W	4648_H	$0000\ 0000_H$	✓	—	—	—
			15-0				—	✓	—	✓
PIS18	Port input buffer selection register 18	16	7-0	R/W	4748_H	$FFFF_H$	✓	—	—	—
			15-0				—	✓	—	✓
PISA18	Port protection status register 18	16	7, 0	R/W	$4A48_H$	0000_H	✓	—	—	—
			9-7, 0				—	✓	—	✓
PPROTS18	Port protection status register 18	32	0	R	$4B48_H$	$0000\ 0000_H$	✓	✓	—	✓
PPCMD18	Port protection command register 18	32	7-0	W	$4C48_H$	$xxxx\ xx00_H$	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.13 Port 19 (P19)

2A.10.13.1 Alternative Function

Table 2A.63 Port 19 (P19)

	Alternative Mode																PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
Port Mode	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	ADC	Special Function				
P19_0															ADCA1I16S		—	C10	—	A16
P19_1															ADCA1I17S		—	B10	—	A15
P19_2															ADCA1I18S		—	A10	—	D15
P19_3															ADCA1I19S		—	B9	—	B15

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2A.64, Control Registers (P19)**.

2A.10.13.2 Control Registers

Table 2A.64 Control Registers (P19)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P19	Port register 19	16	3-0	R/W	$004C_H$	0000_H	—	✓	—	✓
PSR19	Port set/reset register 19	32	19-16, 3-0	R/W	$014C_H$	$0000\ 0000_H$	—	✓	—	✓
PPR19	Port pin read register 19	16	3-0	R	$024C_H$	0000_H	—	✓	—	✓
PM19	Port mode register 19	16	3-0	R/W	$034C_H$	$FFFF_H$	—	✓	—	✓
PNOT19	Port NOT register 19	16	3-0	W	$074C_H$	0000_H	—	✓	—	✓
PMSR19	Port mode set/reset register 19	32	19-16, 3-0	R/W	$084C_H$	$0000\ FFFF_H$	—	✓	—	✓
PIBC19	Port input buffer control register 19	16	3-0	R/W	$404C_H$	0000_H	—	✓	—	✓
PBDC19	Port bidirection control register 19	16	3-0	R/W	$414C_H$	0000_H	—	✓	—	✓
PU19	Pull-up option register 19	16	3-0	R/W	$434C_H$	0000_H	—	✓	—	✓
PD19	Pull-down option register 19	16	3-0	R/W	$444C_H$	0000_H	—	✓	—	✓
PODC19	Port open drain control register 19	32	3-0	R/W	$454C_H$	$0000\ 0000_H$	—	✓	—	✓
PDSC19	Port drive strength control register 19	32	3-0	R/W	$464C_H$	$0000\ 0000_H$	—	✓	—	✓
PIS19	Port input buffer selection register 19	16	3-0	R/W	$474C_H$	$FFFF_H$	—	✓	—	✓
PPROTS19	Port protection status register 19	32	0	R	$4B4C_H$	$0000\ 0000_H$	—	✓	—	✓
PPCMD19	Port protection command register 19	32	7-0	W	$4C4C_H$	$0000\ 0000_H$	—	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.14 Port 20 (P20)

2A.10.14.1 Alternative Function

Table 2A.65 Port 20 (P20)

Port Mode (PMC20_m =0)	Alternative Mode (PMC20_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P20_0	RLIN26RX			PWGA64O	CAN6RX/ INTP6		CSIG3SI						CAN6RX				118	G17	—	J20
P20_1		RLIN26TX		PWGA65O		CAN6TX		CSIG3SO									117	H15	—	J22
P20_2	CAN4RX/ INTP4			PWGA66O	RLIN29RX		CSIG3SC						CAN4RX				116	H16	—	K20
P20_3		CAN4TX		PWGA67O		RLIN29TX	CSIG3RYI	CSIG3RYO									115	H17	—	K21
P20_4	RLIN23RX		INTP22	PWGA59O	CAN7RX/ INTP9		$\overline{\text{CSIG3SSI}}$						CAN7RX				120	F17	—	H22
P20_5		RLIN23TX	INTP23	PWGA60O		CAN7TX											119	G16	—	J21
P20_6		PWGA88O															—	—	—	G21
P20_7		PWGA89O															—	—	—	G22
P20_8		PWGA90O															—	—	—	H20
P20_9		PWGA91O															—	—	—	H21
P20_10		PWGA92O															—	—	—	K22
P20_11		PWGA93O															—	—	—	L20
P20_12		PWGA94O															—	—	—	L21
P20_13		PWGA95O															—	—	—	—
	RLIN215RX	PWGA95O															—	—	—	M20
P20_14																	—	—	—	—
		RLIN215TX															—	—	—	M21
P20_15	RLIN214RX																—	—	—	L22

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.14.2 Control Registers

Table 2A.66 Control Registers (P20)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W^{*1}	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P20	Port register 20	16	5-0	R/W	0050$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PSR20	Port set/reset register 20	32	21-16, 5-0	R/W	0150$_H$	0000 0000$_H$	✓	✓	—	—
			30-16, 14-0				—	—	—	—
			31-16, 15-0				—	—	—	✓
PPR20	Port pin read register 20	16	5-0	R	0250$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PM20	Port mode register 20	16	5-0	R/W	0350$_H$	FFFF$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PMC20	Port mode control register 20	16	5-0	R/W	0450$_H$	0000$_H$	✓	✓	—	—
			13-0				—	—	—	—
			15-0				—	—	—	✓
PFC20	Port function control register 20	16	5-0	R/W	0550$_H$	0000$_H$	✓	✓	—	✓
PFCE20	Port function control expansion register 20	16	5-0	R/W	0650$_H$	0000$_H$	✓	✓	—	✓
PNOT20	Port NOT register 20	16	5-0	W	0750$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PMSR20	Port mode set/reset register 20	32	21-16, 5-0	R/W	0850$_H$	0000 FFFF$_H$	✓	✓	—	—
			30-16, 14-0				—	—	—	—
			31-16, 15-0				—	—	—	✓
PMCSR20	Port mode control set/reset register 20	32	21-16, 5-0	R/W	0950$_H$	0000 0000$_H$	✓	✓	—	—
			29-16, 13-0				—	—	—	—
			31-16, 15-0				—	—	—	✓
PFCAE20	Port function control additional expansion register 20	16	4, 2, 0	R/W	0A50$_H$	0000$_H$	✓	✓	—	✓
PIBC20	Port input buffer control register 20	16	5-0	R/W	4050$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PBDC20	Port bidirection control register 20	16	5-0	R/W	4150$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PIPC20	Port IP control register 20	16	2, 1	R/W	4250$_H$	0000$_H$	✓	✓	—	✓
PU20	Pull-up option register 20	16	5-0	R/W	4350$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PD20	Pull-down option register 20	16	5-0	R/W	4450$_H$	0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PODC20	Port open drain control register 20	32	5-0	R/W	4550$_H$	0000 0000$_H$	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓

Table 2A.66 Control Registers (P20)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
PDSC20	Port drive strength control register 20	32	5-0	R/W	4650H	0000 0000H	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PIS20	Port input buffer selection register 20	16	5-0	R/W	4750H	FFFFH	✓	✓	—	—
			14-0				—	—	—	—
			15-0				—	—	—	✓
PPROTS20	Port protection status register 20	32	0	R	4B50H	0000 0000H	✓	✓	—	✓
PPCMD20	Port protection command register 20	32	7-0	W	4C50H	xxxx xx00H	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.15 Port 21 (P21)

2A.10.15.1 Alternative Function

Table 2A.67 Port 21 (P21)

Port Mode (PMC21_m =0)	Alternative Mode (PMC21_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P21_0																	—	—	—	—
	ETNB1RXDV																—	—	—	H3
P21_1																	—	—	—	—
	MMCA0DAT7																—	—	—	B11
P21_2																	—	—	—	—
	ETNB1RXD2																—	—	—	J2
P21_3																	—	—	—	—
	ETNB1RXD1																—	—	—	J3
P21_4																	—	—	—	—
	ETNB1RXD0																—	—	—	H1
P21_5	ETNB1RXD3																—	—	—	J1
P21_6		ETNB1MDC															—	—	—	K1
P21_7	ETNB1MDIO																—	—	—	L1
P21_8	ETNB1RXERR																—	—	—	L2
P21_9																	—	—	—	M1
P21_10																	—	—	—	M2
P21_11	RLIN213RX																—	—	—	N1
P21_12		RLIN213TX															—	—	—	P1
P21_13	RLIN212RX																—	—	—	N2
P21_14		RLIN212TX															—	—	—	P2

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.15.2 Control Registers

Table 2A.68 Control Registers (P21)

			Effective Bit				Device			
Register	Function	Register Size	Position	R/W*1	Offset Address	Value after Reset	176 Pins	233 Pins	272 Pins	324 Pins
P21	Port register 21	16	4-0	R/W	0054$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PSR21	Port set/reset register 21	32	20-16, 4-0	R/W	0154$_H$	0000 0000$_H$	—	—	—	—
			30-16, 14-0				—	—	—	✓
PPR21	Port pin read register 21	16	4-0	R	0254$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PM21	Port mode register 21	16	4-0	R/W	0354$_H$	FFFF$_H$	—	—	—	—
			14-0				—	—	—	✓
PMC21	Port mode control register 21	16	14-11, 8-0	R/W	0454$_H$	0000$_H$	—	—	—	✓
PNOT21	Port NOT register 21	16	4-0	W	0754$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PMSR21	Port mode set/reset register 21	32	20-16, 4-0	R/W	0854$_H$	0000 FFFF$_H$	—	—	—	—
			30-16, 14-0				—	—	—	✓
PMCSR21	Port mode control set/reset register 21	32	30-27, 24-16, 14-11, 8-0	R/W	0954$_H$	0000 0000$_H$	—	—	—	✓
PIBC21	Port input buffer control register 21	16	4-0	R/W	4054$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PBDC21	Port bidirection control register 21	16	4-0	R/W	4154$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PIPC21	Port IP control register 21	16	7, 1	R/W	4254$_H$	0000$_H$	—	—	—	✓
PU21	Pull-up option register 21	16	4-0	R/W	4354$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PD21	Pull-down option register 21	16	4-0	R/W	4454$_H$	0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PODC21	Port open drain control register 21	32	4-0	R/W	4554$_H$	0000 0000$_H$	—	—	—	—
			14-0				—	—	—	✓
PDSC21	Port drive strength control register 21	32	14-0	R/W	4654$_H$	0000 0000$_H$	—	—	—	✓
PIS21	Port input buffer selection register 21	16	4-0	R/W	4754$_H$	FFFF$_H$	—	—	—	—
			14-0				—	—	—	✓
PISA21	Port input buffer selection advanced register 21	16	10-7, 5-2, 0	R/W	4A54$_H$	0000$_H$	—	—	—	✓
PPROTS21	Port protection status register 21	32	0	R	4B54$_H$	0000 0000$_H$	—	—	—	✓
PPCMD21	Port protection command register 21	32	7-0	W	4C54$_H$	0000 0000$_H$	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.16 Port 22 (P22)

2A.10.16.1 Alternative Function

Table 2A.69 Port 22 (P22)

Port Mode (PMC22_m =0)	Alternative Mode (PMC22_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P22_0																	—	—	—	—
	ETNB1RXCLK																—	—	—	G1
P22_1																	—	—	—	—
		ETNB1TXD3															—	—	—	F2
P22_2																	—	—	—	—
		ETNB1TXD2															—	—	—	F1
P22_3																	—	—	—	—
		ETNB1TXD1															—	—	—	E1
P22_4																	—	—	—	—
		ETNB1TXD0															—	—	—	D2
P22_5																	—	—	—	—
		ETNB1TXEN															—	—	—	D3
P22_6																	—	—	—	—
	ETNB1TXCLK																—	—	—	C2
P22_7																	—	—	—	—
	MMCA0CMD																—	—	—	A5
P22_8																	—	—	—	—
		MMCA0CLK															—	—	—	C6
P22_9																	—	—	—	—
	MMCA0DAT0																—	—	—	B7
P22_10																	—	—	—	—
	MMCA0DAT1																—	—	—	C8
P22_11																	—	—	—	—
	MMCA0DAT2																—	—	—	A8
P22_12																	—	—	—	—
	MMCA0DAT3																—	—	—	B8
P22_13																	—	—	—	—
	MMCA0DAT4																—	—	—	C10

Table 2A.69 Port 22 (P22)

Port Mode (PMC22_m =0)	Alternative Mode (PMC22_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P22_14																	—	—	—	—
	MMCA0DAT5																—	—	—	B10
P22_15																	—	—	—	—
	MMCA0DAT6																—	—	—	A11

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.16.2 Control Registers

Table 2A.70 Control Registers (P22)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P22	Port register 22	16	15-0	R/W	0058_H	0000_H	—	—	—	✓
PSR22	Port set/reset register 22	32	31-16, 15-0	R/W	0158_H	$0000\ 0000_H$	—	—	—	✓
PPR22	Port pin read register 22	16	15-0	R	0258_H	0000_H	—	—	—	✓
PM22	Port mode register 22	16	15-0	R/W	0358_H	$FFFF_H$	—	—	—	✓
PMC22	Port mode control register 22	16	15-0	R/W	0458_H	0000_H	—	—	—	✓
PNOT22	Port NOT register 22	16	15-0	W	0758_H	0000_H	—	—	—	✓
PMSR22	Port mode set/reset register 22	32	31-16, 15-0	R/W	0858_H	$0000\ FFFF_H$	—	—	—	✓
PMCSR22	Port mode control set/reset register 22	32	31-16, 15-0	R/W	0958_H	$0000\ 0000_H$	—	—	—	✓
PIBC22	Port input buffer control register 22	16	15-0	R/W	4058_H	0000_H	—	—	—	✓
PBDC22	Port bidirection control register 22	16	15-0	R/W	4158_H	0000_H	—	—	—	✓
PIPC22	Port IP control register 22	16	15-9, 7	R/W	4258_H	0000_H	—	—	—	✓
PU22	Pull-up option register 22	16	15-0	R/W	4358_H	0000_H	—	—	—	✓
PD22	Pull-down option register 22	16	15-0	R/W	4458_H	0000_H	—	—	—	✓
PODC22	Port open drain control register 22	32	15-0	R/W	4558_H	$0000\ 0000_H$	—	—	—	✓
PDSC22	Port drive strength control register 22	32	15-0	R/W	4658_H	$0000\ 0000_H$	—	—	—	✓
PIS22	Port input buffer selection register 22	16	15-0	R/W	4758_H	$FFFF_H$	—	—	—	✓
PISA22	Port input buffer selection advanced register 22	16	6, 0	R/W	$4A58_H$	0000_H	—	—	—	✓
PPROTS22	Port protection status register 22	32	0	R	$4B58_H$	$0000\ 0000_H$	—	—	—	✓
PPCMD22	Port protection command register 22	32	7-0	W	$4C58_H$	$0000\ 0000_H$	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.17 Port 23 (P23)

2A.10.17.1 Alternative Function

Table 2A.71 Port 23 (P23)

Port Mode (PMC23_m =0)	Alternative Mode (PMC23_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P23_0	$\overline{\text{CSIH4SSI}}$																—	—	—	T21
P23_1		CSIH4SO															—	—	—	T22
P23_2	CSIH4SI																—	—	—	R21
P23_3	CSIH4SC																—	—	—	R22
P23_4	CSIH4RYI	CSIH4RYO															—	—	—	P21
P23_5	CSIG4RYI	CSIG4RYO															—	—	—	P22
P23_6		CSIG4SO															—	—	—	P20
P23_7	CSIG4SI																—	—	—	N20
P23_8	CSIG4SC																—	—	—	N21
P23_9	$\overline{\text{CSIG4SSI}}$																—	—	—	N22
P23_10		RLIN214TX															—	—	—	M22

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.17.2 Control Registers

Table 2A.72 Control Registers (P23)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P23	Port register 23	16	10-0	R/W	$005C_H$	0000_H	—	—	—	✓
PSR23	Port set/reset register 23	32	26-16, 10-0	R/W	$015C_H$	$0000\ 0000_H$	—	—	—	✓
PPR23	Port pin read register 23	16	10-0	R	$025C_H$	0000_H	—	—	—	✓
PM23	Port mode register 23	16	10-0	R/W	$035C_H$	$FFFF_H$	—	—	—	✓
PMC23	Port mode control register 23	16	10-0	R/W	$045C_H$	0000_H	—	—	—	✓
PNOT23	Port NOT register 23	16	10-0	W	$075C_H$	0000_H	—	—	—	✓
PMSR23	Port mode set/reset register 23	32	26-16, 10-0	R/W	$085C_H$	$0000\ FFFF_H$	—	—	—	✓
PMCSR23	Port mode control set/reset register 23	32	26-16, 10-0	R/W	$095C_H$	$0000\ 0000_H$	—	—	—	✓
PIBC23	Port input buffer control register 23	16	10-0	R/W	$405C_H$	0000_H	—	—	—	✓
PBDC23	Port bidirection control register 23	16	10-0	R/W	$415C_H$	0000_H	—	—	—	✓
PIPC23	Port IP control register 23	16	8, 6, 3, 1	R/W	$425C_H$	0000_H	—	—	—	✓
PU23	Pull-up option register 23	16	10-0	R/W	$435C_H$	0000_H	—	—	—	✓
PD23	Pull-down option register 23	16	10-0	R/W	$445C_H$	0000_H	—	—	—	✓
PODC23	Port open drain control register 23	32	10-0	R/W	$455C_H$	$0000\ 0000_H$	—	—	—	✓
PDSC23	Port drive strength control register 23	32	10-0	R/W	$465C_H$	$0000\ 0000_H$	—	—	—	✓
PIS23	Port input buffer selection register 23	16	10-0	R/W	$475C_H$	$FFFF_H$	—	—	—	✓
PPROTS23	Port protection status register 23	32	0	R	$4B5C_H$	$0000\ 0000_H$	—	—	—	✓
PPCMD23	Port protection command register 23	32	7-0	W	$4C5C_H$	$0000\ 0000_H$	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.18 Port 24 (P24)

2A.10.18.1 Alternative Function

Table 2A.73 Port 24 (P24)

Port Mode (PMC24_m =0)	Alternative Mode (PMC24_m = 1)														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P24_0		CAN8TX															—	—	—	B5
P24_1	CAN8RX/ INTP18		CAN8RX														—	—	—	B4
P24_2		CAN9TX															—	—	—	A3
P24_3	CAN9RX/ INTP19		CAN9RX														—	—	—	A4
P24_4		CAN10TX															—	—	—	B3
P24_5	CAN10RX/ INTP20		CAN10RX														—	—	—	A2
P24_6		CAN11TX															—	—	—	B1
P24_7	CAN11RX/ INTP21		CAN11RX														—	—	—	C1

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2A.10.18.2 Control Registers

Table 2A.74 Control Registers (P24)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
P24	Port register 24	16	7-0	R/W	0060_H	0000_H	—	—	—	✓
PSR24	Port set/reset register 24	32	23-16, 7-0	R/W	0160_H	$0000\ 0000_H$	—	—	—	✓
PPR24	Port pin read register 24	16	7-0	R	0260_H	0000_H	—	—	—	✓
PM24	Port mode register 24	16	7-0	R/W	0360_H	$FFFF_H$	—	—	—	✓
PMC24	Port mode control register 24	16	7-0	R/W	0460_H	0000_H	—	—	—	✓
PFC24	Port function control register 24	16	7, 5, 3, 1	R/W	0560_H	0000_H	—	—	—	✓
PNOT24	Port NOT register 24	16	7-0	W	0760_H	0000_H	—	—	—	✓
PMSR24	Port mode set/reset register 24	32	23-16, 7-0	R/W	0860_H	$0000\ FFFF_H$	—	—	—	✓
PMCSR24	Port mode control set/reset register 24	32	23-16, 7-0	R/W	0960_H	$0000\ 0000_H$	—	—	—	✓
PIBC24	Port input buffer control register 24	16	7-0	R/W	4060_H	0000_H	—	—	—	✓
PBDC24	Port bidirection control register 24	16	7-0	R/W	4160_H	0000_H	—	—	—	✓
PU24	Pull-up option register 24	16	7-0	R/W	4360_H	0000_H	—	—	—	✓
PD24	Pull-down option register 24	16	7-0	R/W	4460_H	0000_H	—	—	—	✓
PODC24	Port open drain control register 24	32	7-0	R/W	4560_H	$0000\ 0000_H$	—	—	—	✓
PDSC24	Port drive strength control register 24	32	7-0	R/W	4660_H	$0000\ 0000_H$	—	—	—	✓
PIS24	Port input buffer selection register 24	16	7-0	R/W	4760_H	$FFFF_H$	—	—	—	✓
PPROTS24	Port protection status register 24	32	0	R	$4B60_H$	$0000\ 0000_H$	—	—	—	✓
PPCMD24	Port protection command register 24	32	7-0	W	$4C60_H$	$0000\ 0000_H$	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.19 Analog Port 0 (AP0)

2A.10.19.1 Alternative Function

Table 2A.75 Analog Port 0 (AP0)

Port Mode	Alternative Mode														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
AP0_0															ADCA0I0		106	K15	—	U22
AP0_1															ADCA0I1		105	L17	—	V22
AP0_2															ADCA0I2		104	L16	—	U21
AP0_3															ADCA0I3		103	M17	—	W22
AP0_4															ADCA0I4		102	L15	—	V21
AP0_5															ADCA0I5		101	M16	—	U20
AP0_6															ADCA0I6		100	N17	—	W21
AP0_7															ADCA0I7		99	N16	—	Y22
AP0_8															ADCA0I8		98	M15	—	W20
AP0_9															ADCA0I9		97	P17	—	AA22
AP0_10															ADCA0I10		96	P16	—	V20
AP0_11															ADCA0I11		95	N15	—	AA21
AP0_12															ADCA0I12		94	R17	—	Y21
AP0_13															ADCA0I13		93	P15	—	Y20
AP0_14															ADCA0I14		92	R16	—	W19
AP0_15															ADCA0I15		91	T17	—	AB21

CAUTION

Use ADC functions with their initial settings. For details, see **Table 2A.76, Control Registers (AP0)**.

2A.10.19.2 Control Registers

Table 2A.76 Control Registers (AP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
AP0	Analog port register 0	16	15-0	R/W	$00C8_H$	0000_H	✓	✓	—	✓
APSR0	Analog port set/reset register 0	32	31-16, 15-0	R/W	$01C8_H$	$0000\ 0000_H$	✓	✓	—	✓
APPR0	Analog port pin read register 0	16	15-0	R	$02C8_H$	0000_H	✓	✓	—	✓
APM0	Analog port mode register 0	16	15-0	R/W	$03C8_H$	$FFFF_H$	✓	✓	—	✓
APNOT0	Analog port NOT register 0	16	15-0	W	$07C8_H$	0000_H	✓	✓	—	✓
APMSR0	Analog port mode set/reset register 0	32	31-16, 15-0	R/W	$08C8_H$	$0000\ FFFF_H$	✓	✓	—	✓
APIBC0	Analog port input buffer control register 0	16	15-0	R/W	$40C8_H$	0000_H	✓	✓	—	✓
APBDC0	Analog port bidirection control register 0	16	15-0	R/W	$41C8_H$	0000_H	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.20 Analog Port 1 (AP1)

2A.10.20.1 Alternative Function

Table 2A.77 Analog Port 1 (AP1)

Port Mode	Alternative Mode														ADC	Special Function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
AP1_0															ADCA1I0		133	C17	—	C21
AP1_1															ADCA1I1		132	D15	—	C22
AP1_2															ADCA1I2		131	D16	—	E20
AP1_3															ADCA1I3		130	D17	—	D21
AP1_4															ADCA1I4		129	F14	—	E21
AP1_5															ADCA1I5		128	E15	—	D22
AP1_6															ADCA1I6		127	E16	—	F20
AP1_7															ADCA1I7		126	F15	—	E22
AP1_8															ADCA1I8		125	E17	—	F22
AP1_9															ADCA1I9		124	F16	—	F21
AP1_10															ADCA1I10		123	G14	—	G19
AP1_11															ADCA1I11		122	G15	—	G20
AP1_12															ADCA1I12		137	B16	—	C20
AP1_13															ADCA1I13		136	C15	—	B21
AP1_14															ADCA1I14		135	B17	—	D20
AP1_15															ADCA1I15		134	C16	—	B22

CAUTION

Use ADC functions with their initial settings. For details, see **Table 2A.78, Control Registers (AP1)**.

2A.10.20.2 Control Registers

Table 2A.78 Control Registers (AP1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			176 Pins	233 Pins	272 Pins	324 Pins
AP1	Analog port register 1	16	15-0	R/W	$00CC_H$	0000_H	✓	✓	—	✓
APSR1	Analog port set/reset register 1	32	31-16, 15-0	R/W	$01CC_H$	0000 0000_H	✓	✓	—	✓
APPR1	Analog port pin read register 1	16	15-0	R	$02CC_H$	0000_H	✓	✓	—	✓
APM1	Analog port mode register 1	16	15-0	R/W	$03CC_H$	$FFFF_H$	✓	✓	—	✓
APNOT1	Analog port NOT register 1	16	15-0	W	$07CC_H$	0000_H	✓	✓	—	✓
APMSR1	Analog port mode Set/reset register 1	32	31-16, 15-0	R/W	$08CC_H$	0000 $FFFF_H$	✓	✓	—	✓
APIBC1	Analog port input buffer control register 1	16	15-0	R/W	$40CC_H$	0000_H	✓	✓	—	✓
APBDC1	Analog port bidirection control register 1	16	15-0	R/W	$41CC_H$	0000_H	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2A.10.21 Input Port 0 (IP0)

2A.10.21.1 Alternative Function

Table 2A.79 Input Port 0 (IP0)

	Alternative Mode																PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				176 Pins	233 Pins	272 Pins	324 Pins
Port Mode	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	ADC	Special Function				
IP0_0																XT2	57	T7	—	AA8

2A.10.21.2 Control Registers

Table 2A.80 Control Registers (IP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*[1]			176 Pins	233 Pins	272 Pins	324 Pins
IPPR0	Input port pin read register 0	16	0	R	$02F0_H$	0000_H	✓	✓	—	✓
IPIBC0	Port input buffer control register 0	16	0	R/W	$40F0_H$	0000_H	✓	✓	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

CAUTION

When the IP0_0/XT2 pin is used as an input port, set the IPIBC0.0 bit to 1 and stop the SOSC operation. For details on the settings for SOSC operations, see **Section 12AB.4.2.7, SOSCE — SubOSC Enable Register**. When the IP0_0/XT2 pin is used for the SubOSC (SOSC) not as an input port, set the IPIBC0.0 bit to 0.

2A.11 Port (Special I/O) Function Overview

This section describes the port (special I/O) functions.

2A.11.1 Special I/O after Reset

The special port function after reset is deasserted is shown below.

2A.11.1.1 P8_6: $\overline{\mathrm{RESETOUT}}$

The P8_6 pin ($\overline{\mathrm{RESETOUT}}$ signal) changes PM8.PM8_6 and PODC8.PODC8_6 registers value after reset by OPBT0.RESETOUTEN setting.

The P8_6 pin outputs a low level while a reset is asserted, and pin status of after the reset is different.

(Case 1): OPBT0.RESETOUTEN = 1

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 0: Output mode
- PODC8.PODC8_6 = 1: Open-drain

(Case 2): OPBT0.RESETOUTEN = 0

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 1: Input mode
- PODC8.PODC8_6 = 0: Push-pull

For detail of OPBT0.RESETOUTEN register, see **Section 44.9.2, OPBT0 — Option Byte 0**, also see **Section 9A.1.3, Reset Output ($\overline{\mathrm{RESETOUT}}$)**.

When the P8_6 pin setting is updated with another value, the pin operates by new setting.

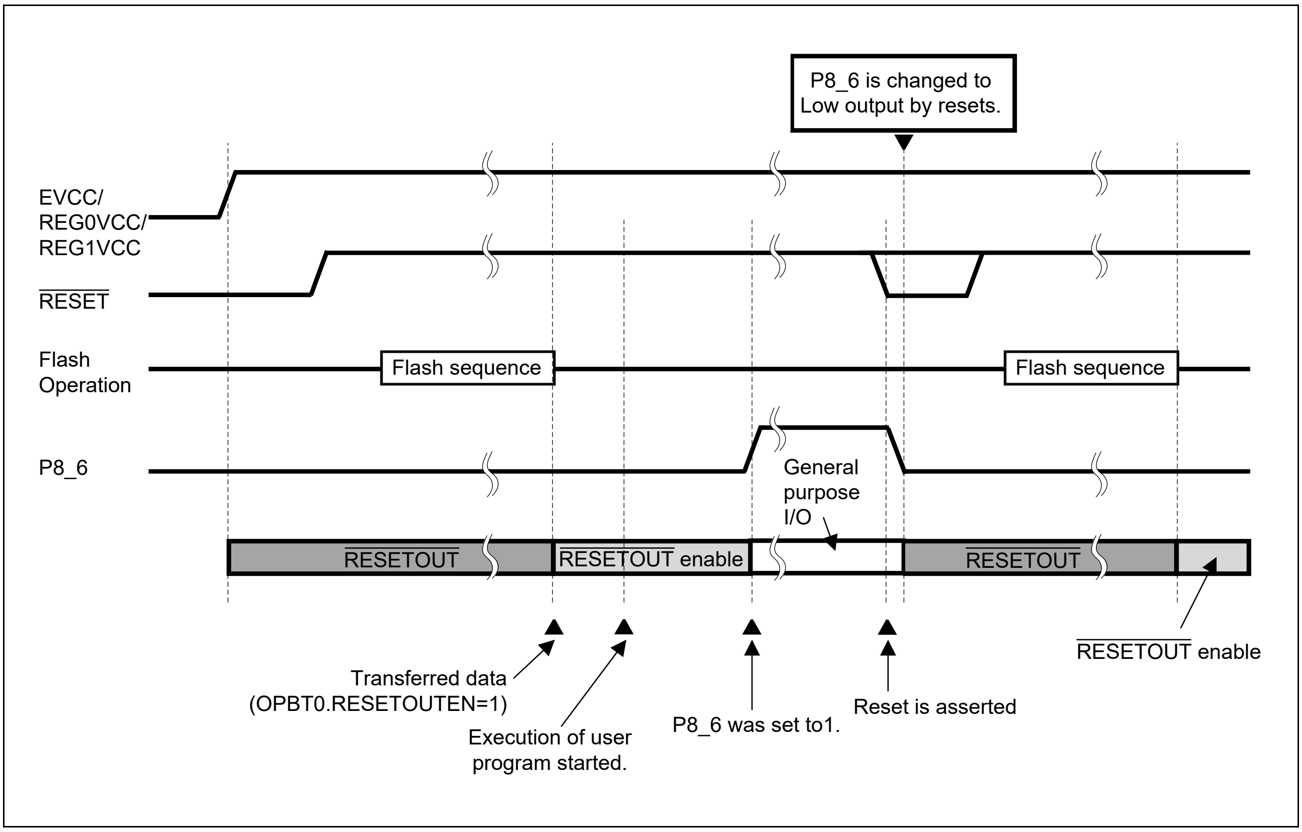
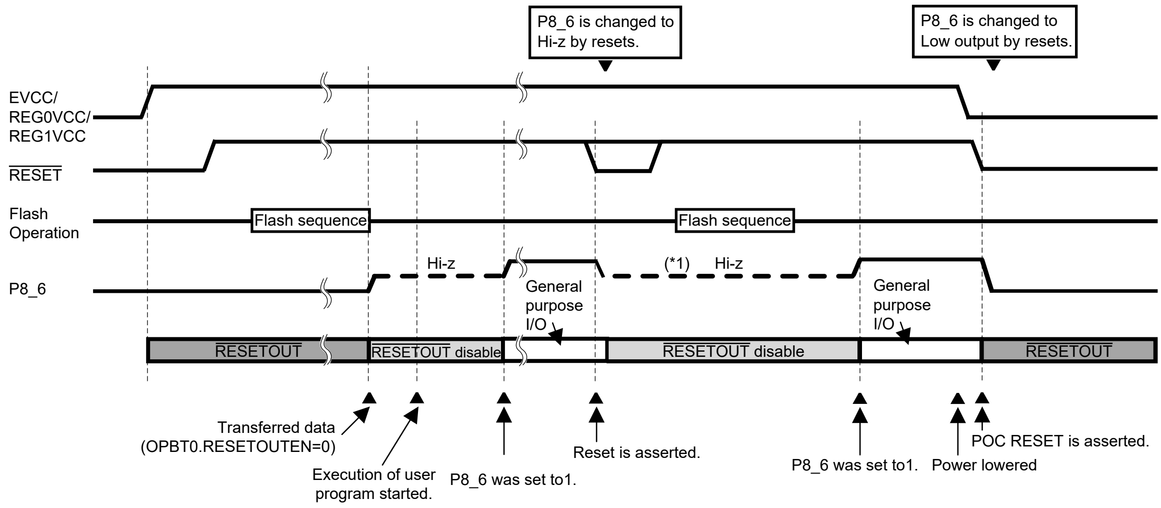


Figure 2A.10 P8_6 Pin (RESETOUT Signal) Operation While a Reset is asserted and released: (Case 1) OPBT0.RESETOUTEN setting is 1



Note 1. When a reset except POC reset occurs with $\overline{\text{RESETOUT}}$ disable (OPBT0.RESETOUTEN = 0), P8_6 pin ($\overline{\text{RESETOUT}}$ signal) will be changed to Hi-z.

Figure 2A.11 P8_6 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation While a Reset is asserted and released: (Case 2) OPBT0.RESETOUTEN setting is 0

2A.11.1.2 JP0_0 to JP0_6: Debug Interface

If the OPJTAG[1:0] setting is the combination below, the pins of the JTAG port group can be used as a debug interface after reset release.

Table 2A.81 Debug Interface

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5	JP0_6
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	$\overline{\text{DCUTRST}}$ input	$\overline{\text{DCURDY}}$ output	$\overline{\text{EVTO}}$ output
0	1	LPD (4 pins)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output	Port/ alternative function
1	0	LPD (1 pin)	LPDIO input/output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

NOTE

For the OPJTAG [1:0] settings, see **Section 44.9.2, OPBT0 — Option Byte 0**.

2A.11.1.3 FPDR(JP0_0), FPDT(JP0_1), FPCK(JP0_2): Flash Programmer

These pins are used for connecting a flash programmer. See *Flash Programmer's Manual* for details.

2A.11.1.4 Mode Pins

The FLMD0 pin in combination with the P10_8: FLMD1 pin can select serial programming mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_2: MODE1 and the P10_1: MODE0 pins can select boundary scan mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_6: MODE2, the P10_2: MODE1 and the P10_1: MODE0 pins can select user boot mode.

For details on the mode selection, see **Section 6, Operating Mode**.

2A.11.1.5 IP0_0: XT2

This pin is the SubOSC (SOSC) input pin. When the IPIBC0_0 bit = 1, the IP0_0/XT2 pin is used as an input port. If you make this setting, stop SOSC operation at the same time.

2A.11.2 A/D Input Alternative I/O

The following ports are permanently connected to A/D input functions. (However, an analog input to the A/D is controlled by the A/D module.)

Table 2A.82 A/D Input Alternative Pins

Port	A/D Input	Device			
		176 Pins	233 Pins	272 Pins	324 Pins
P8_0	ADCA0I0S	✓	✓	—	✓
P8_1	ADCA0I1S	✓	✓	—	✓
P8_2	ADCA0I4S	✓	✓	—	✓
P8_3	ADCA0I5S	✓	✓	—	✓
P8_4	ADCA0I6S	✓	✓	—	✓
P8_5	ADCA0I7S	✓	✓	—	✓
P8_6	ADCA0I8S	✓	✓	—	✓
P8_7	ADCA0I14S	✓	✓	—	✓
P8_8	ADCA0I15S	✓	✓	—	✓
P8_9	ADCA0I16S	✓	✓	—	✓
P8_10	ADCA0I17S	✓	✓	—	✓
P8_11	ADCA0I18S	✓	✓	—	✓
P8_12	ADCA0I19S	✓	✓	—	✓
P9_0	ADCA0I2S	✓	✓	—	✓
P9_1	ADCA0I3S	✓	✓	—	✓
P9_2	ADCA0I9S	✓	✓	—	✓
P9_3	ADCA0I10S	✓	✓	—	✓
P9_4	ADCA0I11S	✓	✓	—	✓
P18_0	ADCA1I0S	✓	✓	—	✓
P18_1	ADCA1I1S	✓	✓	—	✓
P18_2	ADCA1I2S	✓	✓	—	✓
P18_3	ADCA1I3S	✓	✓	—	✓
P18_4	ADCA1I4S	✓	✓	—	✓
P18_5	ADCA1I5S	✓	✓	—	✓
P18_6	ADCA1I6S	✓	✓	—	✓
P18_7	ADCA1I7S	✓	✓	—	✓
P18_8	ADCA1I8S	—	✓	—	✓
P18_9	ADCA1I9S	—	✓	—	✓
P18_10	ADCA1I10S	—	✓	—	✓
P18_11	ADCA1I11S	—	✓	—	✓
P18_12	ADCA1I12S	—	✓	—	✓
P18_13	ADCA1I13S	—	✓	—	✓
P18_14	ADCA1I14S	—	✓	—	✓
P18_15	ADCA1I15S	—	✓	—	✓
P19_0	ADCA1I16S	—	✓	—	✓
P19_1	ADCA1I17S	—	✓	—	✓
P19_2	ADCA1I18S	—	✓	—	✓
P19_3	ADCA1I19S	—	✓	—	✓
AP0_0	ADCA0I0	✓	✓	—	✓
AP0_1	ADCA0I1	✓	✓	—	✓

Table 2A.82 A/D Input Alternative Pins

Port	A/D Input	Device			
		176 Pins	233 Pins	272 Pins	324 Pins
AP0_2	ADCA0I2	✓	✓	—	✓
AP0_3	ADCA0I3	✓	✓	—	✓
AP0_4	ADCA0I4	✓	✓	—	✓
AP0_5	ADCA0I5	✓	✓	—	✓
AP0_6	ADCA0I6	✓	✓	—	✓
AP0_7	ADCA0I7	✓	✓	—	✓
AP0_8	ADCA0I8	✓	✓	—	✓
AP0_9	ADCA0I9	✓	✓	—	✓
AP0_10	ADCA0I10	✓	✓	—	✓
AP0_11	ADCA0I11	✓	✓	—	✓
AP0_12	ADCA0I12	✓	✓	—	✓
AP0_13	ADCA0I13	✓	✓	—	✓
AP0_14	ADCA0I14	✓	✓	—	✓
AP0_15	ADCA0I15	✓	✓	—	✓
AP1_0	ADCA1I0	✓	✓	—	✓
AP1_1	ADCA1I1	✓	✓	—	✓
AP1_2	ADCA1I2	✓	✓	—	✓
AP1_3	ADCA1I3	✓	✓	—	✓
AP1_4	ADCA1I4	✓	✓	—	✓
AP1_5	ADCA1I5	✓	✓	—	✓
AP1_6	ADCA1I6	✓	✓	—	✓
AP1_7	ADCA1I7	✓	✓	—	✓
AP1_8	ADCA1I8	✓	✓	—	✓
AP1_9	ADCA1I9	✓	✓	—	✓
AP1_10	ADCA1I10	✓	✓	—	✓
AP1_11	ADCA1I11	✓	✓	—	✓
AP1_12	ADCA1I12	✓	✓	—	✓
AP1_13	ADCA1I13	✓	✓	—	✓
AP1_14	ADCA1I14	✓	✓	—	✓
AP1_15	ADCA1I15	✓	✓	—	✓

2A.11.3 Special I/O Control

2A.11.3.1 Direct I/O Control (PIPC)

Some alternative functions take over the input and output control of the ports.

The following table lists all alternative functions where PIPCn.PIPCn_m must be set to 1.

For details, see **Section 2A.9.2.3, PIPCn — Port IP Control Register**.

Table 2A.83 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1

Function	Alternative functions Name	Port Name	Power Supply Area	Control	Reference Section
MEMC	MEMC0AD0	P10_6	ISO		**Section 16**
	MEMC0AD1	P10_7	ISO		
	MEMC0AD2	P10_8	ISO		
	MEMC0AD3	P10_9	ISO		
	MEMC0AD4	P10_10	ISO		
	MEMC0AD5	P10_11	ISO		
	MEMC0AD6	P10_12	ISO		
	MEMC0AD7	P10_13	ISO		
	MEMC0AD8	P10_14	ISO		
	MEMC0AD9	P11_1	ISO		
	MEMC0AD10	P11_2	ISO		
	MEMC0AD11	P11_3	ISO		
	MEMC0AD12	P11_4	ISO		
	MEMC0AD13	P11_5	ISO		
	MEMC0AD14	P11_6	ISO		
	MEMC0AD15	P11_7	ISO		
TAPA	TAPA0UP	P10_0	ISO	U phase Hi-Z control	**Section 36**
	TAPA0UN	P10_1	ISO		
	TAPA0VP	P10_2	ISO	V phase Hi-Z control	
	TAPA0VN	P10_3	ISO		
	TAPA0WP	P10_4	ISO	W phase Hi-Z control	
	TAPA0WN	P10_5	ISO		
CSIG	CSIG0SO	P0_13	AWO	Serial data output control signal	**Section 19**
		P10_6	ISO		
	CSIG0SC	P0_14	AWO	Master (1) / slave (0) mode signal	
		P10_7	ISO		
	CSIG1SO	P11_9	ISO	Serial data output control signal	
	CSIG1SC	P11_10	ISO	Master (1) / slave (0) mode signal	
	CSIG2SO	P12_5	ISO	Serial data output control signal	
	CSIG2SC	P12_4	ISO	Master (1) / slave (0) mode signal	
	CSIG3SO	P20_1	ISO	Serial data output control signal	
	CSIG3SC	P20_2	ISO	Master (1) / slave (0) mode signal	
	CSIG4SO	P1_3	AWO	Serial data output control signal	
		P23_6	ISO		
	CSIG4SC	P1_1	AWO	Master (1) / slave (0) mode signal	
		P23_8	ISO		

Table 2A.83 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1

Function	Alternative functions Name	Port Name	Power Supply Area	Control	Reference Section
CSIH	CSIH0SO	P0_3	AWO	Serial data output control signal	**Section 20**
	CSIH0SC	P0_2	AWO	Master (1) / slave (0) mode signal	
	CSIH1SO	P0_5	AWO	Serial data output control signal	
		P10_2	ISO		
	CSIH1SC	P0_6	AWO	Master (1) / slave (0) mode signal	
		P10_1	ISO		
	CSIH2SO	P11_2	ISO	Serial data output control signal	
	CSIH2SC	P11_3	ISO	Master (1) / slave (0) mode signal	
	CSIH3SO	P11_6	ISO	Serial data output control signal	
	CSIH3SC	P11_7	ISO	Master (1) / slave (0) mode signal	
	CSIH4SO	P2_4	AWO	Serial data output control signal	
		P23_1	ISO		
	CSIH4SC	P1_5	AWO	Master (1) / slave (0) mode signal	
		P23_3	ISO		
SFMA	SFMA0IO0	P11_5	ISO	SPIch.0 MOSI0_IO00 output enable	**Section 17**
	SFMA0IO1	P11_4	ISO	SPIch.0 MOSI0_IO10 output enable	
	SFMA0IO2	P11_3	ISO	SPIch.0 IO20 output enable	
	SFMA0IO3	P11_2	ISO	SPIch.0 IO30 output enable	
ETNB	ETNB0MDIO	P12_4	ISO	MDIO output enable	**Section 26**
	ETNB1MDIO	P21_7	ISO	MDIO output enable	
MMCA	MMCA0CMD	P22_7	ISO	MMCA0CMD output enable	**Section 18**
	MMCA0DAT0	P22_9	ISO	MMCA0DAT0 output enable	
	MMCA0DAT1	P22_10	ISO	MMCA0DAT1 output enable	
	MMCA0DAT2	P22_11	ISO	MMCA0DAT2 output enable	
	MMCA0DAT3	P22_12	ISO	MMCA0DAT3 output enable	
	MMCA0DAT4	P22_13	ISO	MMCA0DAT4 output enable	
	MMCA0DAT5	P22_14	ISO	MMCA0DAT5 output enable	
	MMCA0DAT6	P22_15	ISO	MMCA0DAT6 output enable	
	MMCA0DAT7	P21_1	ISO	MMCA0DAT7 output enable	

2A.11.3.2 Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)

The port input buffer characteristics (Type 1 or Type 2) of this device can be selected using the PISn/PISAn/JPIS0 register. The applicable pins are shown in the following table.

The JTAG port input buffer characteristics (Type 1/2 or Type 5) of this device can be selected using the JPISA0 register. The applicable pins are shown in **Table 2A.85, JTAG Port Input Buffer Characteristics Selection**.

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P0_0	SHMT1	SHMT4	—	✓	✓	—	✓
P0_1	SHMT1	SHMT4	—	✓	✓	—	✓
P0_2	SHMT1	SHMT4	—	✓	✓	—	✓
P0_3	SHMT1	SHMT4	—	✓	✓	—	✓
P0_4	SHMT1	SHMT4	—	✓	✓	—	✓
P0_5	SHMT1	SHMT4	—	✓	✓	—	✓
P0_6	SHMT1	SHMT4	—	✓	✓	—	✓
P0_7	SHMT1	SHMT4	—	✓	✓	—	✓
P0_8	SHMT1	SHMT4	—	✓	✓	—	✓
P0_9	SHMT1	SHMT4	—	✓	✓	—	✓
P0_10	SHMT1	SHMT4	—	✓	✓	—	✓
P0_11	SHMT1	SHMT4	—	✓	✓	—	✓
P0_12	SHMT1	SHMT4	—	✓	✓	—	✓
P0_13	SHMT1	SHMT4	—	✓	✓	—	✓
P0_14	SHMT1	SHMT4	—	✓	✓	—	✓
P1_0	SHMT1	SHMT4	—	✓	✓	—	✓
P1_1	SHMT1	SHMT4	—	✓	✓	—	✓
P1_2	SHMT1	SHMT4	—	✓	✓	—	✓
P1_3	SHMT1	SHMT4	—	✓	✓	—	✓
P1_4	SHMT1	SHMT4	—	✓	✓	—	✓
P1_5	SHMT1	SHMT4	—	✓	✓	—	✓
P1_8	SHMT1	SHMT4	—	✓	✓	—	✓
P1_9	SHMT1	SHMT4	—	✓	✓	—	✓
P1_10	SHMT1	SHMT4	—	✓	✓	—	✓
P1_11	SHMT1	SHMT4	—	✓	✓	—	✓
P1_12	SHMT1	SHMT4	—	✓	✓	—	✓
P1_13	SHMT1	SHMT4	—	✓	✓	—	✓
P1_14	SHMT1	SHMT4	—	✓	✓	—	✓
P1_15	SHMT1	SHMT4	—	✓	✓	—	✓
P2_0	SHMT1	SHMT4	—	✓	✓	—	✓
P2_1	SHMT1	SHMT4	—	✓	✓	—	✓
P2_2	SHMT1	SHMT4	—	✓	✓	—	✓
P2_3	SHMT1	SHMT4	—	✓	✓	—	✓
P2_4	SHMT1	SHMT4	—	✓	✓	—	✓
P2_5	SHMT1	SHMT4	—	✓	✓	—	✓
P2_6	SHMT1	SHMT4	—	✓	✓	—	✓

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P2_7	SHMT1	SHMT4	—	—	✓	—	✓
P2_8	SHMT1	SHMT4	—	—	✓	—	✓
P2_9	SHMT1	SHMT4	—	—	✓	—	✓
P2_10	SHMT1	SHMT4	—	—	✓	—	✓
P2_11	SHMT1	SHMT4	—	—	✓	—	✓
P2_12	SHMT1	SHMT4	—	—	✓	—	✓
P2_13	SHMT1	SHMT4	—	—	✓	—	✓
P2_14	SHMT1	SHMT4	—	—	✓	—	✓
P2_15	SHMT1	SHMT4	—	—	✓	—	✓
P3_0	SHMT1	SHMT4	—	—	✓	—	✓
P3_1	SHMT1	SHMT4	—	—	—	—	✓
P3_2	SHMT1	SHMT4	—	—	—	—	✓
P3_3	SHMT1	SHMT4	—	—	—	—	✓
P3_4	SHMT1	SHMT4	—	—	—	—	✓
P3_5	SHMT1	SHMT4	—	—	—	—	✓
P3_6	SHMT1	SHMT4	—	—	—	—	✓
P3_7	SHMT1	SHMT4	—	—	—	—	✓
P3_8	SHMT1	SHMT4	—	—	—	—	✓
P3_9	SHMT1	SHMT4	—	—	—	—	✓
P3_10	SHMT1	SHMT4	—	—	—	—	✓
P3_11	SHMT1	SHMT4	—	—	—	—	✓
P3_12	SHMT1	SHMT4	—	—	—	—	✓
P8_0	SHMT1	SHMT4	—	✓	✓	—	✓
P8_1	SHMT1	SHMT4	—	✓	✓	—	✓
P8_2	SHMT1	SHMT4	—	✓	✓	—	✓
P8_3	SHMT1	SHMT4	—	✓	✓	—	✓
P8_4	SHMT1	SHMT4	—	✓	✓	—	✓
P8_5	SHMT1	SHMT4	—	✓	✓	—	✓
P8_6	SHMT1	SHMT4	—	✓	✓	—	✓
P8_7	SHMT1	SHMT4	—	✓	✓	—	✓
P8_8	SHMT1	SHMT4	—	✓	✓	—	✓
P8_9	SHMT1	SHMT4	—	✓	✓	—	✓
P8_10	SHMT1	SHMT4	—	✓	✓	—	✓
P8_11	SHMT1	SHMT4	—	✓	✓	—	✓
P8_12	SHMT1	SHMT4	—	✓	✓	—	✓
P9_0	SHMT1	SHMT4	—	✓	✓	—	✓
P9_1	SHMT1	SHMT4	—	✓	✓	—	✓
P9_2	SHMT1	SHMT4	—	✓	✓	—	✓
P9_3	SHMT1	SHMT4	—	✓	✓	—	✓
P9_4	SHMT1	SHMT4	—	✓	✓	—	✓
P10_0	SHMT1	SHMT4	TTL	✓	✓	—	✓
P10_1	SHMT1	SHMT4	TTL	✓	✓	—	✓

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P10_2	SHMT1	SHMT4	TTL	✓	✓	—	✓
P10_3	SHMT1	SHMT4	—	✓	✓	—	✓
P10_4	SHMT1	SHMT4	TTL	✓	✓	—	✓
P10_5	SHMT1	SHMT4	TTL	✓	✓	—	✓
P10_6	SHMT1	SHMT4	—	✓	✓	—	✓
P10_7	SHMT1	SHMT4	—	✓	✓	—	✓
P10_8	SHMT1	SHMT4	—	✓	✓	—	✓
P10_9	SHMT1	SHMT4	—	✓	✓	—	✓
P10_10	SHMT1	SHMT4	—	✓	✓	—	✓
P10_11	SHMT1	SHMT4	—	✓	✓	—	✓
P10_12	SHMT1	SHMT4	—	✓	✓	—	✓
P10_13	SHMT1	SHMT4	—	✓	✓	—	✓
P10_14	SHMT1	SHMT4	—	✓	✓	—	✓
P10_15	SHMT1	SHMT4	—	✓	✓	—	✓
P11_0	SHMT1	SHMT4	—	✓	✓	—	✓
P11_1	SHMT1	SHMT4	—	✓	✓	—	✓
P11_2	SHMT1	SHMT4	—	✓	✓	—	✓
P11_3	SHMT1	SHMT4	—	✓	✓	—	✓
P11_4	SHMT1	SHMT4	—	✓	✓	—	✓
P11_5	SHMT1	SHMT4	—	✓	✓	—	✓
P11_6	SHMT1	SHMT4	—	✓	✓	—	✓
P11_7	SHMT1	SHMT4	—	✓	✓	—	✓
P11_8	SHMT1	SHMT4	—	✓	✓	—	✓
P11_9	SHMT1	SHMT4	—	✓	✓	—	✓
P11_10	SHMT1	SHMT4	TTL	✓	✓	—	✓
P11_11	SHMT1	SHMT4	TTL	✓	✓	—	✓
P11_12	SHMT1	SHMT4	TTL	✓	✓	—	✓
P11_15	SHMT1	SHMT4	TTL	✓	✓	—	✓
P12_0	SHMT1	SHMT4	—	✓	✓	—	✓
P12_1	SHMT1	SHMT4	—	✓	✓	—	✓
P12_2	SHMT1	SHMT4	—	✓	✓	—	✓
P12_3	SHMT1	SHMT4	—	✓	✓	—	✓
P12_4	SHMT1	SHMT4	TTL	✓	✓	—	✓
P12_5	SHMT1	SHMT4	—	✓	✓	—	✓
P13_0	SHMT1	SHMT4	—	—	✓	—	✓
P13_1	SHMT1	SHMT4	—	—	✓	—	✓
P13_2	SHMT1	SHMT4	TTL	—	✓	—	✓
P13_3	SHMT1	SHMT4	TTL	—	✓	—	✓

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P13_4	SHMT1	SHMT4	—	—	✓	—	—
			TTL	—	—	—	✓
P13_5	SHMT1	SHMT4	TTL	—	✓	—	✓
P13_6	SHMT1	SHMT4	—	—	✓	—	✓
P13_7	SHMT1	SHMT4	—	—	✓	—	✓
P18_0	SHMT1	SHMT4	TTL	✓	✓	—	✓
P18_1	SHMT1	SHMT4	—	✓	✓	—	✓
P18_2	SHMT1	SHMT4	—	✓	✓	—	✓
P18_3	SHMT1	SHMT4	—	✓	✓	—	✓
P18_4	SHMT1	SHMT4	—	✓	✓	—	✓
P18_5	SHMT1	SHMT4	—	✓	✓	—	✓
P18_6	SHMT1	SHMT4	—	✓	✓	—	✓
P18_7	SHMT1	SHMT4	TTL	✓	✓	—	✓
P18_8	SHMT1	SHMT4	TTL	—	✓	—	✓
P18_9	SHMT1	SHMT4	TTL	—	✓	—	✓
P18_10	SHMT1	SHMT4	—	—	✓	—	✓
P18_11	SHMT1	SHMT4	—	—	✓	—	✓
P18_12	SHMT1	SHMT4	—	—	✓	—	✓
P18_13	SHMT1	SHMT4	—	—	✓	—	✓
P18_14	SHMT1	SHMT4	—	—	✓	—	✓
P18_15	SHMT1	SHMT4	—	—	✓	—	✓
P19_0	SHMT1	SHMT4	—	—	✓	—	✓
P19_1	SHMT1	SHMT4	—	—	✓	—	✓
P19_2	SHMT1	SHMT4	—	—	✓	—	✓
P19_3	SHMT1	SHMT4	—	—	✓	—	✓
P20_0	SHMT1	SHMT4	—	✓	✓	—	✓
P20_1	SHMT1	SHMT4	—	✓	✓	—	✓
P20_2	SHMT1	SHMT4	—	✓	✓	—	✓
P20_3	SHMT1	SHMT4	—	✓	✓	—	✓
P20_4	SHMT1	SHMT4	—	✓	✓	—	✓
P20_5	SHMT1	SHMT4	—	✓	✓	—	✓
P20_6	SHMT1	SHMT4	—	—	—	—	✓
P20_7	SHMT1	SHMT4	—	—	—	—	✓
P20_8	SHMT1	SHMT4	—	—	—	—	✓
P20_9	SHMT1	SHMT4	—	—	—	—	✓
P20_10	SHMT1	SHMT4	—	—	—	—	✓
P20_11	SHMT1	SHMT4	—	—	—	—	✓
P20_12	SHMT1	SHMT4	—	—	—	—	✓
P20_13	SHMT1	SHMT4	—	—	—	—	✓
P20_14	SHMT1	SHMT4	—	—	—	—	✓
P20_15	SHMT1	SHMT4	—	—	—	—	✓

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P21_0	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P21_1	SHMT1	SHMT4	—	—	—	—	✓
P21_2	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P21_3	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P21_4	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P21_5	SHMT1	SHMT4	TTL	—	—	—	✓
P21_6	SHMT1	SHMT4	—	—	—	—	✓
P21_7	SHMT1	SHMT4	TTL	—	—	—	✓
P21_8	SHMT1	SHMT4	TTL	—	—	—	✓
P21_9	SHMT1	SHMT4	TTL	—	—	—	✓
P21_10	SHMT1	SHMT4	TTL	—	—	—	✓
P21_11	SHMT1	SHMT4	—	—	—	—	✓
P21_12	SHMT1	SHMT4	—	—	—	—	✓
P21_13	SHMT1	SHMT4	—	—	—	—	✓
P21_14	SHMT1	SHMT4	—	—	—	—	✓
P22_0	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P22_1	SHMT1	SHMT4	—	—	—	—	✓
P22_2	SHMT1	SHMT4	—	—	—	—	✓
P22_3	SHMT1	SHMT4	—	—	—	—	✓
P22_4	SHMT1	SHMT4	—	—	—	—	✓
P22_5	SHMT1	SHMT4	—	—	—	—	✓
P22_6	SHMT1	SHMT4	—	—	—	—	—
			TTL	—	—	—	✓
P22_7	SHMT1	SHMT4	—	—	—	—	✓
P22_8	SHMT1	SHMT4	—	—	—	—	✓
P22_9	SHMT1	SHMT4	—	—	—	—	✓
P22_10	SHMT1	SHMT4	—	—	—	—	✓
P22_11	SHMT1	SHMT4	—	—	—	—	✓
P22_12	SHMT1	SHMT4	—	—	—	—	✓
P22_13	SHMT1	SHMT4	—	—	—	—	✓
P22_14	SHMT1	SHMT4	—	—	—	—	✓
P22_15	SHMT1	SHMT4	—	—	—	—	✓

Table 2A.84 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device			
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P23_0	SHMT1	SHMT4	—	—	—	—	✓
P23_1	SHMT1	SHMT4	—	—	—	—	✓
P23_2	SHMT1	SHMT4	—	—	—	—	✓
P23_3	SHMT1	SHMT4	—	—	—	—	✓
P23_4	SHMT1	SHMT4	—	—	—	—	✓
P23_5	SHMT1	SHMT4	—	—	—	—	✓
P23_6	SHMT1	SHMT4	—	—	—	—	✓
P23_7	SHMT1	SHMT4	—	—	—	—	✓
P23_8	SHMT1	SHMT4	—	—	—	—	✓
P23_9	SHMT1	SHMT4	—	—	—	—	✓
P23_10	SHMT1	SHMT4	—	—	—	—	✓
P24_0	SHMT1	SHMT4	—	—	—	—	✓
P24_1	SHMT1	SHMT4	—	—	—	—	✓
P24_2	SHMT1	SHMT4	—	—	—	—	✓
P24_3	SHMT1	SHMT4	—	—	—	—	✓
P24_4	SHMT1	SHMT4	—	—	—	—	✓
P24_5	SHMT1	SHMT4	—	—	—	—	✓
P24_6	SHMT1	SHMT4	—	—	—	—	✓
P24_7	SHMT1	SHMT4	—	—	—	—	✓

Table 2A.85 JTAG Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Devices			
	Type 1 (JPIS0_m = 0 & JPISA0_m = 0)	Type 2 (JPIS0_m = 1 & JPISA0_m = 0)	Type 5 (JPISA0_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
JP0_0	SHMT1	SHMT4	TTL[*1,*2,*3,*4]	✓	✓	—	✓
JP0_1	SHMT1	SHMT4	—	✓	✓	—	✓
JP0_2	SHMT1	SHMT4	TTL[*1,*2,*3]	✓	✓	—	✓
JP0_3	SHMT1	SHMT4	TTL[*1,*2]	✓	✓	—	✓
JP0_4	—	SHMT4	—[*1,*2]	✓	✓	—	✓
JP0_5	SHMT1	SHMT4	—	✓	✓	—	✓
JP0_6	SHMT1	SHMT4	—	✓	✓	—	✓

Note 1. TTL is selected for Boundary scan mode without JPISA0 register setting.

Note 2. TTL is selected for Nexus in normal operating mode without JPISA0 register setting.

Note 3. TTL is selected for LPD (4 pins) in normal operating mode without JPISA0 register setting.

Note 4. TTL is selected for LPD (1 pin) in normal operating mode without JPISA0 register setting.

NOTES

1. For the SHMT1, SHMT4 and TTL pin characteristics, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**.
2. For the input buffer after reset, Type 2 (SHMT4) is selected.

2A.11.3.3 Output Buffer Control (PDSC)

The port output driver strength (slow mode/fast mode) can be selected using the PDSCn register. The applicable pins are shown in the following table. Only slow mode is supported for ports other than those listed below.

Table 2A.86 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
JP0_1	10 MHz	40 MHz	✓	✓	—	✓
JP0_2	10 MHz	40 MHz	✓	✓	—	✓
JP0_3	10 MHz	40 MHz	✓	✓	—	✓
JP0_5	10 MHz	40 MHz	✓	✓	—	✓
JP0_6	10 MHz	40 MHz	✓	✓	—	✓
P0_0	10 MHz	40 MHz	✓	✓	—	✓
P0_1	10 MHz	40 MHz	✓	✓	—	✓
P0_2	10 MHz	40 MHz*1	✓	✓	—	✓
P0_3	10 MHz	40 MHz*1	✓	✓	—	✓
P0_4	10 MHz	40 MHz	✓	✓	—	✓
P0_5	10 MHz	40 MHz*2	✓	✓	—	✓
P0_6	10 MHz	40 MHz*2	✓	✓	—	✓
P0_7	10 MHz	40 MHz	✓	✓	—	✓
P0_8	10 MHz	40 MHz	✓	✓	—	✓
P0_9	10 MHz	40 MHz	✓	✓	—	✓
P0_10	10 MHz	40 MHz	✓	✓	—	✓
P0_11	10 MHz	40 MHz	✓	✓	—	✓
P0_12	10 MHz	40 MHz	✓	✓	—	✓
P0_13	10 MHz	40 MHz	✓	✓	—	✓
P0_14	10 MHz	40 MHz	✓	✓	—	✓
P1_0	10 MHz	40 MHz	✓	✓	—	✓
P1_1	10 MHz	40 MHz	✓	✓	—	✓
P1_2	10 MHz	40 MHz	✓	✓	—	✓
P1_3	10 MHz	40 MHz	✓	✓	—	✓
P1_4	10 MHz	40 MHz	✓	✓	—	✓
P1_5	10 MHz	40 MHz*2	✓	✓	—	✓
P1_8	10 MHz	40 MHz	✓	✓	—	✓
P1_9	10 MHz	40 MHz	✓	✓	—	✓
P1_10	10 MHz	40 MHz	✓	✓	—	✓
P1_11	10 MHz	40 MHz	✓	✓	—	✓
P1_12	10 MHz	40 MHz	✓	✓	—	✓
P1_13	10 MHz	40 MHz	✓	✓	—	✓
P1_14	10 MHz	40 MHz	✓	✓	—	✓
P1_15	10 MHz	40 MHz	✓	✓	—	✓
P2_0	10 MHz	40 MHz	✓	✓	—	✓
P2_1	10 MHz	40 MHz	✓	✓	—	✓
P2_2	10 MHz	40 MHz	✓	✓	—	✓
P2_3	10 MHz	40 MHz	✓	✓	—	✓
P2_4	10 MHz	40 MHz*2	✓	✓	—	✓
P2_5	10 MHz	40 MHz	✓	✓	—	✓

Table 2A.86 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P2_6	10 MHz	40 MHz	✓	✓	—	✓
P2_7	10 MHz	40 MHz	—	✓	—	✓
P2_8	10 MHz	40 MHz	—	✓	—	✓
P2_9	10 MHz	40 MHz	—	✓	—	✓
P2_10	10 MHz	40 MHz	—	✓	—	✓
P2_11	10 MHz	40 MHz	—	✓	—	✓
P2_12	10 MHz	40 MHz	—	✓	—	✓
P2_13	10 MHz	40 MHz	—	✓	—	✓
P2_14	10 MHz	40 MHz	—	✓	—	✓
P2_15	10 MHz	40 MHz	—	✓	—	✓
P3_0	10 MHz	40 MHz	—	✓	—	✓
P3_1	10 MHz	40 MHz	—	—	—	✓
P3_2	10 MHz	40 MHz	—	—	—	✓
P3_3	10 MHz	40 MHz	—	—	—	✓
P3_4	10 MHz	40 MHz	—	—	—	✓
P3_5	10 MHz	40 MHz	—	—	—	✓
P3_6	10 MHz	40 MHz	—	—	—	✓
P3_7	10 MHz	40 MHz	—	—	—	✓
P3_8	10 MHz	40 MHz	—	—	—	✓
P3_9	10 MHz	40 MHz	—	—	—	✓
P3_10	10 MHz	40 MHz	—	—	—	✓
P3_11	10 MHz	40 MHz	—	—	—	✓
P3_12	10 MHz	40 MHz	—	—	—	✓
P10_0	10 MHz	40 MHz	✓	✓	—	✓
P10_1	10 MHz	40 MHz*2	✓	✓	—	✓
P10_2	10 MHz	40 MHz*2	✓	✓	—	✓
P10_3	10 MHz	40 MHz	✓	✓	—	✓
P10_4	10 MHz	40 MHz	✓	✓	—	✓
P10_5	10 MHz	40 MHz	✓	✓	—	✓
P10_6	10 MHz	40 MHz	✓	✓	—	✓
P10_7	10 MHz	40 MHz	✓	✓	—	✓
P10_8	10 MHz	40 MHz	✓	✓	—	✓
P10_9	10 MHz	40 MHz	✓	✓	—	✓
P10_10	10 MHz	40 MHz	✓	✓	—	✓
P10_11	10 MHz	40 MHz	✓	✓	—	✓
P10_12	10 MHz	40 MHz	✓	✓	—	✓
P10_13	10 MHz	40 MHz	✓	✓	—	✓
P10_14	10 MHz	40 MHz	✓	✓	—	✓
P10_15	10 MHz	40 MHz	✓	✓	—	✓
P11_0	10 MHz	40 MHz	✓	✓	—	✓
P11_1	10 MHz	40 MHz	✓	✓	—	✓
P11_2	10 MHz	40 MHz*2	✓	✓	—	✓
P11_3	10 MHz	40 MHz*2	✓	✓	—	✓

Table 2A.86 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P11_4	10 MHz	40 MHz	✓	✓	—	✓
P11_5	10 MHz	40 MHz	✓	✓	—	✓
P11_6	10 MHz	40 MHz*2	✓	✓	—	✓
P11_7	10 MHz	40 MHz*2	✓	✓	—	✓
P11_8	10 MHz	40 MHz	✓	✓	—	✓
P11_9	10 MHz	40 MHz	✓	✓	—	✓
P11_10	10 MHz	40 MHz	✓	✓	—	✓
P11_11	10 MHz	40 MHz	✓	✓	—	✓
P11_12	10 MHz	40 MHz	✓	✓	—	✓
P11_15	10 MHz	40 MHz	✓	✓	—	✓
P12_0	10 MHz	40 MHz	✓	✓	—	✓
P12_1	10 MHz	40 MHz	✓	✓	—	✓
P12_2	10 MHz	40 MHz	✓	✓	—	✓
P12_3	10 MHz	40 MHz	✓	✓	—	✓
P12_4	10 MHz	40 MHz	✓	✓	—	✓
P12_5	10 MHz	40 MHz	✓	✓	—	✓
P13_0	10 MHz	40 MHz	—	✓	—	✓
P13_1	10 MHz	40 MHz	—	✓	—	✓
P13_2	10 MHz	40 MHz	—	✓	—	✓
P13_3	10 MHz	40 MHz	—	✓	—	✓
P13_4	10 MHz	40 MHz	—	✓	—	✓
P13_5	10 MHz	40 MHz	—	✓	—	✓
P13_6	10 MHz	40 MHz	—	✓	—	✓
P13_7	10 MHz	40 MHz	—	✓	—	✓
P18_0	10 MHz	40 MHz	✓	✓	—	✓
P18_1	10 MHz	40 MHz	✓	✓	—	✓
P18_2	10 MHz	40 MHz	✓	✓	—	✓
P18_3	10 MHz	40 MHz	✓	✓	—	✓
P18_4	10 MHz	40 MHz	✓	✓	—	✓
P18_5	10 MHz	40 MHz	✓	✓	—	✓
P18_6	10 MHz	40 MHz	✓	✓	—	✓
P18_7	10 MHz	40 MHz	✓	✓	—	✓
P18_8	10 MHz	40 MHz	—	✓	—	✓
P18_9	10 MHz	40 MHz	—	✓	—	✓
P18_10	10 MHz	40 MHz	—	✓	—	✓
P18_11	10 MHz	40 MHz	—	✓	—	✓
P18_12	10 MHz	40 MHz	—	✓	—	✓
P18_13	10 MHz	40 MHz	—	✓	—	✓
P18_14	10 MHz	40 MHz	—	✓	—	✓
P18_15	10 MHz	40 MHz	—	✓	—	✓
P19_0	10 MHz	40 MHz	—	✓	—	✓
P19_1	10 MHz	40 MHz	—	✓	—	✓
P19_2	10 MHz	40 MHz	—	✓	—	✓

Table 2A.86 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P19_3	10 MHz	40 MHz	—	✓	—	✓
P20_0	10 MHz	40 MHz	✓	✓	—	✓
P20_1	10 MHz	40 MHz	✓	✓	—	✓
P20_2	10 MHz	40 MHz	✓	✓	—	✓
P20_3	10 MHz	40 MHz	✓	✓	—	✓
P20_4	10 MHz	40 MHz	✓	✓	—	✓
P20_5	10 MHz	40 MHz	✓	✓	—	✓
P20_6	10 MHz	40 MHz	—	—	—	✓
P20_7	10 MHz	40 MHz	—	—	—	✓
P20_8	10 MHz	40 MHz	—	—	—	✓
P20_9	10 MHz	40 MHz	—	—	—	✓
P20_10	10 MHz	40 MHz	—	—	—	✓
P20_11	10 MHz	40 MHz	—	—	—	✓
P20_12	10 MHz	40 MHz	—	—	—	✓
P20_13	10 MHz	40 MHz	—	—	—	✓
P20_14	10 MHz	40 MHz	—	—	—	✓
P20_15	10 MHz	40 MHz	—	—	—	✓
P21_0	10 MHz	40 MHz	—	—	—	✓
P21_1	10 MHz	40 MHz	—	—	—	✓
P21_2	10 MHz	40 MHz	—	—	—	✓
P21_3	10 MHz	40 MHz	—	—	—	✓
P21_4	10 MHz	40 MHz	—	—	—	✓
P21_5	10 MHz	40 MHz	—	—	—	✓
P21_6	10 MHz	40 MHz	—	—	—	✓
P21_7	10 MHz	40 MHz	—	—	—	✓
P21_8	10 MHz	40 MHz	—	—	—	✓
P21_9	10 MHz	40 MHz	—	—	—	✓
P21_10	10 MHz	40 MHz	—	—	—	✓
P21_11	10 MHz	40 MHz	—	—	—	✓
P21_12	10 MHz	40 MHz	—	—	—	✓
P21_13	10 MHz	40 MHz	—	—	—	✓
P21_14	10 MHz	40 MHz	—	—	—	✓
P22_0	10 MHz	40 MHz	—	—	—	✓
P22_1	10 MHz	40 MHz	—	—	—	✓
P22_2	10 MHz	40 MHz	—	—	—	✓
P22_3	10 MHz	40 MHz	—	—	—	✓
P22_4	10 MHz	40 MHz	—	—	—	✓
P22_5	10 MHz	40 MHz	—	—	—	✓
P22_6	10 MHz	40 MHz	—	—	—	✓
P22_7	10 MHz	40 MHz	—	—	—	✓
P22_8	10 MHz	40 MHz	—	—	—	✓
P22_9	10 MHz	40 MHz	—	—	—	✓
P22_10	10 MHz	40 MHz	—	—	—	✓

Table 2A.86 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	176 Pins	233 Pins	272 Pins	324 Pins
P22_11	10 MHz	40 MHz	—	—	—	✓
P22_12	10 MHz	40 MHz	—	—	—	✓
P22_13	10 MHz	40 MHz	—	—	—	✓
P22_14	10 MHz	40 MHz	—	—	—	✓
P22_15	10 MHz	40 MHz	—	—	—	✓
P23_0	10 MHz	40 MHz	—	—	—	✓
P23_1	10 MHz	40 MHz*2	—	—	—	✓
P23_2	10 MHz	40 MHz	—	—	—	✓
P23_3	10 MHz	40 MHz*2	—	—	—	✓
P23_4	10 MHz	40 MHz	—	—	—	✓
P23_5	10 MHz	40 MHz	—	—	—	✓
P23_6	10 MHz	40 MHz	—	—	—	✓
P23_7	10 MHz	40 MHz	—	—	—	✓
P23_8	10 MHz	40 MHz	—	—	—	✓
P23_9	10 MHz	40 MHz	—	—	—	✓
P23_10	10 MHz	40 MHz	—	—	—	✓
P24_0	10 MHz	40 MHz	—	—	—	✓
P24_1	10 MHz	40 MHz	—	—	—	✓
P24_2	10 MHz	40 MHz	—	—	—	✓
P24_3	10 MHz	40 MHz	—	—	—	✓
P24_4	10 MHz	40 MHz	—	—	—	✓
P24_5	10 MHz	40 MHz	—	—	—	✓
P24_6	10 MHz	40 MHz	—	—	—	✓
P24_7	10 MHz	40 MHz	—	—	—	✓

Note 1. Supports Cload: 100 pF (The load capacitance of CSIH0 is 100 pF.)

Note 2. Supports Cload: 50 pF (The load capacitance of CSIH1 to CSIH4 are 50 pF.)

Note 3. In some of the functions, Fast mode or Slow mode is specified. For details, see **Section 47A.5, AC Characteristics**.

2A.12 Noise Filter & Edge/Level Detector

The input signals at some pins are passed through a filter to remove noise and glitches. The RH850/F1KH supports both analog and digital filters.

It also supports the function for edge and level detection after the signals have passed through a filter.

The first part of this section provides an overview of port input pins that are equipped with a filter and the filter type, noise filter & edge/level detection control registers and control bits, and register addresses.

For details on the digital/analog filter function and noise filter & edge/level detection control registers, see **Section 2A.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2A.12.1 Port Filter Assignment

A list of the input pins that incorporate an analog or digital filter is provided below.

2A.12.1.1 Input Pins that Incorporate Analog Filter Type A

The input pins of analog filter type A incorporate an analog filter and edge/level detection function. Edge/level detection is controlled by the following registers.

- Filter control register FCLA0CTLm_<*name*> (m = 0 to 7)
  A dedicated FCLA0CTLm_<*name*> register is provided for each pin in a port that incorporates an analog filter.

Table 2A.87 Input Pins that Incorporate Analog Filter Type A

Module Name	Input Pin	FCLA0CTL Register Configuration		Device			
		Register	Address	176 Pins	233 Pins	272 Pins	324 Pins
FCLA0	NMI	FCLA0CTL0_NMI	FFC3 4000_H	✓	✓	—	✓
	INTP0	FCLA0CTL0_INTPL	FFC3 4020_H	✓	✓	—	✓
	INTP1	FCLA0CTL1_INTPL	FFC3 4024_H	✓	✓	—	✓
	INTP2	FCLA0CTL2_INTPL	FFC3 4028_H	✓	✓	—	✓
	INTP3	FCLA0CTL3_INTPL	FFC3 $402C_H$	✓	✓	—	✓
	INTP4	FCLA0CTL4_INTPL	FFC3 4030_H	✓	✓	—	✓
	INTP5	FCLA0CTL5_INTPL	FFC3 4034_H	✓	✓	—	✓
	INTP6	FCLA0CTL6_INTPL	FFC3 4038_H	✓	✓	—	✓
	INTP7	FCLA0CTL7_INTPL	FFC3 $403C_H$	✓	✓	—	✓
	INTP8	FCLA0CTL0_INTPH	FFC3 4040_H	✓	✓	—	✓
	INTP9	FCLA0CTL1_INTPH	FFC3 4044_H	✓	✓	—	✓
	INTP10	FCLA0CTL2_INTPH	FFC3 4048_H	✓	✓	—	✓
	INTP11	FCLA0CTL3_INTPH	FFC3 $404C_H$	✓	✓	—	✓
	INTP12	FCLA0CTL4_INTPH	FFC3 4050_H	✓	✓	—	✓
	INTP13	FCLA0CTL5_INTPH	FFC3 4054_H	✓	✓	—	✓
	INTP14	FCLA0CTL6_INTPH	FFC3 4058_H	✓	✓	—	✓
	INTP15	FCLA0CTL7_INTPH	FFC3 $405C_H$	✓	✓	—	✓
	INTP16	FCLA0CTL0_INTPU	FFC3 $40A0_H$	✓	✓	—	✓
	INTP17	FCLA0CTL1_INTPU	FFC3 $40A4_H$	✓	✓	—	✓

Table 2A.87 Input Pins that Incorporate Analog Filter Type A

		FCLA0CTL Register Configuration		Device			
Module Name	Input Pin	Register	Address	176 Pins	233 Pins	272 Pins	324 Pins
FCLA0	INTP18	FCLA0CTL2_INTPU	FFC3 40A8$_H$	✓	✓	—	✓
	INTP19	FCLA0CTL3_INTPU	FFC3 40AC$_H$	✓	✓	—	✓
	INTP20	FCLA0CTL4_INTPU	FFC3 40B0$_H$	✓	✓	—	✓
	INTP21	FCLA0CTL5_INTPU	FFC3 40B4$_H$	✓	✓	—	✓
	INTP22	FCLA0CTL6_INTPU	FFC3 40B8$_H$	✓	✓	—	✓
	INTP23	FCLA0CTL7_INTPU	FFC3 40BC$_H$	✓	✓	—	✓

2A.12.1.2 Input Pins that Incorporate Analog Filter Type B

The input pins of analog filter type B incorporate an analog filter. Edge/level detection is controlled by the registers for individual peripheral functions.

Table 2A.88 Input Pins that Incorporate Analog Filter Type B

		Device			
Input Pin	Edge/Level Detection	176 Pins	233 Pins	272 Pins	324 Pins
TAUJ0I0	Edge detection*1	✓	✓	—	✓
TAUJ0I1	Edge detection*1	✓	✓	—	✓
TAUJ0I2	Edge detection*1	✓	✓	—	✓
TAUJ0I3	Edge detection*1	✓	✓	—	✓
TAUJ1I0	Edge detection*1	✓	✓	—	✓
TAUJ1I1	Edge detection*1	✓	✓	—	✓
TAUJ1I2	Edge detection*1	✓	✓	—	✓
TAUJ1I3	Edge detection*1	✓	✓	—	✓
TAUJ2I0	Edge detection*1	✓	✓	—	✓
TAUJ2I1	Edge detection*1	✓	✓	—	✓
TAUJ2I2	Edge detection*1	✓	✓	—	✓
TAUJ2I3	Edge detection*1	✓	✓	—	✓
TAUJ3I0	Edge detection*1	✓	✓	—	✓
TAUJ3I1	Edge detection*1	✓	✓	—	✓
TAUJ3I2	Edge detection*1	✓	✓	—	✓
TAUJ3I3	Edge detection*1	✓	✓	—	✓
TAPA0ESO	Edge detection*2	✓	✓	—	✓
KR0I0	Low level detection	✓	✓	—	✓
KR0I1	Low level detection	✓	✓	—	✓
KR0I2	Low level detection	✓	✓	—	✓
KR0I3	Low level detection	✓	✓	—	✓
KR0I4	Low level detection	✓	✓	—	✓
KR0I5	Low level detection	✓	✓	—	✓
KR0I6	Low level detection	✓	✓	—	✓
KR0I7	Low level detection	✓	✓	—	✓

Note 1. For details on edge detection for TAUJ, see **Section 33.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register**.

Note 2. For details on edge detection for TAPA, see **Section 36.3.2, TAPAnCTL0 — TAPA Control Register 0**.

2A.12.1.3 Input Pins that Incorporate Analog Filter Type C

The input pins of analog filter type C only incorporate an analog filter function.

Table 2A.89 Input Pins that Incorporate Analog Filter Type C

Input Pin
FLMD0
FLMD1
MODE0
MODE1
MODE2
$\overline{\text{RESET}}$
$\overline{\text{DCUTRST}}$

2A.12.1.4 Input Pins that Incorporate Digital Filter Type D

The input pins of digital filter type D incorporate a digital filter and edge detection function. The digital filter and edge detection are controlled by the following registers.

- Filter control register FCLA0CTLm_<*name*> (m = 0 to 2)
  Each port with a digital filter has a special FCLA0CTLm_<*name*> register.
- Digital noise elimination control register DNFA<*name*>CTL
  Each DNFA<*name*>CTL control register controls digital filter processing for three input signals per group.
- Digital noise elimination enable register DNFA<*name*>EN
  The setting of the DNFA<*name*>ENL[2:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for three input signals per group.

Table 2A.90 Input Pins that Incorporate Digital Filter Type D

	Device				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Filter Control Register	
Input Pin	176 Pins	233 Pins	272 Pins	324 Pins	Control Register	Address	Control Register	Control Bit	Address	Control Register	Address
ADCA0TRG0	✓	✓	—	✓	DNFAADCTL0CTL	FFC3 00A0$_H$	DNFAADCTL0EN (DNFAADCTL0ENL)	DNFAADCTL0ENL0	FFC3 00A4$_H$ (FFC3 00AC$_H$)	FCLA0CTL0_ADC0	FFC3 4060$_H$
ADCA0TRG1	✓	✓	—	✓				DNFAADCTL0ENL1		FCLA0CTL1_ADC0	FFC3 4064$_H$
ADCA0TRG2	✓	✓	—	✓				DNFAADCTL0ENL2		FCLA0CTL2_ADC0	FFC3 4068$_H$
ADCA1TRG0	✓	✓	—	✓	DNFAADCTL1CTL	FFC3 00C0$_H$	DNFAADCTL1EN (DNFAADCTL1ENL)	DNFAADCTL1ENL0	FFC3 00C4$_H$ (FFC3 00CC$_H$)	FCLA0CTL0_ADC1	FFC3 4080$_H$
ADCA1TRG1	✓	✓	—	✓				DNFAADCTL1ENL1		FCLA0CTL1_ADC1	FFC3 4084$_H$
ADCA1TRG2	✓	✓	—	✓				DNFAADCTL1ENL2		FCLA0CTL2_ADC1	FFC3 4088$_H$

2A.12.1.5 Input Pins that Incorporate Digital Filter Type E

The input pins of digital filter type E incorporate a digital filter. The digital filter is controlled by the following registers. Edge detection is controlled by the registers for individual peripheral functions.

- Digital noise elimination control register DNFA<*name*>CTL
  Each DNFA<*name*>CTL control register controls digital filter processing for up to 16 input signals per group.
- Digital noise elimination enable register DNFA<*name*>EN
  The setting of the DNFA<*name*>ENL[7:0] and DNFA<*name*>ENH[7:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for up to 16 input signals per group.

Table 2A.91 Input Pins that Incorporate Digital Filter Type E

	Devises				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
Input Pin	176 Pins	233 Pins	272 Pins	324 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUD0I0	✓	✓	—	✓	DNFATAUD0ICTL	FFC3 0000$_H$	DNFATAUD0IEN (DNFATAUD0IENH/ DNFATAUD0IENL)	DNFATAUD0IENL0	FFC3 0004$_H$ (FFC3 0008$_H$/ FFC3 000C$_H$)	*1
TAUD0I1	✓	✓	—	✓				DNFATAUD0IENL1		
TAUD0I2	✓	✓	—	✓				DNFATAUD0IENL2		
TAUD0I3	✓	✓	—	✓				DNFATAUD0IENL3		
TAUD0I4	✓	✓	—	✓				DNFATAUD0IENL4		
TAUD0I5	✓	✓	—	✓				DNFATAUD0IENL5		
TAUD0I6	✓	✓	—	✓				DNFATAUD0IENL6		
TAUD0I7	✓	✓	—	✓				DNFATAUD0IENL7		
TAUD0I8	✓	✓	—	✓				DNFATAUD0IENH0		
TAUD0I9	✓	✓	—	✓				DNFATAUD0IENH1		
TAUD0I10	✓	✓	—	✓				DNFATAUD0IENH2		
TAUD0I11	✓	✓	—	✓				DNFATAUD0IENH3		
TAUD0I12	✓	✓	—	✓				DNFATAUD0IENH4		
TAUD0I13	✓	✓	—	✓				DNFATAUD0IENH5		
TAUD0I14	✓	✓	—	✓				DNFATAUD0IENH6		
TAUD0I15	✓	✓	—	✓				DNFATAUD0IENH7		
TAUB0I0	✓	✓	—	✓	DNFATAUB0ICTL	FFC3 0020$_H$	DNFATAUB0IEN (DNFATAUB0IENH/ DNFATAUB0IENL)	DNFATAUB0IENL0	FFC3 0024$_H$ (FFC3 0028$_H$/ FFC3 002C$_H$)	*2
TAUB0I1	✓	✓	—	✓				DNFATAUB0IENL1		
TAUB0I2	✓	✓	—	✓				DNFATAUB0IENL2		
TAUB0I3	✓	✓	—	✓				DNFATAUB0IENL3		
TAUB0I4	✓	✓	—	✓				DNFATAUB0IENL4		
TAUB0I5	✓	✓	—	✓				DNFATAUB0IENL5		
TAUB0I6	✓	✓	—	✓				DNFATAUB0IENL6		
TAUB0I7	✓	✓	—	✓				DNFATAUB0IENL7		
TAUB0I8	✓	✓	—	✓				DNFATAUB0IENH0		
TAUB0I9	✓	✓	—	✓				DNFATAUB0IENH1		
TAUB0I10	✓	✓	—	✓				DNFATAUB0IENH2		
TAUB0I11	✓	✓	—	✓				DNFATAUB0IENH3		
TAUB0I12	✓	✓	—	✓				DNFATAUB0IENH4		
TAUB0I13	✓	✓	—	✓				DNFATAUB0IENH5		
TAUB0I14	✓	✓	—	✓				DNFATAUB0IENH6		
TAUB0I15	✓	✓	—	✓				DNFATAUB0IENH7		

Table 2A.91 Input Pins that Incorporate Digital Filter Type E

	Devises				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
Input Pin	176 Pins	233 Pins	272 Pins	324 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUB1I0	✓	✓	—	✓	DNFATAUB1ICTL	FFC3 0040$_{H}$	DNFATAUB1IEN (DNFATAUB1IENH/ DNFATAUB1IENL)	DNFATAUB1IENL0	FFC3 0044$_{H}$ (FFC3 0048$_{H}$/ FFC3 004C$_{H}$)	*2
TAUB1I1	✓	✓	—	✓				DNFATAUB1IENL1		
TAUB1I2	✓	✓	—	✓				DNFATAUB1IENL2		
TAUB1I3	✓	✓	—	✓				DNFATAUB1IENL3		
TAUB1I4	✓	✓	—	✓				DNFATAUB1IENL4		
TAUB1I5	✓	✓	—	✓				DNFATAUB1IENL5		
TAUB1I6	✓	✓	—	✓				DNFATAUB1IENL6		
TAUB1I7	✓	✓	—	✓				DNFATAUB1IENL7		
TAUB1I8	✓	✓	—	✓				DNFATAUB1IENH0		
TAUB1I9	✓	✓	—	✓				DNFATAUB1IENH1		
TAUB1I10	✓	✓	—	✓				DNFATAUB1IENH2		
TAUB1I11	✓	✓	—	✓				DNFATAUB1IENH3		
TAUB1I12	✓	✓	—	✓				DNFATAUB1IENH4		
TAUB1I13	✓	✓	—	✓				DNFATAUB1IENH5		
TAUB1I14	✓	✓	—	✓				DNFATAUB1IENH6		
TAUB1I15	✓	✓	—	✓				DNFATAUB1IENH7		
ENCA0TIN0	✓	✓	—	✓	DNFAENCA0ICTL	FFC3 0060$_{H}$	DNFAENCA0IEN (DNFAENCA0IENL)	DNFAENCA0IENL0	FFC3 0064$_{H}$ (FFC3 006C$_{H}$)	*3
ENCA0TIN1	✓	✓	—	✓				DNFAENCA0IENL1		
ENCA0E0	✓	✓	—	✓				DNFAENCA0IENL2		
ENCA0E1	✓	✓	—	✓				DNFAENCA0IENL3		
ENCA0EC	✓	✓	—	✓				DNFAENCA0IENL4		
SENT0RX	✓	✓	—	✓	DNFASENTICTL	FFC3 00E0$_{H}$	DNFASENTIEN (DNFASENTIENL)	DNFASENTIENL0	FFC3 00E4$_{H}$ (FFC3 00EC$_{H}$)	—*4
SENT1RX	✓	✓	—	✓				DNFASENTIENL1		

Note 1. For the setting for TAUD edge detection, see **Section 32.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register**.

Note 2. For the setting for TAUB edge detection, see **Section 31.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register**.

Note 3. For the setting for ENCA edge detection, see **Section 35.3.3, ENCAnIOC0 — ENCAn I/O Control Register 0**.

Note 4. RSENT does not have the edge detection.

2A.12.2 Clock Supply for Port Filters

The following table shows the clock supply for each filter type in each port domain.

Table 2A.92 Clock Supply for Port Filters

Peripheral Function	Port Domain*[1]	Filter Type	Filter Clock	Setting Register	
				Source Clock Selection	Clock Selection
ADCA0	Always-On area (AWO area)	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
ADCA1	Isolated area (ISO area)	Digital filter type D	DNFATCKI	CKSC_IADCAS_CTL	CKSC_IADCAD_CTL
TAUD0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
TAUB0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
TAUB1	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
ENCA0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
SENTn	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—

Note 1. Power Domain

NOTE

For the Setting Register, see **Section 12AB.4.3, Clock Selector Control Register.**

2A.13 Description of Port Noise Filter & Edge/Level Detection

External signals pass through different types of filters according to the use of each external input signal.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2A.13.1 Overview

2A.13.1.1 Analog Filter Types

Analog filters have fixed characteristics.

- Type A: An analog filter with edge detection or level detection.
  Used for external interrupt signals.
- Type B: An analog filter
  Edge detection is performed by each peripheral function. Used for the timer input signals, asynchronous Hi-Z control input signals, and key return input signals.
- Type C: An analog filter only
  Used for the external RESET input and mode signals.

2A.13.1.2 Digital Filter Types

The digital filter characteristics can be adjusted to suit the application.

- Type D: A digital filter with edge detection.
  Used for the A/D converter external trigger pin.
- Type E: A digital filter. Edge detection is performed by each peripheral function.
  Used for the timer input signals and encoder input signals.

2A.13.2 Analog Filters

2A.13.2.1 Analog Filter Characteristic

See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for the input conditions for signals input to pins that incorporate an analog filter.

2A.13.2.2 Analog Filter Control Registers

A dedicated FCLA0CTLm_<*name*> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Table 2A.87, Input Pins that Incorporate Analog Filter Type A**.

2A.13.2.3 Analog Filter in Standby Mode

Analog filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Analog filters in the Always-On area (AWO area) always operate.

The analog filter in standby mode and its wake-up capability depend on the filter types. See the description of the analog filter types below.

(1) Analog Filter Type A

A block diagram of analog filter type A is shown below.

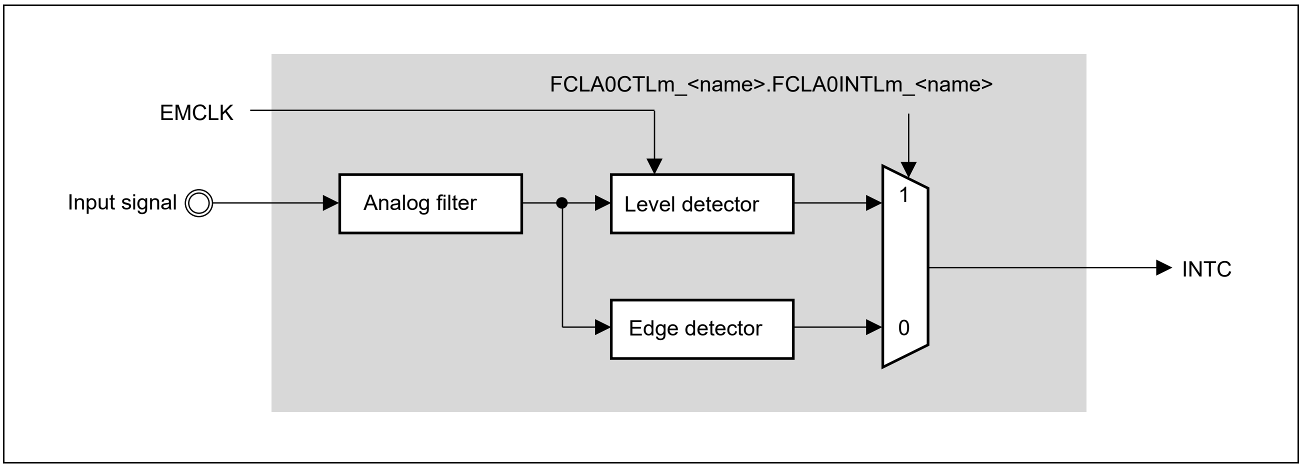


Figure 2A.12 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and glitches, an output signal is generated according to whether an event is detected; that is whether a specified level is detected or whether a change in the level (an edge) occurs.

Whether a level or an edge is detected is selected by the control bit FCLA0CTLm_<*name*>.FCLA0INTLm_<*name*>.

- FCLA0INTLm_<*name*> bit = 0: Edge detection

  Whether a rising or falling edge is detected can be specified by setting the FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> and FCLA0CTLm_<*name*>.FCLA0INTFm_<*name*> bits.

- FCLA0INTLm_<*name*> bit = 1: Level detection

  The detection of a high level or low level can be specified by setting FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> bit.

The table below summarizes the detection conditions of the analog filter.

Table 2A.93 Analog Filter Event Detection Conditions

FCLA0INTLm_<name>	FCLA0INTFm_<*name*>	FCLA0INTRm_<*name*>	Edge Detection	Level Detection
0	0	0	No edge detected	Disabled
	0	1	Rising edge	
	1	0	Falling edge	
	1	1	Rising and falling edges	
1	X	0	Disabled	Low level
	X	1		High level

Analog filter type A in Standby mode

The output signal of an analog filter type A can always be used as a standby mode wake-up signal.

(2) Analog filter type B

A block diagram of analog filter type B is shown below.

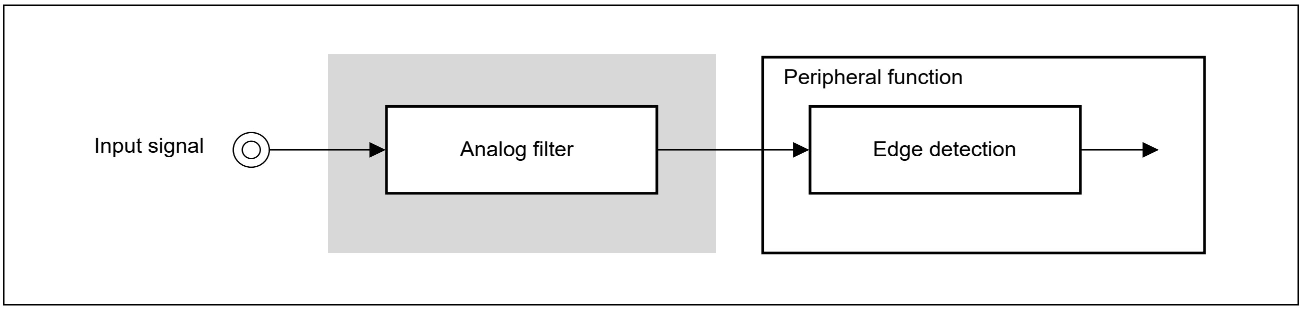


Figure 2A.13 Block Diagram of Analog Filter Type B

Analog filter type B in Standby mode

The output signal of an analog filter type B can always be used as a standby mode wake-up signal.

(3) Analog filter type C

A block diagram of analog filter type C is shown below.

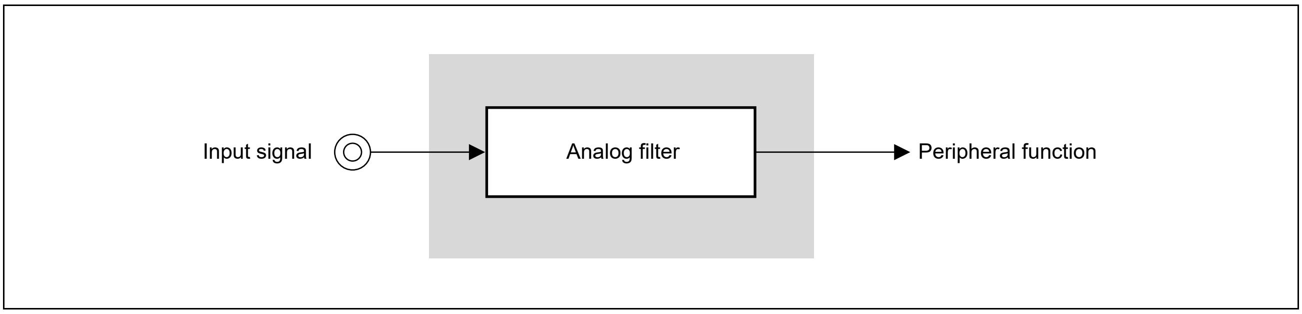


Figure 2A.14 Block Diagram of Analog Filter Type C

The generated signals are always input signals that have passed through an analog filter.

Analog filter type C in Standby mode

Pins equipped with type C analog filters in this product do not support the input of event signals to trigger wake-up from standby.

2A.13.3 Digital Filters

2A.13.3.1 Digital Filter Characteristic

The digital filters allow the filter characteristics to be adjusted accordingly to the needs of the application.

The input signal is sampled with the sampling frequency fs.

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] select the sampling frequency based on
  $f_s = f_{DNFATCKI} / 2^{DNFA<name>PRS[2:0]}$
  where fDNFATCKI is the frequency of the DNFATCKI clock.
- DNFA<*name*>CTL.DNFA<*name*>NFSTS[1:0] determines the number of same level samples, “s”, (2 to 5):

  s = DNFA<*name*>NFSTS[1:0] + 2

External signal pulses shorter than the following are suppressed at all times.

$s \times 1/f_s$

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

$(s + 1) \times 1/f_s$

External signal pulses in the following range may be suppressed or judged as valid.

$s \times 1/f_s$ to $(s + 1) \times 1/f_s$

The filter operation is illustrated in the figure below with DNFA<*name*>NFSTS[1:0] = 01_B, i.e. s = 3 same level samples.

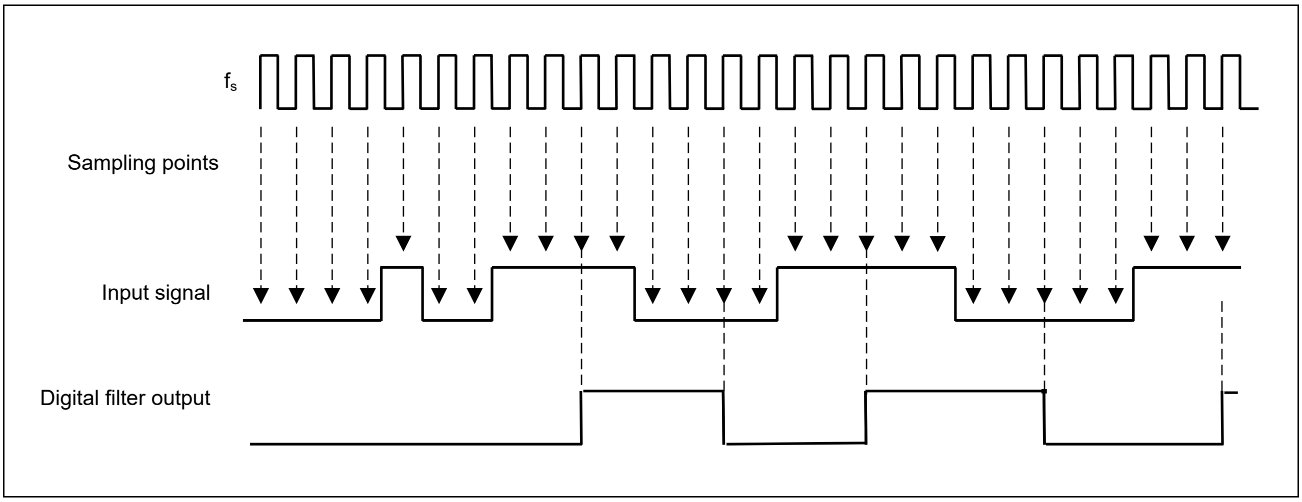


Figure 2A.15 Digital Filter Function

2A.13.3.2 Digital Filter Groups

The input signals processed through digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] and DNFA<*name*>NFSTS[1:0] apply to the signals.

However, the digital filter for each signal can be enabled or disabled separately by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

CAUTIONS

1. When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) for the port pin to switch to the alternative function.
   s = DNFA<*name*>NFSTS[1:0] + 2
   $s \times 1/f_s + 2 \times 1/f_{DNFATCKI}$
2. When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.
   $s \times 1/f_s + 3 \times 1/f_{DNFATCKI}$

2A.13.3.3 Digital Filters in Standby Mode

Digital filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Digital filters on the Always-On area (AWO area) are always operating.

Digital noise elimination requires the clock supply DNFATCKI to operate.
Pins equipped with digital filters in this product do not support the input of event signals to trigger wake-up from standby.

2A.13.3.4 Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFA<*name*>CTL and digital noise elimination enable register DNFA<*name*>EN are used to set all the filters in the same group (<*name*> = peripheral function group).

The DNFA<*name*>CTL register specifies the characteristics of the digital noise elimination filter for the digital filter of <*name*>.

The DNFA<*name*>EN register enables/disables each filter by setting the corresponding bit in DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The FCLA0CTLm_ADCn registers are ordered in groups of 3 registers with the same index n. The register index n is in 0 or 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2A.90, Input Pins that Incorporate Digital Filter Type D** and **Table 2A.91, Input Pins that Incorporate Digital Filter Type E** in **Section 2A.12.1, Port Filter Assignment**.

CAUTION

Do not change any control register settings while the corresponding digital filter is enabled by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1. Otherwise an unintended filter output may be generated.

(1) Digital filter type D

A block diagram of digital filter type D is shown below.

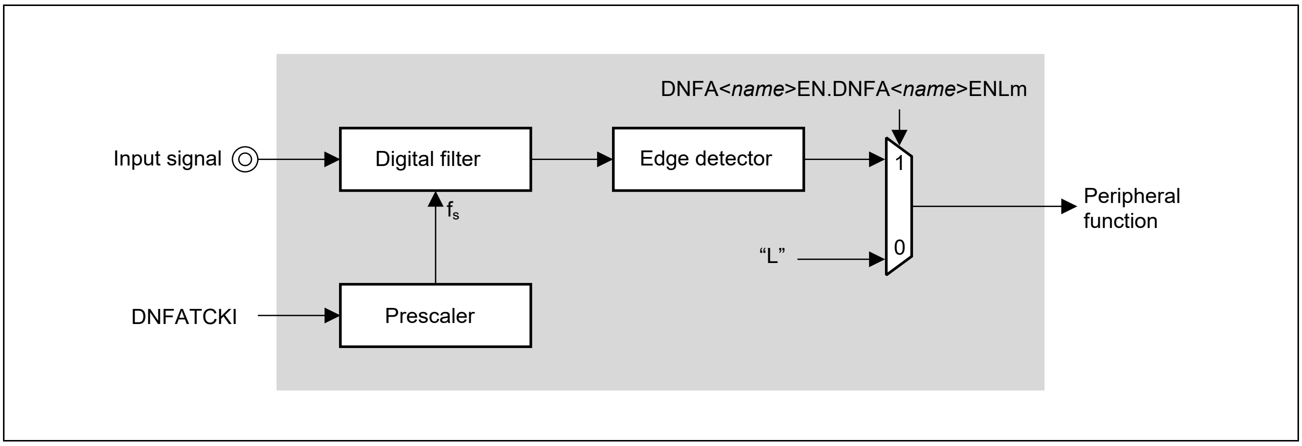


Figure 2A.16 Block Diagram of Digital Filter Type D

The generated signal depends on the register setting, as shown in the following table.

Table 2A.94 Output Options for Digital Filter Type D

DNFA<*name*>EN.DNFA<*name*>ENLm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

(2) Digital filter type E

A block diagram of digital filter type E is shown below.

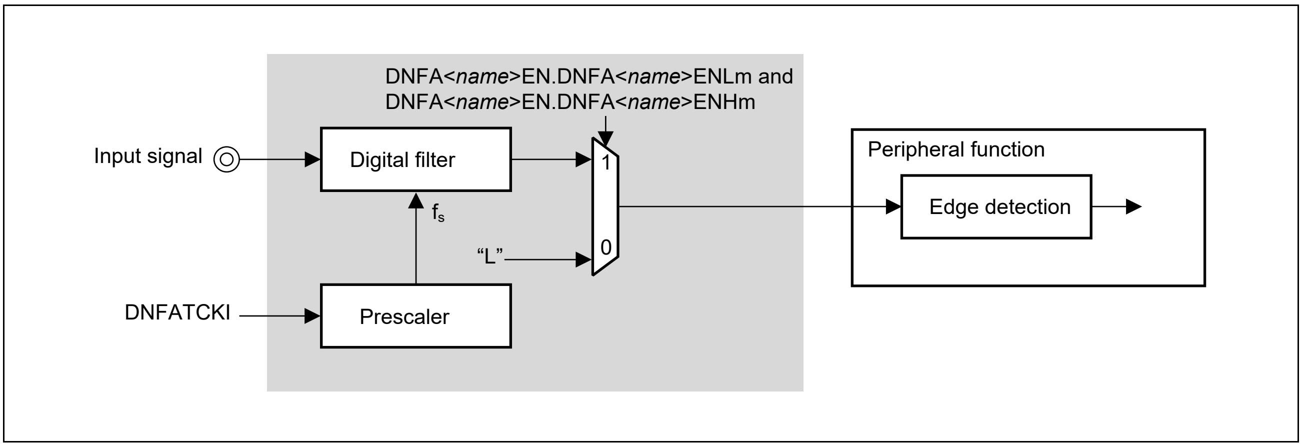


Figure 2A.17 Block Diagram of Digital Filter Type E

The generated signal depends on the register setting, as shown in the following table.

Table 2A.95 Output Options for Digital Filter Type E

DNFA<*name*>EN.DNFA<*name*>ENLm and DNFA<*name*>EN.DNFA<*name*>ENHm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

2A.13.4 Filter Control Registers

The analog and digital filters are controlled and operated by the following registers:

Table 2A.96 List of Filter Registers

Module Name	Register Name	Symbol	Address
FCLA0	Filter control register m	FCLA0CTLm_<name>	The addresses are shown in the tables in **Section 2A.12.1, Port Filter Assignment**.
DNF	Digital noise elimination control register	DNFA<name>CTL	
	Digital noise elimination enable register	DNFA<name>EN	
	Digital noise elimination enable H register	DNFA<name>ENH	
	Digital noise elimination enable L register	DNFA<name>ENL	

2A.13.4.1 FCLA0CTLm_<name> — Filter Control Register

This register controls the analog and digital filter operation.

Access: This register can be read or written in 8-bit units.

Address: The allocation of input signals to FCLA0CTLm_<name> registers and the address of each register are shown in the tables in **Section 2A.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLA0INTLm _<name>	FCLA0INTFm _<name>	FCLA0INTRm _<name>
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 2A.97 FCLA0CTLm_<name> Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	FCLA0INTLm _<name>	Detection Mode Selection 0: Edge detection 1: Level detection **NOTE:** This bit is only valid for analog filter type A.
1	FCLA0INTFm _<name>	● In level detection mode (FCLA0INTLm_<name> = 1): This bit has no effect. ● In edge detection mode (FCLA0INTLm_<name> = 0): Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.
0	FCLA0INTRm _<name>	● In level detection mode (FCLA0INTLm_<name> = 1): Detected level selection 0: Low level detection 1: High level detection ● In edge detection mode (FCLA0INTLm_<name> = 0): Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.

CAUTION

Digital filter type D: Always set bit 2 to "0".

2A.13.4.2 DNFA<name>CTL — Digital Noise Elimination Control Register

This register is used to specify the filter characteristics of the digital noise elimination filter.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit units.

Address: For the correspondence between the DNFA<name>CTL register and input signals, and the addresses of individual registers, see **Table 2A.90, Input Pins that Incorporate Digital Filter Type D** and **Table 2A.91, Input Pins that Incorporate Digital Filter Type E** in **Section 2A.12.1, Port Filter Assignment.**

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	DNFA<name>NFSTS[1:0]		—	—	DNFA<name>PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2A.98 DNFA<name>CTL Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>6, 5</td><td>DNFA<name>
NFSTS[1:0]</td><td>The DNFA<name>NFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid.
<table>
<tr><th>DNFA<name>NFSTS[1:0]</th><th>Number of Samples</th></tr>
<tr><td>00_B</td><td>2</td></tr>
<tr><td>01_B</td><td>3</td></tr>
<tr><td>10_B</td><td>4</td></tr>
<tr><td>11_B</td><td>5</td></tr>
</table></td></tr>
<tr><td>4, 3</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>2 to 0</td><td>DNFA<name>
PRS[2:0]</td><td>Digital filter sampling clock selection
<table>
<tr><th>DNFA<name>PRS[2:0]</th><th>Sampling Clock Frequency</th></tr>
<tr><td>000_B</td><td>DNFATCKI/1</td></tr>
<tr><td>001_B</td><td>DNFATCKI/2</td></tr>
<tr><td>010_B</td><td>DNFATCKI/4</td></tr>
<tr><td>011_B</td><td>DNFATCKI/8</td></tr>
<tr><td>100_B</td><td>DNFATCKI/16</td></tr>
<tr><td>101_B</td><td>DNFATCKI/32</td></tr>
<tr><td>110_B</td><td>DNFATCKI/64</td></tr>
<tr><td>111_B</td><td>DNFATCKI/128</td></tr>
</table></td></tr>
</table>

2A.13.4.3 DNFA<name>EN — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 16-bit units.

The upper- and lower-order bytes (DNFA<name>ENH[7:0] and DNFA<name>ENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH. and DNFA<name>ENL.

Address: For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Table 2A.90, Input Pins that Incorporate Digital Filter Type D** and **Table 2A.91, Input Pins that Incorporate Digital Filter Type E** in **Section 2A.12.1, Port Filter Assignment.**

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFA<name>ENH7	DNFA<name>ENH6	DNFA<name>ENH5	DNFA<name>ENH4	DNFA<name>ENH3	DNFA<name>ENH2	DNFA<name>ENH1	DNFA<name>ENH0	DNFA<name>ENL7	DNFA<name>ENL6	DNFA<name>ENL5	DNFA<name>ENL4	DNFA<name>ENL3	DNFA<name>ENL2	DNFA<name>ENL1	DNFA<name>ENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2A.99 DNFA<name>EN Register Contents

Bit Position	Bit Name	Function
15 to 0	DNFA<name> ENH[7:0] DNFA<name> ENL[7:0]	Digital Noise Elimination Enable/Disable Control 0: Fixed to low level 1: Input signal passed through filter

2A.13.4.4 DNFA<name>ENH – Digital Noise Elimination Enable H Register

Setting in this register correspond to those of the 8 upper-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENH register and input signals, and the addresses of individual registers, see **Table 2A.91, Input Pins that Incorporate Digital Filter Type E** in **Section 2A.12.1, Port Filter Assignment.**

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA<name> ENH7	DNFA<name> ENH6	DNFA<name> ENH5	DNFA<name> ENH4	DNFA<name> ENH3	DNFA<name> ENH2	DNFA<name> ENH1	DNFA<name> ENH0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2A.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

2A.13.4.5 DNFA<name>ENL – Digital Noise Elimination Enable L Register

Setting in this register correspond to those of the 8 lower-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENL register and input signals, and the addresses of individual registers, see **Table 2A.90, Input Pins that Incorporate Digital Filter Type D** and **Table 2A.91, Input Pins that Incorporate Digital Filter Type E** in **Section 2A.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA <name> ENL7	DNFA <name> ENL6	DNFA <name> ENL5	DNFA <name> ENL4	DNFA <name> ENL3	DNFA <name> ENL2	DNFA <name> ENL1	DNFA <name> ENL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2A.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

Section 2B Pin Function of RH850/F1KM-S4

This section describes the pin and port functions.

Section 2B.1, Pin Connection Diagram to **Section 2B.5, Recommended Connection of Unused Pins** describe the pin connections and respective pins.

Section 2B.6, Features of RH850/F1KM Port to **Section 2B.13, Description of Port Noise Filter & Edge/Level Detection** describe the general port functions.

2B.1 Pin Connection Diagram

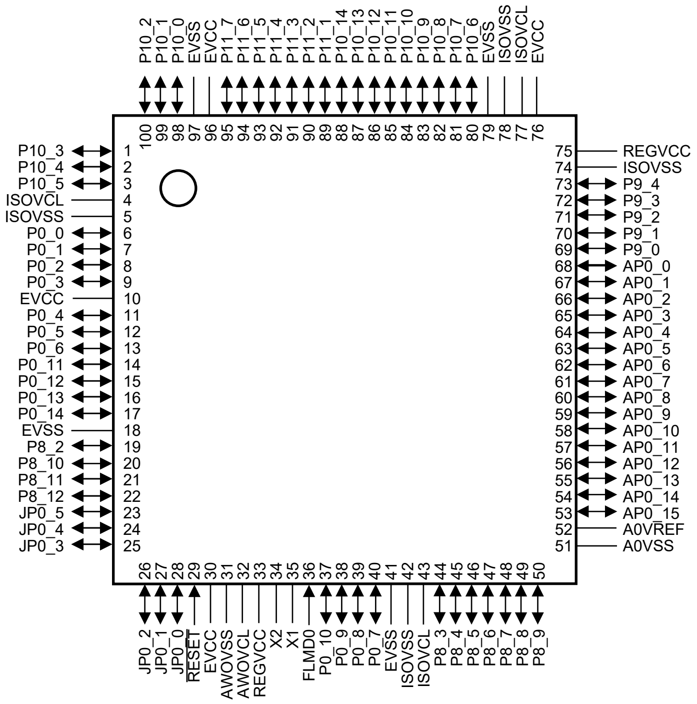


Figure 2B.1 Pin Connection Diagram (100-Pin LQFP)

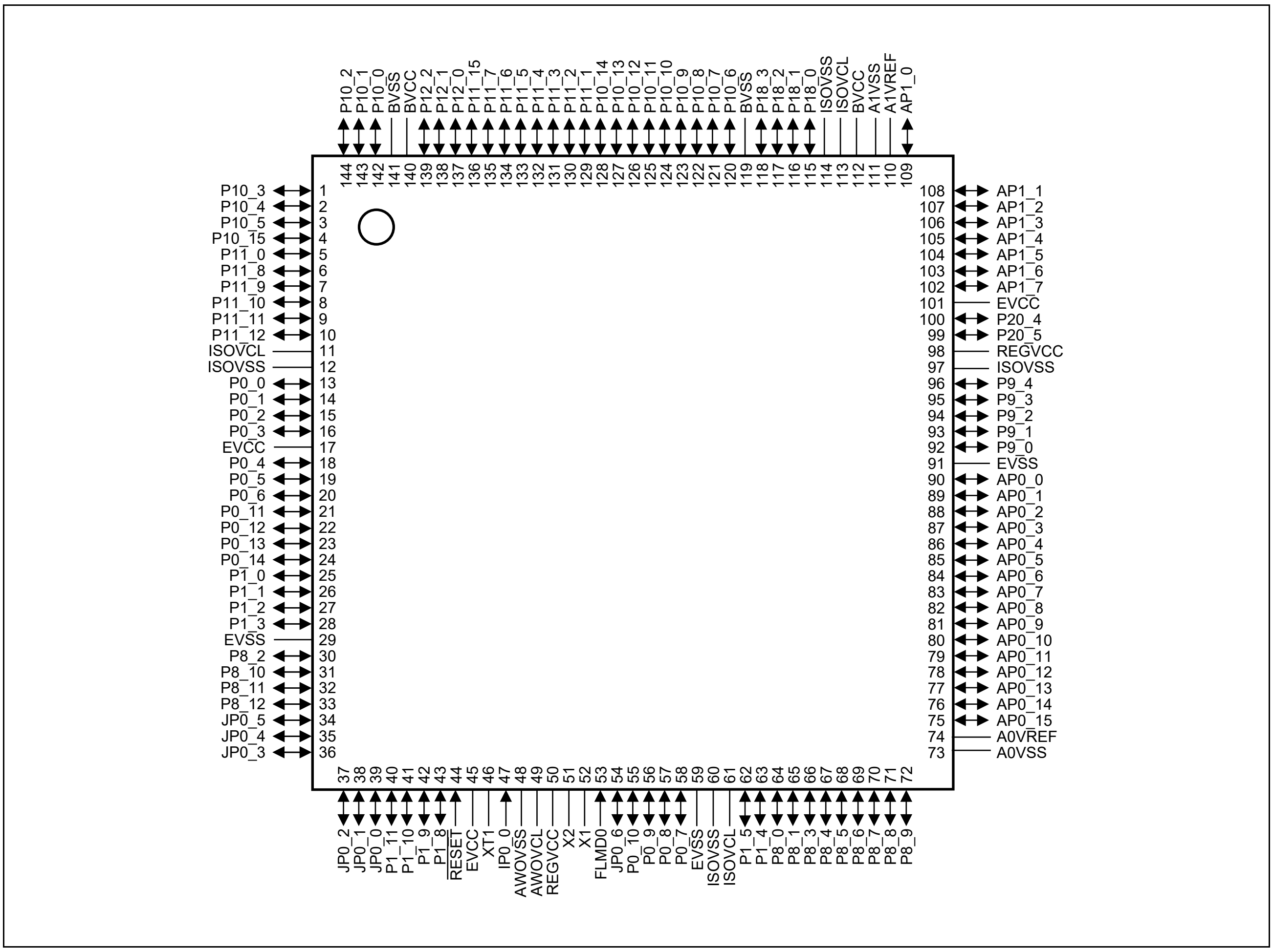


Figure 2B.2 Pin Connection Diagram (144-Pin LQFP)

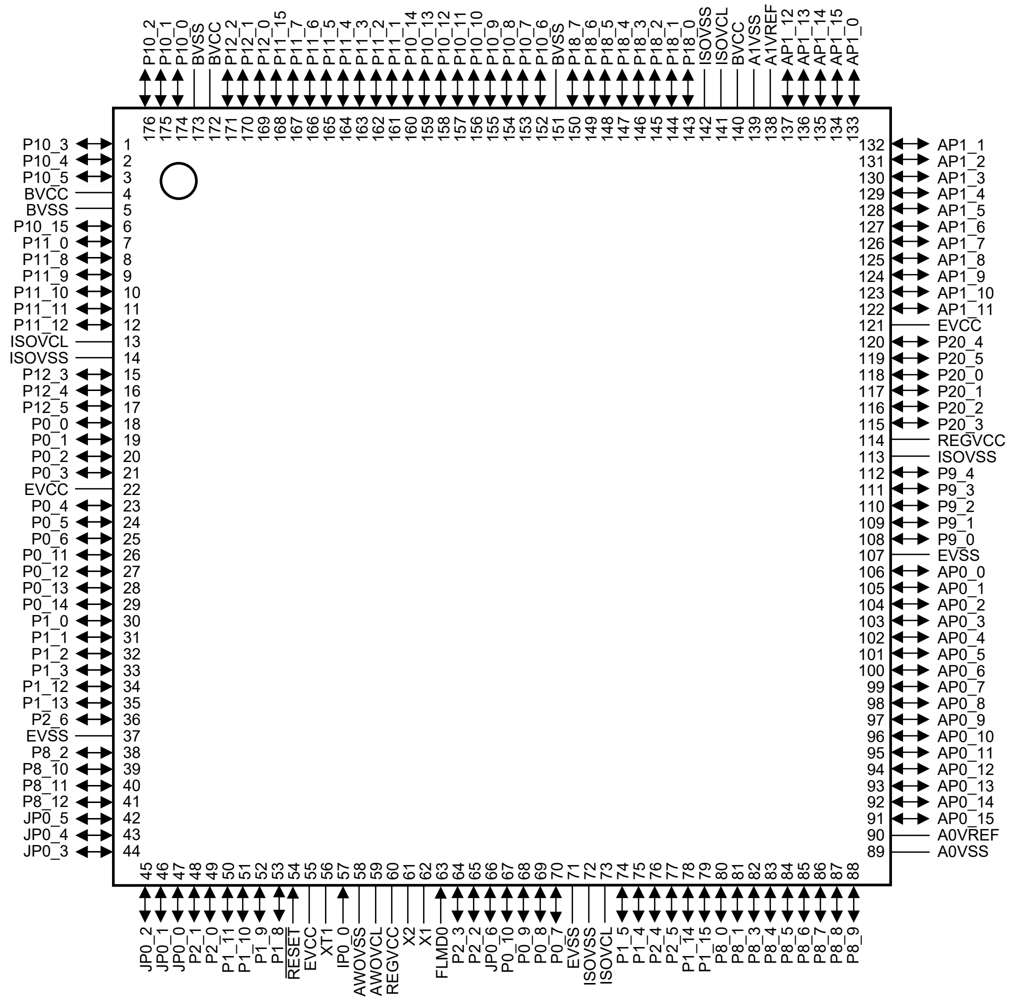
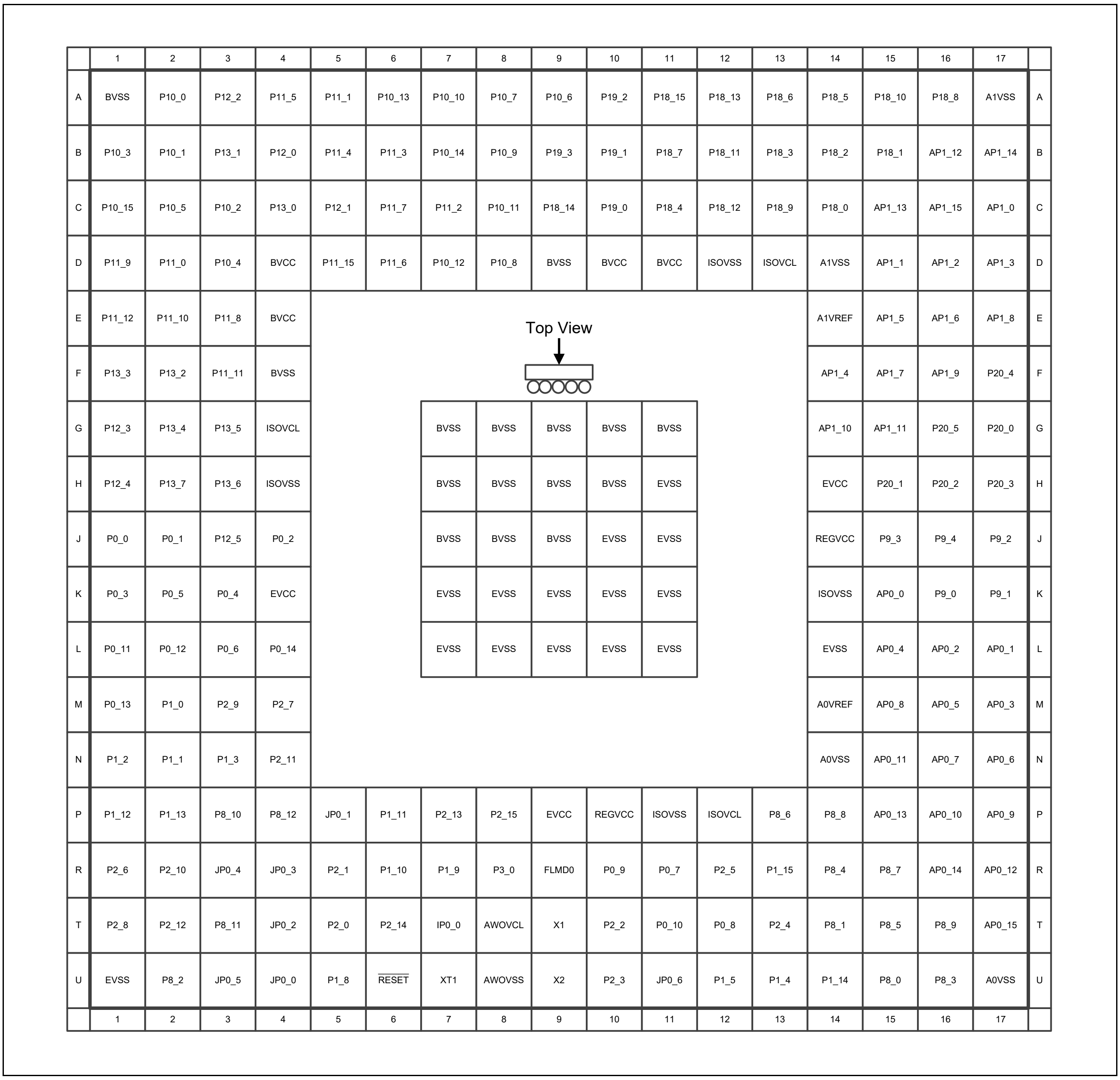
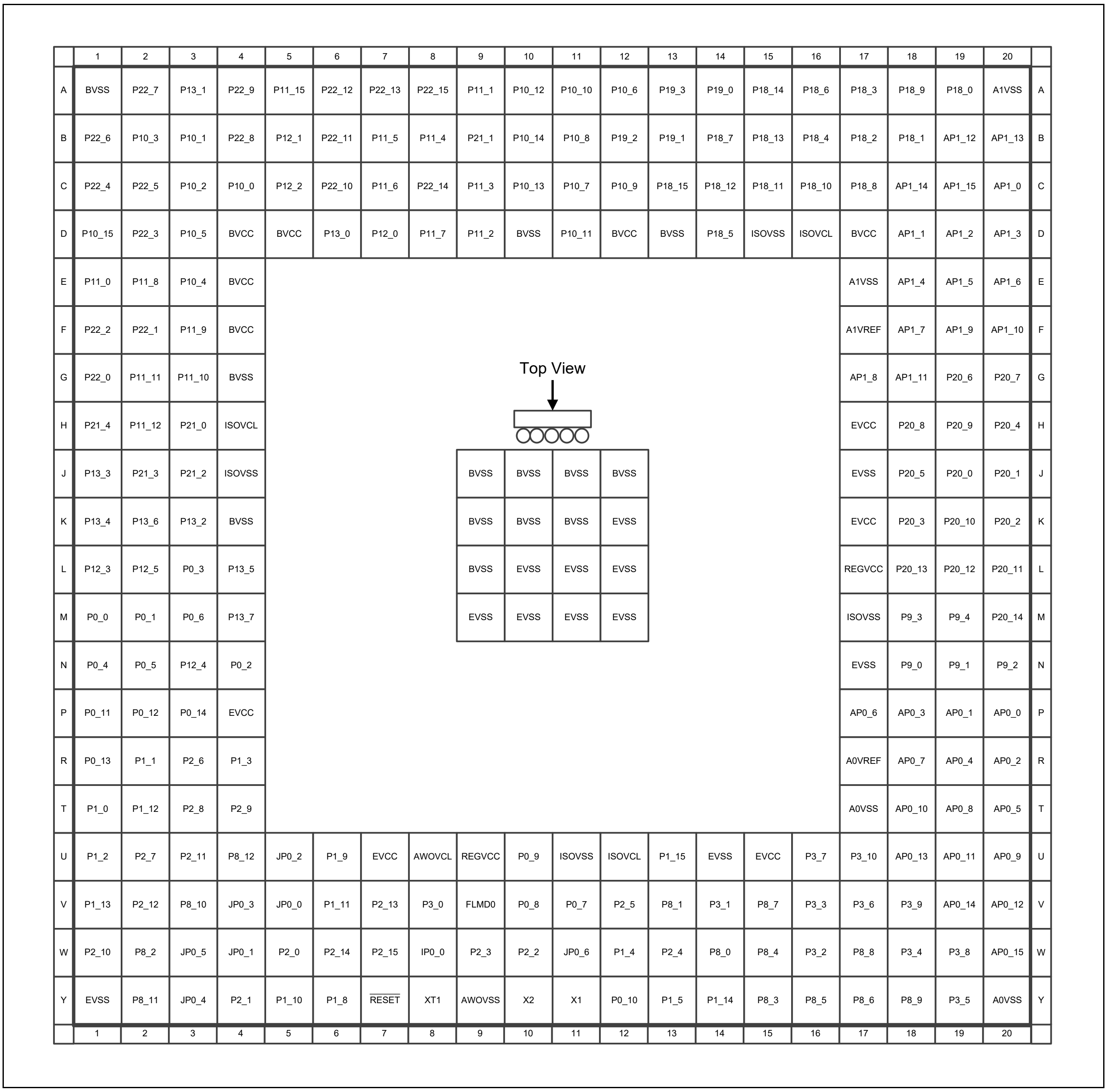


Figure 2B.3 Pin Connection Diagram (176-Pin LQFP)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	BVSS	P10_0	P12_2	P11_5	P11_1	P10_13	P10_10	P10_7	P10_6	P19_2	P18_15	P18_13	P18_6	P18_5	P18_10	P18_8	A1VSS	A
B	P10_3	P10_1	P13_1	P12_0	P11_4	P11_3	P10_14	P10_9	P19_3	P19_1	P18_7	P18_11	P18_3	P18_2	P18_1	AP1_12	AP1_14	B
C	P10_15	P10_5	P10_2	P13_0	P12_1	P11_7	P11_2	P10_11	P18_14	P19_0	P18_4	P18_12	P18_9	P18_0	AP1_13	AP1_15	AP1_0	C
D	P11_9	P11_0	P10_4	BVCC	P11_15	P11_6	P10_12	P10_8	BVSS	BVCC	BVCC	ISOVSS	ISOVCL	A1VSS	AP1_1	AP1_2	AP1_3	D
E	P11_12	P11_10	P11_8	BVCC										A1VREF	AP1_5	AP1_6	AP1_8	E
F	P13_3	P13_2	P11_11	BVSS										AP1_4	AP1_7	AP1_9	P20_4	F
G	P12_3	P13_4	P13_5	ISOVCL			BVSS	BVSS	BVSS	BVSS	BVSS			AP1_10	AP1_11	P20_5	P20_0	G
H	P12_4	P13_7	P13_6	ISOVSS			BVSS	BVSS	BVSS	BVSS	EVSS			EVCC	P20_1	P20_2	P20_3	H
J	P0_0	P0_1	P12_5	P0_2			BVSS	BVSS	BVSS	EVSS	EVSS			REGVCC	P9_3	P9_4	P9_2	J
K	P0_3	P0_5	P0_4	EVCC			EVSS	EVSS	EVSS	EVSS	EVSS			ISOVSS	AP0_0	P9_0	P9_1	K
L	P0_11	P0_12	P0_6	P0_14			EVSS	EVSS	EVSS	EVSS	EVSS			EVSS	AP0_4	AP0_2	AP0_1	L
M	P0_13	P1_0	P2_9	P2_7										A0VREF	AP0_8	AP0_5	AP0_3	M
N	P1_2	P1_1	P1_3	P2_11										A0VSS	AP0_11	AP0_7	AP0_6	N
P	P1_12	P1_13	P8_10	P8_12	JP0_1	P1_11	P2_13	P2_15	EVCC	REGVCC	ISOVSS	ISOVCL	P8_6	P8_8	AP0_13	AP0_10	AP0_9	P
R	P2_6	P2_10	JP0_4	JP0_3	P2_1	P1_10	P1_9	P3_0	FLMD0	P0_9	P0_7	P2_5	P1_15	P8_4	P8_7	AP0_14	AP0_12	R
T	P2_8	P2_12	P8_11	JP0_2	P2_0	P2_14	IP0_0	AWOVCL	X1	P2_2	P0_10	P0_8	P2_4	P8_1	P8_5	P8_9	AP0_15	T
U	EVSS	P8_2	JP0_5	JP0_0	P1_8	$\overline{\text{RESET}}$	XT1	AWOVSS	X2	P2_3	JP0_6	P1_5	P1_4	P1_14	P8_0	P8_3	A0VSS	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Figure 2B.4 Pin Connection Diagram (233-Pin FPBGA)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	BVSS	P22_7	P13_1	P22_9	P11_15	P22_12	P22_13	P22_15	P11_1	P10_12	P10_10	P10_6	P19_3	P19_0	P18_14	P18_6	P18_3	P18_9	P18_0	A1VSS	A
B	P22_6	P10_3	P10_1	P22_8	P12_1	P22_11	P11_5	P11_4	P21_1	P10_14	P10_8	P19_2	P19_1	P18_7	P18_13	P18_4	P18_2	P18_1	AP1_12	AP1_13	B
C	P22_4	P22_5	P10_2	P10_0	P12_2	P22_10	P11_6	P22_14	P11_3	P10_13	P10_7	P10_9	P18_15	P18_12	P18_11	P18_10	P18_8	AP1_14	AP1_15	AP1_0	C
D	P10_15	P22_3	P10_5	BVCC	BVCC	P13_0	P12_0	P11_7	P11_2	BVSS	P10_11	BVCC	BVSS	P18_5	ISOVSS	ISOVCL	BVCC	AP1_1	AP1_2	AP1_3	D
E	P11_0	P11_8	P10_4	BVCC													A1VSS	AP1_4	AP1_5	AP1_6	E
F	P22_2	P22_1	P11_9	BVCC													A1VREF	AP1_7	AP1_9	AP1_10	F
G	P22_0	P11_11	P11_10	BVSS													AP1_8	AP1_11	P20_6	P20_7	G
H	P21_4	P11_12	P21_0	ISOVCL													EVCC	P20_8	P20_9	P20_4	H
J	P13_3	P21_3	P21_2	ISOVSS					BVSS	BVSS	BVSS	BVSS					EVSS	P20_5	P20_0	P20_1	J
K	P13_4	P13_6	P13_2	BVSS					BVSS	BVSS	BVSS	EVSS					EVCC	P20_3	P20_10	P20_2	K
L	P12_3	P12_5	P0_3	P13_5					BVSS	EVSS	EVSS	EVSS					REGVCC	P20_13	P20_12	P20_11	L
M	P0_0	P0_1	P0_6	P13_7					EVSS	EVSS	EVSS	EVSS					ISOVSS	P9_3	P9_4	P20_14	M
N	P0_4	P0_5	P12_4	P0_2													EVSS	P9_0	P9_1	P9_2	N
P	P0_11	P0_12	P0_14	EVCC													AP0_6	AP0_3	AP0_1	AP0_0	P
R	P0_13	P1_1	P2_6	P1_3													A0VREF	AP0_7	AP0_4	AP0_2	R
T	P1_0	P1_12	P2_8	P2_9													A0VSS	AP0_10	AP0_8	AP0_5	T
U	P1_2	P2_7	P2_11	P8_12	JP0_2	P1_9	EVCC	AWOVCL	REGVCC	P0_9	ISOVSS	ISOVCL	P1_15	EVSS	EVCC	P3_7	P3_10	AP0_13	AP0_11	AP0_9	U
V	P1_13	P2_12	P8_10	JP0_3	JP0_0	P1_11	P2_13	P3_0	FLMD0	P0_8	P0_7	P2_5	P8_1	P3_1	P8_7	P3_3	P3_6	P3_9	AP0_14	AP0_12	V
W	P2_10	P8_2	JP0_5	JP0_1	P2_0	P2_14	P2_15	IP0_0	P2_3	P2_2	JP0_6	P1_4	P2_4	P8_0	P8_4	P3_2	P8_8	P3_4	P3_8	AP0_15	W
Y	EVSS	P8_11	JP0_4	P2_1	P1_10	P1_8	$\overline{\text{RESET}}$	XT1	AWOVSS	X2	X1	P0_10	P1_5	P1_14	P8_3	P8_5	P8_6	P8_9	P3_5	A0VSS	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Figure 2B.5 Pin Connection Diagram (272-Pin FPBGA)

Table 2B.1 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / $\overline{\text{CSIH1SSI}}$ / TAPA0VN
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0TRG2 / $\overline{\text{CSIG0SSI}}$ / ADCA0SEL0 / TAPA0WP
3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / CSIG0RYI / CSIG0RYO / ADCA0SEL1 / TAPA0WN
4	ISOVCL
5	ISOVSS
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
14	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / PWGA34O
15	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
16	P0_13 / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5 / RLIN32RX
17	P0_14 / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX / RLIN32TX
18	EVSS
19	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
20	P8_10 / DPIN14 / PWGA42O / ADCA0I17S / CSIH3CSS3
21	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / ADCA0I18S / CSIH1CSS4
22	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / ADCA0I19S / CSIH1CSS5
23	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT / RTCA0OUT
24	JP0_4 / $\overline{\text{DCUTRST}}$
25	JP0_3 / INTP3 / TAUJ0I2 / TAUJ0O2 / DCUTMS / CSCXFOUT
26	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
27	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
28	JP0_0 / INTP0 / FPDR / FPDT / DCUTDI / LPDI/ LPDIO / TAUJ2I0 / TAUJ2O0
29	$\overline{\text{RESET}}$
30	EVCC
31	AWOVSS
32	AWOVCL
33	REGVCC
34	X2
35	X1
36	FLMD0
37	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
38	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
39	P0_8 / RLIN21TX / DPIN6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX / CSIH0CSS6
40	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
41	EVSS
42	ISOVSS
43	ISOVCL

Table 2B.1 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
44	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
45	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
46	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / PWGA37O / ADCA0I7S / INTP9
47	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
48	P8_7 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S / CSIH3CSS0
49	P8_8 / PWGA40O / ADCA0SEL1 / ADCA0I15S / CSIH3CSS1
50	P8_9 / PWGA41O / ADCA0SEL2 / ADCA0I16S / CSIH3CSS2
51	A0VSS
52	A0VREF
53	AP0_15 / ADCA0I15
54	AP0_14 / ADCA0I14
55	AP0_13 / ADCA0I13
56	AP0_12 / ADCA0I12
57	AP0_11 / ADCA0I11
58	AP0_10 / ADCA0I10
59	AP0_9 / ADCA0I9
60	AP0_8 / ADCA0I8
61	AP0_7 / ADCA0I7
62	AP0_6 / ADCA0I6
63	AP0_5 / ADCA0I5
64	AP0_4 / ADCA0I4
65	AP0_3 / ADCA0I3
66	AP0_2 / ADCA0I2
67	AP0_1 / ADCA0I1
68	AP0_0 / ADCA0I0
69	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA / ADCA0I2S
70	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL / ADCA0I3S
71	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
72	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
73	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
74	ISOVSS
75	REGVCC
76	EVCC
77	ISOVCL
78	ISOVSS
79	EVSS
80	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MODE2
81	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / TAUJ3I1 / TAUJ3O1
82	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / TAUJ3I2 / TAUJ3O2 / FLMD1
83	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / FLXA0RXDB
84	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
85	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1
86	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3

Table 2B.1 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
87	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / CAN7TX / RLIN32RX
88	P10_14 / PWGA19O / FLXA0RXDA / TAUB0I7 / TAUB0O7 / CAN7RX / INTP9 / RLIN32TX / $\overline{\text{CSIH3SSI}}$
89	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / PWGA26O / TAUB0I13 / TAUB0O13 / CSIH0CSS7
90	P11_2 / CSIH2SO / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / RLIN32RX
91	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / RLIN32TX
92	P11_4 / CSIH2SI / CAN3TX / PWGA29O
93	P11_5 / CAN5RX / INTP5 / PWGA30O / CSIH3SI
94	P11_6 / INTP13 / CAN5TX / PWGA31O / CSIH3SO
95	P11_7 / INTP5 / PWGA32O / CSIH3SC
96	EVCC
97	EVSS
98	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / PWGA0O / CSIH1SI / CSCXFOUT / TAUJ1I3 / TAUJ1O3 / TAPA0UP
99	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / CSIH1SC / MODE0 / TAUJ3I0 / TAUJ3O0 / TAPA0UN
100	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / CSIH1SO / MODE1 / TAPA0VP

Table 2B.2 Pin Assignment 144-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O
3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / PWGA54O
4	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O
7	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O
8	P11_10 / CSIG1SC / PWGA50O
9	P11_11 / CSIG1SI / RLIN25TX / PWGA51O
10	P11_12 / RLIN25RX / PWGA52O
11	ISOVCL
12	ISOVSS
13	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
14	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
15	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
16	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
17	EVCC
18	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 /DPIN8 / TAUB0I12 / TAUB0O12
19	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
20	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
21	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / PWGA34O
22	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
23	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
24	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
25	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0
26	P1_1 / INTP18 / RLIN33TX / TAUJ2I1 / TAUJ2O1
27	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2
28	P1_3 / INTP19 / CAN3TX / DPIN23 / TAUJ2I3 / TAUJ2O3
29	EVSS
30	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
31	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
32	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
33	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
34	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
35	JP0_4 / $\overline{\text{DCUTRST}}$
36	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
37	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
38	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
39	JP0_0 / INTP0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO / TAUJ2I0 / TAUJ2O0
40	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
41	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
42	P1_9 / DPIN20 / INTP21

Table 2B.2 Pin Assignment 144-Pin LQFP

Pin No.	Pin Name
43	P1_8
44	$\overline{\text{RESET}}$
45	EVCC
46	XT1
47	IP0_0 / XT2
48	AWOVSS
49	AWOVCL
50	REGVCC
51	X2
52	X1
53	FLMD0
54	JP0_6 / $\overline{\text{EVTO}}$
55	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
56	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
57	P0_8/ INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
58	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
59	EVSS
60	ISOVSS
61	ISOVCL
62	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20
63	P1_4 / RLIN35RX / INTP15 / DPIN18
64	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / ADCA0I0S / RIIC1SDA / SENT0RX
65	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / ADCA0I1S / RIIC1SCL / SENT0SPCO
66	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
67	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
68	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
69	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
70	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
71	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
72	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
73	A0VSS
74	A0VREF
75	AP0_15 / ADCA0I15
76	AP0_14 / ADCA0I14
77	AP0_13 / ADCA0I13
78	AP0_12 / ADCA0I12
79	AP0_11 / ADCA0I11
80	AP0_10 / ADCA0I10
81	AP0_9 / ADCA0I9
82	AP0_8 / ADCA0I8
83	AP0_7 / ADCA0I7
84	AP0_6 / ADCA0I6
85	AP0_5 / ADCA0I5
86	AP0_4 / ADCA0I4

Table 2B.2 Pin Assignment 144-Pin LQFP

Pin No.	Pin Name
87	AP0_3 / ADCA0I3
88	AP0_2 / ADCA0I2
89	AP0_1 / ADCA0I1
90	AP0_0 / ADCA0I0
91	EVSS
92	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA
93	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL
94	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
95	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S
96	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
97	ISOVSS
98	REGVCC
99	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
100	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9
101	EVCC
102	AP1_7 / ADCA1I7
103	AP1_6 / ADCA1I6
104	AP1_5 / ADCA1I5
105	AP1_4 / ADCA1I4
106	AP1_3 / ADCA1I3
107	AP1_2 / ADCA1I2
108	AP1_1 / ADCA1I1
109	AP1_0 / ADCA1I0
110	A1VREF
111	A1VSS
112	BVCC
113	ISOVCL
114	ISOVSS
115	P18_0 / CSIG1RYI / CSIG1RYO / PWGA61O / ADCA1I0S / TAUJ3I0 / TAUJ3O0
116	P18_1 / PWGA62O / ADCA1I1S / TAUJ3I1 / TAUJ3O1
117	P18_2 / PWGA63O / ADCA1I2S / TAUJ3I2 / TAUJ3O2
118	P18_3 / ADCA1I3S / TAUJ3I3 / TAUJ3O3
119	BVSS
120	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1 / RLIN24RX / MODE2
121	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / RLIN24TX / TAUJ3I1 / TAUJ3O1
122	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / FLMD1 / TAUJ3I2 / TAUJ3O2
123	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / FLXA0RXDB
124	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
125	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1
126	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
127	P10_13 / $\overline{\mathrm{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / CAN7TX
128	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\mathrm{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / CAN7RX / INTP9

Table 2B.2 Pin Assignment 144-Pin LQFP

Pin No.	Pin Name
129	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13
130	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / SFMA0IO3
131	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / RLIN32TX / SFMA0IO2
132	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / SFMA0IO1
133	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / SFMA0IO0
134	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / SFMA0SSL
135	P11_7 / INTP5 / PWGA32O / CSIH3SC / SFMA0CLK
136	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O
137	P12_0 / CAN2TX / PWGA56O
138	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O
139	P12_2 / INTP19 / RLIN34TX / PWGA58O
140	BVCC
141	BVSS
142	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / TAUJ1I3 / TAUJ1O3
143	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0 / TAUJ3I0 / TAUJ3O0
144	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Table 2B.3 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK / RLIN37RX / INTP17
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O
4	BVCC
5	BVSS
6	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
7	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
8	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
9	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
10	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
11	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
12	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
13	ISOVCL
14	ISOVSS
15	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
16	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
17	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
18	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
19	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
20	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
21	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
22	EVCC
23	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 /DPIN8 / TAUB0I12 / TAUB0O12
24	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
25	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
26	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
27	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
28	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
29	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
30	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0
31	P1_1 / INTP18 / RLIN33TX / TAUJ2I1 / TAUJ2O1
32	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2
33	P1_3 / INTP19 / CAN3TX / DPIN23 / TAUJ2I3 / TAUJ2O3
34	P1_12 / CAN4RX / INTP4 / RLIN36TX
35	P1_13 / CAN4TX / RLIN36RX / INTP16
36	P2_6 / ADCA0SEL2
37	EVSS
38	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
39	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
40	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
41	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
42	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT

Table 2B.3 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
43	JP0_4 / $\overline{\text{DCUTRST}}$
44	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
45	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
46	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
47	JP0_0 / INTP0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO / TAUJ2I0 / TAUJ2O0
48	P2_1 / RLIN27TX / CAN6TX
49	P2_0 / RLIN27RX / INTP6 / CAN6RX
50	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
51	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
52	P1_9 / DPIN20 / INTP21
53	P1_8
54	$\overline{\text{RESET}}$
55	EVCC
56	XT1
57	IP0_0 / XT2
58	AWOVSS
59	AWOVCL
60	REGVCC
61	X2
62	X1
63	FLMD0
64	P2_3 / RLIN28TX
65	P2_2 / RLIN28RX
66	JP0_6 / $\overline{\text{EVTO}}$
67	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
68	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
69	P0_8/ INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
70	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
71	EVSS
72	ISOVSS
73	ISOVCL
74	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20
75	P1_4 / RLIN35RX / INTP15 / DPIN18
76	P2_4 / RLIN29RX / ADCA0SEL0
77	P2_5 / RLIN29TX / ADCA0SEL1
78	P1_14 / RLIN23RX / CAN7RX / INTP9
79	P1_15 / RLIN23TX / CAN7TX
80	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / ADCA0I0S / RIIC1SDA / SENT0RX
81	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / ADCA0I1S / RIIC1SCL / SENT0SPCO
82	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
83	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
84	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
85	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
86	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S

Table 2B.3 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
87	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
88	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
89	A0VSS
90	A0VREF
91	AP0_15 / ADCA0I15
92	AP0_14 / ADCA0I14
93	AP0_13 / ADCA0I13
94	AP0_12 / ADCA0I12
95	AP0_11 / ADCA0I11
96	AP0_10 / ADCA0I10
97	AP0_9 / ADCA0I9
98	AP0_8 / ADCA0I8
99	AP0_7 / ADCA0I7
100	AP0_6 / ADCA0I6
101	AP0_5 / ADCA0I5
102	AP0_4 / ADCA0I4
103	AP0_3 / ADCA0I3
104	AP0_2 / ADCA0I2
105	AP0_1 / ADCA0I1
106	AP0_0 / ADCA0I0
107	EVSS
108	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA
109	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL
110	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
111	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S
112	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
113	ISOVSS
114	REGVCC
115	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
116	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
117	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
118	P20_0 / RLIN26RX / PWGA64O / INTP6 / CAN6RX / CSIG3SI
119	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
120	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
121	EVCC
122	AP1_11 / ADCA1I11
123	AP1_10 / ADCA1I10
124	AP1_9 / ADCA1I9
125	AP1_8 / ADCA1I8
126	AP1_7 / ADCA1I7
127	AP1_6 / ADCA1I6
128	AP1_5 / ADCA1I5
129	AP1_4 / ADCA1I4
130	AP1_3 / ADCA1I3

Table 2B.3 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
131	AP1_2 / ADCA1I2
132	AP1_1 / ADCA1I1
133	AP1_0 / ADCA1I0
134	AP1_15 / ADCA1I15
135	AP1_14 / ADCA1I14
136	AP1_13 / ADCA1I13
137	AP1_12 / ADCA1I12
138	A1VREF
139	A1VSS
140	BVCC
141	ISOVCL
142	ISOVSS
143	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / ADCA1I0S / TAUJ3I0 / TAUJ3O0
144	P18_1 / PWGA62O / ETNB0TXD0 / ADCA1I1S / TAUJ3I1 / TAUJ3O1
145	P18_2 / PWGA63O / ETNB0TXD1 / ADCA1I2S / TAUJ3I2 / TAUJ3O2
146	P18_3 / PWGA71O / ETNB0TXD2 / ADCA1I3S / TAUJ3I3 / TAUJ3O3
147	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
148	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
149	P18_6 / ADCA1I6S
150	P18_7 / ETNB0TXCLK / ADCA1I7S
151	BVSS
152	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1 / MEMC0AD0 / RLIN24RX / MODE2
153	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
154	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1 / TAUJ3I2 / TAUJ3O2
155	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
156	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
157	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
158	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
159	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
160	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
161	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
162	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3
163	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
164	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
165	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
166	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL

Table 2B.3 Pin Assignment 176-Pin LQFP

Pin No.	Pin Name
167	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
168	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$ / ETNB0RXERR / RLIN36TX
169	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
170	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
171	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
172	BVCC
173	BVSS
174	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1I3 / TAUJ1O3
175	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / MODE0 / TAUJ3I0 / TAUJ3O0
176	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
A1	BVSS
A2	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1I3 / TAUJ1O3
A3	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
A4	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
A5	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
A6	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
A7	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
A8	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
A9	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0 / RLIN24RX / MODE2
A10	P19_2 / ADCA1I18S
A11	P18_15 / ADCA1I15S
A12	P18_13 / ADCA1I13S
A13	P18_6 / ADCA1I6S
A14	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
A15	P18_10 / ADCA1I10S
A16	P18_8 / ADCA1I8S
A17	A1VSS
B1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK / RLIN37RX / INTP17
B2	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / MODE0 / TAUJ3I0 / TAUJ3O0
B3	P13_1 / MEMC0A20
B4	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
B5	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
B6	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
B7	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
B8	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
B9	P19_3 / ADCA1I19S
B10	P19_1 / ADCA1I17S
B11	P18_7 / ETNB0TXCLK / ADCA1I7S
B12	P18_11 / ADCA1I11S
B13	P18_3 / PWGA71O / ETNB0TXD2 / ADCA1I3S / TAUJ3I3 / TAUJ3O3
B14	P18_2 / PWGA63O / ETNB0TXD1 / ADCA1I2S / TAUJ3I2 / TAUJ3O2
B15	P18_1 / PWGA62O / ETNB0TXD0 / ADCA1I1S / TAUJ3I1 / TAUJ3O1
B16	AP1_12 / ADCA1I12
B17	AP1_14 / ADCA1I14
C1	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
C2	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
C3	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1
C4	P13_0 / MEMC0A19
C5	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
C6	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
C7	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3
C8	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
C9	P18_14 / ADCA1I14S
C10	P19_0 / ADCA1I16S
C11	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
C12	P18_12 / ADCA1I12S
C13	P18_9 / ADCA1I9S
C14	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / ADCA1I0S / TAUJ3I0 / TAUJ3O0
C15	AP1_13 / ADCA1I13
C16	AP1_15 / ADCA1I15
C17	AP1_0 / ADCA1I0
D1	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
D2	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
D3	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
D4	BVCC
D5	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$ / ETNB0RXERR / RLIN36TX
D6	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL
D7	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
D8	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1 / TAUJ3I2 / TAUJ3O2
D9	BVSS
D10	BVCC
D11	BVCC
D12	ISOVSS
D13	ISOVCL
D14	A1VSS
D15	AP1_1 / ADCA1I1
D16	AP1_2 / ADCA1I2
D17	AP1_3 / ADCA1I3
E1	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
E2	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
E3	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
E4	BVCC
E14	A1VREF
E15	AP1_5 / ADCA1I5
E16	AP1_6 / ADCA1I6
E17	AP1_8 / ADCA1I8

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
F1	P13_3 / ETNB0RXERR
F2	P13_2 / ETNB0RXDV
F3	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
F4	BVSS
F14	AP1_4 / ADCA1I4
F15	AP1_7 / ADCA1I7
F16	AP1_9 / ADCA1I9
F17	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
G1	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
G2	P13_4
G3	P13_5 / MEMC0A21
G4	ISOVCL
G7	BVSS
G8	BVSS
G9	BVSS
G10	BVSS
G11	BVSS
G14	AP1_10 / ADCA1I10
G15	AP1_11 / ADCA1I11
G16	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX
G17	P20_0 / RLIN26RX / PWGA64O / CAN6RX / INTP6 / CSIG3SI
H1	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
H2	P13_7 / PWGA73O
H3	P13_6 / MEMC0A22 / PWGA72O
H4	ISOVSS
H7	BVSS
H8	BVSS
H9	BVSS
H10	BVSS
H11	EVSS
H14	EVCC
H15	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
H16	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
H17	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
J1	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
J2	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
J3	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
J4	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
J7	BVSS
J8	BVSS
J9	BVSS
J10	EVSS
J11	EVSS
J14	REGVCC
J15	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
J16	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
J17	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
K1	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
K2	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
K3	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
K4	EVCC
K7	EVSS
K8	EVSS
K9	EVSS
K10	EVSS
K11	EVSS
K14	ISOVSS
K15	AP0_0 / ADCA0I0
K16	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA
K17	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL
L1	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
L2	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
L3	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
L4	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
L7	EVSS
L8	EVSS
L9	EVSS
L10	EVSS
L11	EVSS
L14	EVSS
L15	AP0_4 / ADCA0I4
L16	AP0_2 / ADCA0I2
L17	AP0_1 / ADCA0I1
M1	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
M2	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0
M3	P2_9 / PWGA77O
M4	P2_7 / RLIN210RX
M14	A0VREF
M15	AP0_8 / ADCA0I8
M16	AP0_5 / ADCA0I5
M17	AP0_3 / ADCA0I3
N1	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2
N2	P1_1 / INTP18 / RLIN33TX / TAUJ2I1 / TAUJ2O1
N3	P1_3 / INTP19 / CAN3TX / DPIN23 / TAUJ2I3 / TAUJ2O3
N4	P2_11 / PWGA79O
N14	A0VSS
N15	AP0_11 / ADCA0I11
N16	AP0_7 / ADCA0I7

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
N17	AP0_6 / ADCA0I6
P1	P1_12 / CAN4RX / INTP4 / RLIN36TX
P2	P1_13 / CAN4TX / RLIN36RX / INTP16
P3	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
P4	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
P5	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
P6	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
P7	P2_13 / RLIN211TX
P8	P2_15 / PWGA75O
P9	EVCC
P10	REGVCC
P11	ISOVSS
P12	ISOVCL
P13	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
P14	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
P15	AP0_13 / ADCA0I13
P16	AP0_10 / ADCA0I10
P17	AP0_9 / ADCA0I9
R1	P2_6 / ADCA0SEL2
R2	P2_10 / PWGA78O
R3	JP0_4 / $\overline{\text{DCUTRST}}$
R4	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
R5	P2_1 / RLIN27TX / CAN6TX
R6	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
R7	P1_9 / DPIN20 / INTP21
R8	P3_0 / PWGA76O
R9	FLMD0
R10	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
R11	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
R12	P2_5 / RLIN29TX / ADCA0SEL1
R13	P1_15 / RLIN23TX / CAN7TX
R14	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
R15	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
R16	AP0_14 / ADCA0I14
R17	AP0_12 / ADCA0I12
T1	P2_8 / RLIN210TX
T2	P2_12 / RLIN211RX
T3	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
T4	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
T5	P2_0 / RLIN27RX / CAN6RX / INTP6
T6	P2_14 / PWGA74O
T7	IP0_0 / XT2
T8	AWOVCL
T9	X1
T10	P2_2 / RLIN28RX

Table 2B.4 Pin Assignment 233-Pin FPBGA

Pin No.	Pin Name
T11	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
T12	P0_8 / INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
T13	P2_4 / RLIN29RX / ADCA0SEL0
T14	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / ADCA0I1S / RIIC1SCL / SENT0SPCO
T15	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
T16	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
T17	AP0_15 / ADCA0I15
U1	EVSS
U2	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
U3	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
U4	JP0_0 / INTP0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO / TAUJ2I0 / TAUJ2O0
U5	P1_8
U6	$\overline{\text{RESET}}$
U7	XT1
U8	AWOVSS
U9	X2
U10	P2_3 / RLIN28TX
U11	JP0_6 / $\overline{\text{EVTO}}$
U12	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20
U13	P1_4 / RLIN35RX / INTP15 / DPIN18
U14	P1_14 / RLIN23RX / CAN7RX / INTP9
U15	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / ADCA0I0S / RIIC1SDA / SENT0RX
U16	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 /CSIH0CSS1 / INTP7/ PWGA23O / CAN7TX / ADCA0I5S
U17	A0VSS

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
A1	BVSS
A2	P22_7
A3	P13_1 / MEMC0A20
A4	P22_9
A5	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$ / ETNB0RXERR / RLIN36TX
A6	P22_12
A7	P22_13
A8	P22_15
A9	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / INTP20 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
A10	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
A11	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4 / TAUJ3I3 / TAUJ3O3
A12	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0 / RLIN24RX / MODE2
A13	P19_3 / ADCA1I19S
A14	P19_0 / ADCA1I16S
A15	P18_14 / ADCA1I14S
A16	P18_6 / ADCA1I6S
A17	P18_3 / PWGA71O / ETNB0TXD2 / ADCA1I3S / TAUJ3I3 / TAUJ3O3
A18	P18_9 / ADCA1I9S
A19	P18_0 / CSIG1RYI / CSIG1RYO / ETNB0LINK / PWGA61O / ADCA1I0S / TAUJ3I0 / TAUJ3O0
A20	A1VSS
B1	P22_6
B2	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK / RLIN37RX / INTP17
B3	P10_1 / TAUD0I3 / TAUD0O3 / INTP18 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / ETNB0RXD0 / MEMC0A20 / MODE0 / TAUJ3I0 / TAUJ3O0
B4	P22_8
B5	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
B6	P22_11
B7	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13 / SFMA0IO0
B8	P11_4 / CSIH2SI / CAN3TX / INTP21 / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12 / SFMA0IO1
B9	P21_1
B10	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8 / CAN7RX / INTP9
B11	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1 / TAUJ3I2 / TAUJ3O2
B12	P19_2 / ADCA1I18S
B13	P19_1 / ADCA1I17S
B14	P18_7 / ETNB0TXCLK / ADCA1I7S
B15	P18_13 / ADCA1I13S
B16	P18_4 / CSIH1CSS4 / ETNB0TXD3 / ADCA1I4S
B17	P18_2 / PWGA63O / ETNB0TXD1 / ADCA1I2S / TAUJ3I2 / TAUJ3O2
B18	P18_1 / PWGA62O / ETNB0TXD0 / ADCA1I1S / TAUJ3I1 / TAUJ3O1
B19	AP1_12 / ADCA1I12

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
B20	AP1_13 / ADCA1I13
C1	P22_4
C2	P22_5
C3	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / ETNB0RXD1 / MEMC0A21 / RLIN37TX / MODE1
C4	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19 / ETNB0RXCLK / TAUJ1I3 / TAUJ1O3
C5	P12_2 / INTP19 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18 / CSIG2RYI / CSIG2RYO
C6	P22_10
C7	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14 / SFMA0SSL
C8	P22_14
C9	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11 / RLIN32TX / SFMA0IO2
C10	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7 / CAN7TX
C11	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1 / RLIN24TX / TAUJ3I1 / TAUJ3O1
C12	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
C13	P18_15 / ADCA1I15S
C14	P18_12 / ADCA1I12S
C15	P18_11 / ADCA1I11S
C16	P18_10 / ADCA1I10S
C17	P18_8 / ADCA1I8S
C18	AP1_14 / ADCA1I14
C19	AP1_15 / ADCA1I15
C20	AP1_0 / ADCA1I0
D1	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
D2	P22_3
D3	P10_5 / TAUD0I11 / TAUD0O11 / CAN6RX / INTP6 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO / ETNB0RXD3 / PWGA54O
D4	BVCC
D5	BVCC
D6	P13_0 / MEMC0A19
D7	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / $\overline{\text{CSIG2SSI}}$ / MEMC0A16 / RLIN36RX / INTP16
D8	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15 / SFMA0CLK
D9	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10 / SFMA0IO3
D10	BVSS
D11	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
D12	BVCC
D13	BVSS
D14	P18_5 / CSIH1CSS5 / ETNB0TXEN / ADCA1I5S
D15	ISOVSS
D16	ISOVCL
D17	BVCC

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
D18	AP1_1 / ADCA1I1
D19	AP1_2 / ADCA1I2
D20	AP1_3 / ADCA1I3
E1	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
E2	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
E3	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / CAN6TX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$ / PWGA53O / ETNB0RXD2 / MEMC0A22
E4	BVCC
E17	A1VSS
E18	AP1_4 / ADCA1I4
E19	AP1_5 / ADCA1I5
E20	AP1_6 / ADCA1I6
F1	P22_2
F2	P22_1
F3	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
F4	BVCC
F17	A1VREF
F18	AP1_7 / ADCA1I7
F19	AP1_9 / ADCA1I9
F20	AP1_10 / ADCA1I10
G1	P22_0
G2	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$ / ETNB0RXDV
G3	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
G4	BVSS
G17	AP1_8 / ADCA1I8
G18	AP1_11 / ADCA1I11
G19	P20_6 / PWGA88O
G20	P20_7 / PWGA89O
H1	P21_4
H2	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
H3	P21_0
H4	ISOVCL
H17	EVCC
H18	P20_8 / PWGA90O
H19	P20_9 / PWGA91O
H20	P20_4 / RLIN23RX / INTP22 / PWGA59O / CAN7RX / INTP9 / $\overline{\text{CSIG3SSI}}$
J1	P13_3 / ETNB0RXERR
J2	P21_3
J3	P21_2
J4	ISOVSS
J9	BVSS
J10	BVSS
J11	BVSS
J12	BVSS
J17	EVSS
J18	P20_5 / RLIN23TX / INTP23 / PWGA60O / CAN7TX

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
J19	P20_0 / RLIN26RX / PWGA64O / CAN6RX / INTP6 / CSIG3SI
J20	P20_1 / RLIN26TX / PWGA65O / CAN6TX / CSIG3SO
K1	P13_4
K2	P13_6 / MEMC0A22 / PWGA72O
K3	P13_2 / ETNB0RXDV
K4	BVSS
K9	BVSS
K10	BVSS
K11	BVSS
K12	EVSS
K17	EVCC
K18	P20_3 / CAN4TX / PWGA67O / RLIN29TX / CSIG3RYI / CSIG3RYO
K19	P20_10 / PWGA92O
K20	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX / CSIG3SC
L1	P12_3 / RLIN27RX / PWGA68O / CSIG2SI / $\overline{\text{MEMC0BEN0}}$ / TAUB1I6 / TAUB1O6
L2	P12_5 / PWGA70O / ETNB0MDC / CSIG2SO / TAUB1I4 / TAUB1O4
L3	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
L4	P13_5 / MEMC0A21
L9	BVSS
L10	EVSS
L11	EVSS
L12	EVSS
L17	REGVCC
L18	P20_13 / PWGA95O
L19	P20_12 / PWGA94O
L20	P20_11 / PWGA93O
M1	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
M2	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
M3	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
M4	P13_7 / MEMC0A23 / PWGA73O
M9	EVSS
M10	EVSS
M11	EVSS
M12	EVSS
M17	ISOVSS
M18	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / INTP16 / ADCA0I10S
M19	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / INTP17 / ADCA0I11S
M20	P20_14
N1	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
N2	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
N3	P12_4 / RLIN27TX / PWGA69O / CSIG2SC / ETNB0MDIO / $\overline{\text{MEMC0BEN1}}$
N4	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
N17	EVSS

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
N18	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S / TAUJ1I1 / TAUJ1O1 / SENT1RX / RIIC1SDA
N19	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / RIIC1SCL
N20	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
P1	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
P2	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
P3	P0_14 / INTP17 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
P4	EVCC
P17	AP0_6 / ADCA0I6
P18	AP0_3 / ADCA0I3
P19	AP0_1 / ADCA0I1
P20	AP0_0 / ADCA0I0
R1	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
R2	P1_1 / INTP18 / RLIN33TX / TAUJ2I1 / TAUJ2O1
R3	P2_6 / ADCA0SEL2
R4	P1_3 / INTP19 / CAN3TX / DPIN23 / TAUJ2I3 / TAUJ2O3
R17	A0VREF
R18	AP0_7 / ADCA0I7
R19	AP0_4 / ADCA0I4
R20	AP0_2 / ADCA0I2
T1	P1_0 / RLIN33RX / INTP13 / TAUJ2I0 / TAUJ2O0
T2	P1_12 / CAN4RX / INTP4 / RLIN36TX
T3	P2_8 / RLIN210TX
T4	P2_9 / PWGA77O
T17	A0VSS
T18	AP0_10 / ADCA0I10
T19	AP0_8 / ADCA0I8
T20	AP0_5 / ADCA0I5
U1	P1_2 / CAN3RX / INTP3 / DPIN19 / TAUJ2I2 / TAUJ2O2
U2	P2_7 / RLIN210RX
U3	P2_11 / PWGA79O
U4	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / INTP23 / RLIN25TX / ADCA0I19S
U5	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
U6	P1_9 / DPIN20 / INTP21
U7	EVCC
U8	AWOVCL
U9	REGVCC
U10	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
U11	ISOVSS
U12	ISOVCL
U13	P1_15 / RLIN23TX / CAN7TX
U14	EVSS
U15	EVCC
U16	P3_7 / PWGA86O
U17	P3_10

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
U18	AP0_13 / ADCA0I13
U19	AP0_11 / ADCA0I11
U20	AP0_9 / ADCA0I9
V1	P1_13 / CAN4TX / RLIN36RX / INTP16
V2	P2_12 / RLIN211RX
V3	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / RLIN37RX / INTP17 / ADCA0I17S
V4	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
V5	JP0_0 / INTP0 / FPDR / FPDT / DCUTDI / LPDI / LPDIO / TAUJ2I0 / TAUJ2O0
V6	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22 / INTP14
V7	P2_13 / RLIN211TX
V8	P3_0 / PWGA76O
V9	FLMD0
V10	P0_8 / INTP16 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
V11	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
V12	P2_5 / RLIN29TX / ADCA0SEL1
V13	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / CAN6TX / ADCA0I1S / RIIC1SCL / SENT0SPCO
V14	P3_1 / PWGA80O
V15	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
V16	P3_3 / PWGA82O
V17	P3_6 / PWGA85O
V18	P3_9
V19	AP0_14 / ADCA0I14
V20	AP0_12 / ADCA0I12
W1	P2_10 / PWGA78O
W2	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / RLIN37TX / ADCA0I4S
W3	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
W4	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
W5	P2_0 / RLIN27RX / CAN6RX / INTP6
W6	P2_14 / PWGA74O
W7	P2_15 / PWGA75O
W8	IP0_0 / XT2
W9	P2_3 / RLIN28TX
W10	P2_2 / RLIN28RX
W11	JP0_6 / $\overline{\text{EVTO}}$
W12	P1_4 / RLIN35RX / INTP15 / DPIN18
W13	P2_4 / RLIN29RX / ADCA0SEL0
W14	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / CAN6RX / INTP6 / ADCA0I0S / RIIC1SDA / SENT0RX
W15	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / CAN7RX / INTP9 / ADCA0I6S
W16	P3_2 / PWGA81O
W17	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / RLIN34RX / INTP14 / ADCA0I15S
W18	P3_4 / PWGA83O
W19	P3_8 / PWGA87O
W20	AP0_15 / ADCA0I15
Y1	EVSS

Table 2B.5 Pin Assignment 272-Pin FPBGA

Pin No.	Pin Name
Y2	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / RLIN25RX / ADCA0I18S
Y3	JP0_4 / $\overline{\text{DCUTRST}}$
Y4	P2_1 / RLIN27TX / CAN6TX
Y5	P1_10 / RLIN24RX / DPIN21 / INTP22 / ADCA1TRG1
Y6	P1_8
Y7	$\overline{\text{RESET}}$
Y8	XT1
Y9	AWOVSS
Y10	X2
Y11	X1
Y12	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
Y13	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17 / INTP20
Y14	P1_14 / RLIN23RX / CAN7RX / INTP9
Y15	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / CAN7TX / ADCA0I5S
Y16	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
Y17	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
Y18	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / RLIN34TX / ADCA0I16S
Y19	P3_5 / PWGA84O
Y20	A0VSS

2B.2 Pin Description

Table 2B.6 Pin Functions

	No. of Pins							
Pin Name	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	IO	Pin Function	Unit
AnVREF	✓	✓	✓	✓	✓	—	ADCAn voltage supply and reference voltage	ADCAn
	n = 0	n = 0, 1	n = 0, 1	n = 0, 1	n = 0, 1			
AnVSS	✓	✓	✓	✓	✓	—	ADCAn ground	
	n = 0	n = 0, 1	n = 0, 1	n = 0, 1	n = 0, 1			
ADCA0Im	✓	✓	✓	✓	✓	I	ADCA0 input channel m with 12-bit resolution	
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
ADCA1Im	—	✓	✓	✓	✓	I	ADCA1 input channel m with 12-bit resolution	
		m = 0 to 7	m = 0 to 15	m = 0 to 15	m = 0 to 15			
ADCA0ImS	✓	✓	✓	✓	✓	I	ADCA0 input channel m with 10-bit resolution	
	m = 2 to 11, 14 to 19	m = 0 to 11, 14 to 19	m = 0 to 11, 14 to 19	m = 0 to 11, 14 to 19	m = 0 to 11, 14 to 19			
ADCA1ImS	—	✓	✓	✓	✓	I	ADCA1 input channel m with 10-bit resolution	
		m = 0 to 3	m = 0 to 7	m = 0 to 19	m = 0 to 19			
ADCA0SELy	✓	✓	✓	✓	✓	O	Selection pin y for ADCA0 input and external MPX	
	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2			
ADCAnTRGy	✓	✓	✓	✓	✓	I	ADCAn external trigger pin y	
	n = 0, y = 0 to 2	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2			
AP0_m	✓	✓	✓	✓	✓	IO	Analog port 0_m	Port
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
AP1_m	—	✓	✓	✓	✓	IO	Analog port 1_m	
		m = 0 to 7	m = 0 to 15	m = 0 to 15	m = 0 to 15			
APO	✓	✓	✓	✓	✓	O	Port output signal for analog input	LPS0
AWOVCL	✓	✓	✓	✓	✓	—	Voltage regulator for Always-On area (AWO area) capacitor connection	Power
AWOVSS	✓	✓	✓	✓	✓	—	Internal logic for Always-On area (AWO area) ground	
BVCC	—	✓	✓	✓	✓	—	Port buffer voltage supply	
BVSS	—	✓	✓	✓	✓	—	Port buffer ground	
CANzRX	✓	✓	✓	✓	✓	I	CANz receive data input	RCFDCn
	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7			
CANzTX	✓	✓	✓	✓	✓	O	CANz transmit data output	
	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7			
CSCXFOUT	✓	✓	✓	✓	✓	O	Clock output	Clock
CSIGnRYI	✓	✓	✓	✓	✓	I	CSIGn ready (1) / busy (0) input signal	CSIGn
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIGnRYO	✓	✓	✓	✓	✓	O	CSIGn ready (1) / busy (0) output signal	
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIGnSC	✓	✓	✓	✓	✓	IO	CSIGn serial clock signal	
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIGnSI	✓	✓	✓	✓	✓	I	CSIGn serial data input	
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIGnSO	✓	✓	✓	✓	✓	O	CSIGn serial data output	
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{CSIGnSSI}}$	✓	✓	✓	✓	✓	I	CSIGn SS function control input signal	
	n = 0	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS0	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS1	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 1	
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			

Table 2B.6 Pin Functions

Pin Name	No. of Pins					IO	Pin Function	Unit
	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins			
CSIHnCSS2	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 2	CSIHn
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS3	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 3	
	n = 0, 2, 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS4	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 4	
	n = 0, 1	n = 0 to 2	n = 0 to 2	n = 0 to 2	n = 0 to 2			
CSIHnCSS5	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 5	
	n = 0, 1	n = 0 to 2	n = 0 to 2	n = 0 to 2	n = 0 to 2			
CSIHnCSS6	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 6	
	n = 0	n = 0	n = 0	n = 0	n = 0			
CSIHnCSS7	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 7	
	n = 0	n = 0	n = 0	n = 0	n = 0			
CSIHnRYI	✓	✓	✓	✓	✓	I	CSIHn ready (1) / busy (0) input signal	
	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnRYO	✓	✓	✓	✓	✓	O	CSIHn ready (1) / busy (0) output signal	
	n = 0, 1	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSC	✓	✓	✓	✓	✓	IO	CSIHn serial clock signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSI	✓	✓	✓	✓	✓	I	CSIHn serial data input	
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSO	✓	✓	✓	✓	✓	O	CSIHn serial data output	
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{CSIHnSSI}}$	✓	✓	✓	✓	✓	I	CSIHn slave select input signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{DCURDY}}$	✓	✓	✓	✓	✓	O	Debug ready	OCD
DCUTCK	✓	✓	✓	✓	✓	I	Debug clock	
DCUTDI	✓	✓	✓	✓	✓	I	Debug data input	
DCUTDO	✓	✓	✓	✓	✓	O	Debug data output	
DCUTMS	✓	✓	✓	✓	✓	I	Debug mode select	
$\overline{\text{DCUTRST}}$	✓	✓	✓	✓	✓	I	Debug reset	OCD
DPINm	✓	✓	✓	✓	✓	I	Digital port input m	LPS0
	m = 1 to 16	m = 0 to 23	m = 0 to 23	m = 0 to 23	m = 0 to 23			
DPO	✓	✓	✓	✓	✓	O	Port output signal for digital input	
ENCA0TINm	✓	✓	✓	✓	✓	I	ENCA0 capture trigger input m	ENCAn
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1			
ENCA0E0	✓	✓	✓	✓	✓	I	ENCA0 encoder input 0	
ENCA0E1	✓	✓	✓	✓	✓	I	ENCA0 encoder input 1	
ENCA0EC	✓	✓	✓	✓	✓	I	ENCA0 encoder clear input	
ETNB0LINK	—	—	✓	✓	✓	I	PHY link status	ETNBn
ETNB0MDC	—	—	✓	✓	✓	O	PHY management clock	
ETNB0MDIO	—	—	✓	✓	✓	IO	Management transmit / receive data signal	
ETNB0RXCLK	—	—	✓	✓	✓	I	MII receive clock	
ETNB0RXD[3:0]	—	—	✓	✓	✓	I	MII receive data input	
ETNB0RXDV	—	—	✓	✓	✓	I	MII receive data valid	
ETNB0RXERR	—	—	✓	✓	✓	I	MII receive error	

Table 2B.6 Pin Functions

	No. of Pins							
Pin Name	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	IO	Pin Function	Unit
ETNB0TXCLK	—	—	✓	✓	✓	I	MII transmit clock	ETNBn
ETNB0TXD[3:0]	—	—	✓	✓	✓	O	MII transmit data output	
ETNB0TXEN	—	—	✓	✓	✓	O	MII transmit data enable	
EVCC	✓	✓	✓	✓	✓	—	Port buffer voltage supply	Power
EVSS	✓	✓	✓	✓	✓	—	Port buffer ground	
$\overline{\text{EVTO}}$	—	✓	✓	✓	✓	O	Event output	TEU_OUT
FLMD0	✓	✓	✓	✓	✓	I	Operating mode select pin 0	Mode
FLMD1	✓	✓	✓	✓	✓	I	Operating mode select pin 1	
FLXA0RXDA	✓	✓	✓	✓	✓	I	FLXA0 channel A receive data input	FLXAn
FLXA0RXDB	✓	✓	✓	✓	✓	I	FLXA0 channel B receive data input	
FLXA0STPWT	✓	✓	✓	✓	✓	I	FLXA0 stop watch trigger input	
FLXA0TXDA	✓	✓	✓	✓	✓	O	FLXA0 channel A transmit data output	
FLXA0TXDB	✓	✓	✓	✓	✓	O	FLXA0 channel B transmit data output	
FLXA0TXENA	✓	✓	✓	✓	✓	O	FLXA0 channel A transmit enable	
FLXA0TXENB	✓	✓	✓	✓	✓	O	FLXA0 channel B transmit enable	
FPDR	✓	✓	✓	✓	✓	I	Serial Communication Interface RXD	FLASH
FPDT	✓	✓	✓	✓	✓	O	Serial Communication Interface TXD	
FPCK	✓	✓	✓	✓	✓	I	Serial Communication Interface clock	
INTPm	✓	✓	✓	✓	✓	I	External interrupt input m	INTC
	m = 0 to 13	m = 0 to 23	m = 0 to 23	m = 0 to 23	m = 0 to 23			
IP0_0	—	✓	✓	✓	✓	I	Input port 0_0	Port
ISOVCL	✓	✓	✓	✓	✓	—	Voltage regulator for Isolated area (ISO area) capacitor connection	Power
ISOVSS	✓	✓	✓	✓	✓	—	Internal logic for Isolated area (ISO area) ground	
JP0_m	✓	✓	✓	✓	✓	IO	JTAG port 0_m	JTAG
	m = 0 to 5	m = 0 to 6	m = 0 to 6	m = 0 to 6	m = 0 to 6			
KR0Im	✓	✓	✓	✓	✓	I	KR0 key input signal	KRn
	m = 0 to 7	m = 0 to 7	m = 0 to 7	m = 0 to 7	m = 0 to 7			
LPDCLK	✓	✓	✓	✓	✓	I	LPD clock input (4-pin mode)	LPD
LPDCLKOUT	✓	✓	✓	✓	✓	O	LPD clock output (4-pin mode)	
LPDI	✓	✓	✓	✓	✓	I	LPD data input (4-pin mode)	
LPDIO	✓	✓	✓	✓	✓	IO	LPD data input / output (1-pin mode)	
LPDO	✓	✓	✓	✓	✓	O	LPD data output (4-pin mode)	
MEMC0Am	—	—	✓	✓	✓	O	MEMC0 address m	MEMCn
			m = 16 to 22	m = 16 to 22	m = 16 to 23			
MEMC0ADm	—	—	✓	✓	✓	IO	MEMC0 address / data m	
			m = 0 to 15	m = 0 to 15	m = 0 to 15			
$\overline{\text{MEMC0ASTB}}$	—	—	✓	✓	✓	O	MEMC0 address strobe	
$\overline{\text{MEMC0BENm}}$	—	—	✓	✓	✓	O	MEMC0 byte enable m	
			m = 0, 1	m = 0, 1	m = 0, 1			
MEMC0CLK	—	—	✓	✓	✓	O	MEMC0 clock output	
$\overline{\text{MEMC0CSm}}$	—	—	✓	✓	✓	O	MEMC0 chip select m	
			m = 0 to 3	m = 0 to 3	m = 0 to 3			
$\overline{\text{MEMC0RD}}$	—	—	✓	✓	✓	O	MEMC0 read strobe	
$\overline{\text{MEMC0WAIT}}$	—	—	✓	✓	✓	I	MEMC0 wait input	
$\overline{\text{MEMC0WR}}$	—	—	✓	✓	✓	O	MEMC0 write strobe	
MODEm	✓	✓	✓	✓	✓	I	Sub operating mode select	Mode
	m = 0 to 2	m = 0 to 2	m = 0 to 2	m = 0 to 2	m = 0 to 2			
NMI	✓	✓	✓	✓	✓	I	External non-maskable interrupt input	INTC

Table 2B.6 Pin Functions

	No. of Pins							
Pin Name	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	IO	Pin Function	Unit
P0_m	✓ m = 0 to 14	✓ m = 0 to 14	✓ m = 0 to 14	✓ m = 0 to 14	✓ m = 0 to 14	IO	Port 0_m	Port
P1_m	—	✓ m = 0 to 5, 8 to 11	✓ m = 0 to 5, 8 to 15	✓ m = 0 to 5, 8 to 15	✓ m = 0 to 5, 8 to 15	IO	Port 1_m	
P2_m	—	—	✓ m = 0 to 6	✓ m = 0 to 15	✓ m = 0 to 15	IO	Port 2_m	
P3_m	—	—	—	✓ m = 0	✓ m = 0 to 10	IO	Port3_m	
P8_m	✓ m = 2 to 12	✓ m = 0 to 12	✓ m = 0 to 12	✓ m = 0 to 12	✓ m = 0 to 12	IO	Port 8_m	
P9_m	✓ m = 0 to 4	✓ m = 0 to 4	✓ m = 0 to 4	✓ m = 0 to 4	✓ m = 0 to 4	IO	Port 9_m	
P10_m	✓ m = 0 to 14	✓ m = 0 to 15	✓ m = 0 to 15	✓ m = 0 to 15	✓ m = 0 to 15	IO	Port 10_m	
P11_m	✓ m = 1 to 7	✓ m = 0 to 12, 15	✓ m = 0 to 12, 15	✓ m = 0 to 12, 15	✓ m = 0 to 12, 15	IO	Port 11_m	
P12_m	—	✓ m = 0 to 2	✓ m = 0 to 5	✓ m = 0 to 5	✓ m = 0 to 5	IO	Port 12_m	
P13_m	—	—	—	✓ m = 0 to 7	✓ m = 0 to 7	IO	Port 13_m	
P18_m	—	✓ m = 0 to 3	✓ m = 0 to 7	✓ m = 0 to 15	✓ m = 0 to 15	IO	Port 18_m	
P19_m	—	—	—	✓ m = 0 to 3	✓ m = 0 to 3	IO	Port19_m	
P20_m	—	✓ m = 4, 5	✓ m = 0 to 5	✓ m = 0 to 5	✓ m = 0 to 14	IO	Port 20_m	
P21_m	—	—	—	—	✓ m = 0 to 4	IO	Port21_m	
P22_m	—	—	—	—	✓ m = 0 to 15	IO	Port22_m	
PWGAnO	✓ n = 0 to 13, 16 to 23, 26 to 47	✓ n = 0 to 63	✓ n = 0 to 71	✓ n = 0 to 79	✓ n = 0 to 95	O	PWGAn output signal	PWGAn
REGVCC	✓	✓	✓	✓	✓	—	Voltage regulators voltage supply	Power
$\overline{\text{RESET}}$	✓	✓	✓	✓	✓	I	External reset input	Reset
$\overline{\text{RESETOUT}}$	✓	✓	✓	✓	✓	O	Reset output	
RIICnSCL	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	IO	RIICn serial clock	RIICn
RIICnSDA	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	✓ n = 0, 1	IO	RIICn serial data	
RLIN2mRX	✓ m = 0 to 2	✓ m = 0 to 5	✓ m = 0 to 9	✓ m = 0 to 11	✓ m = 0 to 11	I	RLIN2m receive data input	RLIN24n
RLIN2mTX	✓ m = 0 to 2	✓ m = 0 to 5	✓ m = 0 to 9	✓ m = 0 to 11	✓ m = 0 to 11	O	RLIN2m transmit data output	
RLIN3nRX	✓ n = 0 to 2	✓ n = 0 to 5	✓ n = 0 to 7	✓ n = 0 to 7	✓ n = 0 to 7	I	RLIN3n receive data input	RLIN3n
RLIN3nTX	✓ n = 0 to 2	✓ n = 0 to 5	✓ n = 0 to 7	✓ n = 0 to 7	✓ n = 0 to 7	O	RLIN3n transmit data output	
RTCA0OUT	✓	✓	✓	✓	✓	O	RTCA0 1Hz output	RTCAn

Table 2B.6 Pin Functions

	No. of Pins							
Pin Name	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	IO	Pin Function	Unit
SELDPk	✓	✓	✓	✓	✓	O	External multiplexer select signal output k for the digital port	LPS0
	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2			
SENTnRX	✓	✓	✓	✓	✓	I	SENT receive data input	RSENTn
	n = 1	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
SENTnSPCO	✓	✓	✓	✓	✓	O	SENT SPC Extension Output	
	n = 1	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
SFMA0CLK	—	✓	✓	✓	✓	O	SFMA0 clock	SFMAn
SFMA0IOm	—	✓	✓	✓	✓	IO	SFMA0 master data input / output	
		m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3			
SFMA0SSL	—	✓	✓	✓	✓	O	SFMA0 slave select	
TAPA0ESO	✓	✓	✓	✓	✓	I	Hi-Z control	TAPAn
TAPA0UN	✓	✓	✓	✓	✓	O	Motor control output U phase (negative)	
TAPA0UP	✓	✓	✓	✓	✓	O	Motor control output U phase (positive)	
TAPA0VN	✓	✓	✓	✓	✓	O	Motor control output V phase (negative)	
TAPA0VP	✓	✓	✓	✓	✓	O	Motor control output V phase (positive)	
TAPA0WN	✓	✓	✓	✓	✓	O	Motor control output W phase (negative)	
TAPA0WP	✓	✓	✓	✓	✓	O	Motor control output W phase (positive)	
TAUBnIm	✓	✓	✓	✓	✓	I	TAUBn channel input m	TAUBn
	n = 0, m = 0 to 8, 10, 12 to 15	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUBnOm	✓	✓	✓	✓	✓	O	TAUBn channel output m	
	n = 0, m = 0 to 8, 10, 12 to 15	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUD0Im	✓	✓	✓	✓	✓	I	TAUD0 channel input m	TAUDn
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUD0Om	✓	✓	✓	✓	✓	O	TAUD0 channel output m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUJnIm	✓	✓	✓	✓	✓	I	TAUJn channel input m	TAUJn
	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3			
TAUJnOm	✓	✓	✓	✓	✓	O	TAUJn channel output m	
	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3	n = 0 to 3, m = 0 to 3			
X1, X2	✓	✓	✓	✓	✓	—	Main OSC connections	MOSC
XT1, XT2	—	✓	✓	✓	✓	—	Sub OSC connections	SOSC

CAUTION

When pin functions for a peripheral module are allocated to multiple pins, use the pins from the same port group or nearby pins as the pins for a given channel.

- **(e.g.)** When RS-CANFD channel 0 is used:

CAN0TX	P0_0	P10_1
CAN0RX	P0_1	P10_0

Use one of the following pin combinations:

- P0_0 and P0_1, or
- P10_0 and P10_1.

The combinations of P0_0 and P10_0, and P0_1 and P10_1 are not allowed.

2B.3 Pin Functions During and After Reset

Table 2B.7 Pin Functions During and After Reset

Pins	During Reset	After Reset
JP0_0	High impedance	JP0_0: Input Serial programming mode: FPDR, FPDT (1 wire UART) FPDR (2 wire UART) Nexus I/F: DCUTDI input LPD (4 pins): LPDI input LPD (1 pin): LPDIO input/output
JP0_1	High impedance	JP0_1: Input Serial programming mode: FPDT Nexus I/F: DCUTDO output LPD (4 pins): LPDO output LPD (1 pin): High impedance
JP0_2	High impedance	JP0_2: Input Serial programming mode: FPCK Nexus I/F: DCUTCK input LPD (4 pins): LPDCLK input LPD (1 pin): High impedance
JP0_3	High impedance	JP0_3: Input Serial programming mode: High impedance Nexus I/F: DCUTMS input LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_4	Input[*3,*5]	JP0_4: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCUTRST}}$ input[*1] LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_5	High impedance	JP0_5: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCURDY}}$ output LPD (4 pins): LPDCLKOUT output LPD (1 pin): High impedance
JP0_6	High impedance	JP0_6: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{EVTO}}$ output LPD (4 pins): High impedance LPD (1 pin): High impedance
P8_6	Output[*2,*4]	Output (OPBT0.RESETOUTEN = 1)[*2] High impedance (OPBT0.RESETOUTEN = 0)[*2,*4]
P0 to P3, P8 to P13, P18 to P22 (except P8_6, P10_1, P10_2, P10_6 and P10_8)	High impedance	High impedance
P10_1	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE0 input (FLMD0 = 1, FLMD1 = 1)
P10_2	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE1 input (FLMD0 = 1, FLMD1 = 1)

Table 2B.7 Pin Functions During and After Reset

Pins	During Reset	After Reset
P10_6	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 1) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 0) MODE2 input (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 1)
P10_8	High impedance	High impedance (FLMD0 = 0) FLMD1 input (FLMD0 = 1)
FLMD0	Input	Input
$\overline{\text{RESET}}$	Input	Input
AP0, AP1	High impedance	High impedance

Note 1. When Nexus is enabled and no external device is connected, the level of the pin must always be fixed to low level.

Note 2. $\overline{\text{RESETOUT}}$ is output. For details, see **Section 2B.11, Port (Special I/O) Function Overview**.

Note 3. When the power is turned on or when $\overline{\text{RESET}}$ is low level, JP0_4 pin should be driven low level.

Note 4. If OPBT0.RESETOUTEN = 0, P8_6 pin status has a possibility to become unstable (less than 15 μs) at the transition moment to reset status by internal reset factors.

Note 5. When $\overline{\text{RESET}}$ is low level, on-chip pull-down resistor is connected to JP0_4.

2B.4 Port State in Standby Mode

For the port state in standby mode, see **Section 14.1.4, I/O Buffer Control**.

2B.5 Recommended Connection of Unused Pins

If the pins are not used, it is recommended to connect them as shown below.

Table 2B.8 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
A0VREF, A1VREF[*1]	**[Excluding 100-Pin LQFP]** Connected to EVCC or BVCC **[100-Pin LQFP]** Connected to EVCC
A0VSS, A1VSS[*1]	**[Excluding 100-Pin LQFP]** Connected to EVSS or BVSS **[100-Pin LQFP]** Connected to EVSS
$\overline{\text{RESET}}$	**[Excluding 100-Pin LQFP]** Connected to EVCC or BVCC via a resistor **[100-Pin LQFP]** Connected to EVCC via a resistor
XT1	Connected to REGVCC or AWOVSS via a resistor[*3] (bit 0 of IPIBC0 = 1) Connected to AWOVSS (bit 0 of IPIBC0 = 0)
X1	Connected to AWOVSS via a resistor
X2	Open
IP0_0	Connected to REGVCC or AWOVSS via a resistor[*3] (bit 0 of IPIBC0 = 1) Open (bit 0 of IPIBC0 = 0)
JP0 (excluding JP0_4) P0 P1 P2 P3 P8 (excluding P8_6) P9 P20	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P8_6	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS via a resistor
P10_1, P10_2, P10_6, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open

Table 2B.8 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
P10 (excluding P10_1, P10_2, P10_6, P10_8) P11 P12 P13 P18 P19 P21 P22	**[Excluding 100-Pin LQFP]** Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to BVCC or BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open **[100-Pin LQFP]** Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
AP1	Input: Open (when the PIBCn_m bit is 0) Connected to A1VREF or A1VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC via a resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Connected to EVCC via a resistor $\overline{\text{DCUTRST}}$ (JP0_4): Connected to EVSS via a resistor[*2] $\overline{\text{DCURDY}}$ /LPDCLKOUT (JP0_5):Open $\overline{\text{EVTO}}$ (JP0_6): Open[*1]

Note 1. Only available for 272-pin, 233-pin, 176-pin and 144-pin devices

Note 2. For in case when a debugging interface is used, this pin should be connected to EVCC through resistor depending on the development tool made by a third party.

Note 3. XT1 = IP0_0 (XT2) = REGVCC or AWOVSS should be set.
XT1 is connected to IP0_0 (XT2) through an internal resistor. Therefore, it is necessary to maintain equal voltage level in order not to make a current path.

2B.6 Features of RH850/F1KM Port

2B.6.1 Port Group

The RH850/F1KM provides the following port groups, indicated by the numbers in the table below.

Table 2B.9 Port Groups in RH850/F1KM-S4

No. of Pins	Port Group	RH850/F1KM-S4
100 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
144 pins	Number	13
	Name	P0, P1, P8 to P12, P18, P20, JP0, AP0, AP1, IP0
176 pins	Number	14
	Name	P0 to P2, P8 to P12, P18, P20, JP0, AP0, AP1, IP0
233 pins	Number	17
	Name	P0 to P3, P8 to P13, P18 to P20, JP0, AP0, AP1, IP0
272 pins	Number	19
	Name	P0 to P3, P8 to P13, P18 to P22, JP0, AP0, AP1, IP0

2B.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n”. For example, the port mode control register of the Pn pin is PMCn (n = 0 to 3, 8 to 13, 18 to 22).

2B.6.3 Register Base Addresses

Port and JTAG port base addresses are listed in the following table.

Port and JTAG port register addresses are given as offsets from the base addresses.

Table 2B.10 Register Base Addresses

Base Address Name	Base Address
<PORTn_base>	FFC1 0000_H
<JPORT0_base>	FFC2 0000_H

2B.6.4 Clock Supply

The clock supply to ports is shown in the following table.

Table 2B.11 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Port	Register access clock	CPUCLK_UL

2B.7 Port Functions

This product has various pins for input/output ports. The ports are organized in port groups.

The RH850/F1KM also has several control registers to enable pins to be used as other than general-purpose input/output pins.

For a description of the terms pin, port, and port group, see **Section 2B.7.2, Terms**

2B.7.1 Functional Overview

- All the port settings can be specified individually.
- The maximum number of bits (pins) in a port is 16.
- The output level of any pin can be set independently without affecting the other pins in the same port.
- Input buffers are enabled through registers settings.
- Pin level is read by dedicated port-pin-read register (PPR)
- All possible port functions are shown in the tables listed below.
  Table 2B.40, JTAG Port 0 (JP0), **Table 2B.42, Port 0 (P0)**, **Table 2B.44, Port 1 (P1)**, **Table 2B.46, Port 2 (P2)**, **Table 2B.48, Port 3 (P3)**, **Table 2B.50, Port 8 (P8)**, **Table 2B.52, Port 9 (P9)**, **Table 2B.54, Port 10 (P10)**, **Table 2B.56, Port 11 (P11)**, **Table 2B.58, Port 12 (P12)**, **Table 2B.60, Port 13 (P13)**, **Table 2B.62, Port 18 (P18)**, **Table 2B.64, Port 19 (P19)**, **Table 2B.66, Port 20 (P20)**, **Table 2B.68, Port 21 (P21)**, **Table 2B.70, Port 22 (P22)**, **Table 2B.72, Analog Port 0 (AP0)**, **Table 2B.74, Analog Port 1 (AP1)**, **Table 2B.76, Input Port 0 (IP0)**, and **Section 2B.9.2, Pin Function Configuration**.

CAUTION

Some input or output functions may be assigned to more than one port. Only activate a given function on a single pin. Do not activate a function on multiple pins at the same time. This also applies in cases where multiple peripheral functions are assigned to a single multiplexed function and only one of these functions is used.

[Example]

INTP0 is assigned to the following pins on this device. However, the INTP0 function should not be activated on more than one pin. After activating the function on one pin, do not activate it on another.

- JP0_0 (1st input alternative function)
- P0_1 (2nd, 3rd input alternative function)
- P10_0 (2nd input alternative function)

In the above case, when the 1st input alternative function (INTP0) of JP0_0 is selected, using the 2nd input alternative function (CAN0RX/INTP0) of P0_1 only for the CAN signal is also prohibited.

2B.7.2 Terms

The following terms are used in this section:

Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

Port mode and ports

A pin in port mode works as a general-purpose input/output pin. It is then called “port”.

The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions, such as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that two different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function of the pin at that time.

2B.7.2.1 JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging.

JTAG port group registers and bit names are prefixed by a “J”. For example, JP0 denotes JTAG port group 0, and JPM0.JPM0_m denotes the JPM0_m port mode bit of the JPM0 port mode register.

NOTE

In this section, the descriptions about all ports and their registers other than PFCAEn and PIPCn apply to the JTAG port unless otherwise specified.

2B.7.3 Overview of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCn_m bit = 0)
  A pin in port mode operates as a general-purpose input/output pin. The I/O mode is selected by setting the PMn.PMn_m bit.
- Software I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 0)
  In this mode, the pins operate as alternative functions. The I/O mode is selected by setting the PMn.PMn_m bit.
- Direct I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 1)
  In this mode, the pins operate as alternative functions. Unlike the software I/O control alternative mode, however, the I/O mode is directly controlled by the alternative function.

An overview of the register settings is given in the tables below.

Table 2B.12 Pin Function Configuration (Overview)

Mode	Bit			I/O
	PMCn_m	PMn_m	PIPCn_m	
Port mode	0	0	X	O
		1[*1]		I
Software I/O control alternative mode	1	0	0	O
		1	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

Note 1. The input buffer must be enabled (PIBCn_m bit = 1).

- Software I/O control alternative mode (PIPCn.PIPCn_m bit = 0)
  - Output (PMn_m bit = 0): Alternative output mode 1 to Alternative output mode 7
  - Input (PMn_m bit = 1): Alternative input mode 1 to Alternative input mode 7
- Direct I/O control alternative mode (PIPCn.PIPCn_m bit = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 7 and Alternative input mode 1 to Alternative input mode 7 is directly selected by the alternative function.

Table 2B.13 Alternative Mode Selection Overview (PMCn.PMCn_m Bit = 1)

	Register					
Mode	PIPC*1	PM*1	PFCAE	PFCE	PFC	I/O
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)		1				I
Alternative output mode 2 (ALT-OUT2)		0			1	O
Alternative input mode 2 (ALT-IN2)		1				I
Alternative output mode 3 (ALT-OUT3)		0		1	0	O
Alternative input mode 3 (ALT-IN3)		1				I
Alternative output mode 4 (ALT-OUT4)		0			1	O
Alternative input mode 4 (ALT-IN4)		1				I
Alternative output mode 5 (ALT-OUT5)		0	1	0	0	O
Alternative input mode 5 (ALT-IN5)		1				I
Alternative output mode 6 (ALT-OUT6)		0			1	O
Alternative input mode 6 (ALT-IN6)		1				I
Alternative output mode 7 (ALT-OUT7)		0		1	0	O
Alternative input mode 7 (ALT-IN7)		1				I
Other than the above	Setting prohibited					

Note 1. If PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

If a pin is in alternative mode (PMCn.PMCn_m bit = 1), one of up to seven alternative functions can be selected for that pin by using the PFCn, PFCEn, and PFCAEn registers.

2B.7.4 Pin Data Input/Output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.

2B.7.4.1 Output Data

In the port mode (PMCn.PMCn_m bit = 0), the value of the Pn.Pn_m bit is output from the Pn_m pin.

2B.7.4.2 Input Data

When the PPRn register is read, either the value of the Pn_m pin, the value of the Pn.Pn_m bit, or the value output by the alternative function is returned.

Which value is returned depends on the pin mode and setting of several control bits.

The different PPRn read modes are shown in the table below.

Table 2B.14 PPRn_m Read Values

PMC n_m	PM n_m	PIBC n_m	PIPC n_m	PODC n_m	Mode	PPRn_m Read Value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m bit
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m bit*1
				1	Port open-drain output	
1	1	X	0	X	Software I/O control alternative input	Pn_m pin
	0			0	Software I/O control alternative push-pull output	Output signal from the alternative function*1
				1	Software I/O control alternative open-drain output	
	X		1	0	Direct I/O control alternative input or push-pull output	I/O port in alternative mode: ● Input: Pn_m pin ● Output: Output signal from the alternative function*1
				1	Direct I/O control alternative input or open-drain output	

Note 1. When PBDCn_m = 1, the level of the Pn_m pin is returned by the PPRn_m bit.

The control registers in the above table have the following effects:

- PMCn.PMCn_m bit
  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).
- PMn.PMn_m bit
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) when the port mode (PMCn_m = 0) and software I/O control alternative mode (PMCn_m = 1, PIPCn_m = 0) have been selected.
- PIBCn.PIBCn_m bit
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). If the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit; otherwise the Pn_m pin level is returned.
- PIPCn.PIPCn_m bit
  This bit selects software I/O control alternative mode or direct I/O control alternative mode.
- PODCn.PODCn_m bit
  This bit selects push-pull output (PODCn_m = 0) or open-drain output (PODCn_m = 1).

- PBDCn.PBDCn_m bit
  In output mode, when this bit is set to 1, the pin enters the bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

CAUTION

When using Pn_m as an alternative output function (PMCn.PMCn_m bit = 1, PMn.PMn_m bit = 0), the level of the Pn_m pin can be read at the PPRn.PPRn_m bit by enabling bidirectional mode (PBDCn.PBDCn_m bit = 1).
Note, however, that the level of the Pn_m pin will be input to the alternative function that the Pn_m pin is being used as.

2B.7.4.3 Writing to the Pn Register

The data to be output via port Pn_m in port mode (PMCn.PMCn_m bit = 0) is held in port register Pn.

Pn data can be overwritten in two ways:

- By writing data directly to the Pn register.
  In this case, new data can be written directly to the Pn register.
- By performing an indirect bitwise operation (a "set", "reset", or "not" operation) on the Pn register.
  An indirect bitwise operation ("set", "reset", or "not") can be performed on the Pn register by using the following two registers:
  - Port Set/Reset register PSRn
    If the PSRn.PSRn (m + 16) bit = 1, the value of the Pn.Pn_m bit is determined by the value of the PSRn.PSRn_m bit.
    In other words, the Pn_m bit can be set or reset without writing directly to the Pn register.
  - Port NOT register PNOTn
    By setting PNOTn.PNOTn_m bit to 1, the Pn.Pn_m bit can be inverted without writing directly to the Pn register.

An indirect bitwise operation on the Pn register ("set", "reset", or "not") has no effect on the bits that do not need to be updated, allowing you to overwrite only the bit or bits that need to be overwritten.

2B.8 Schematic View of Port Control

The following figure is a schematic view of the port control functions.

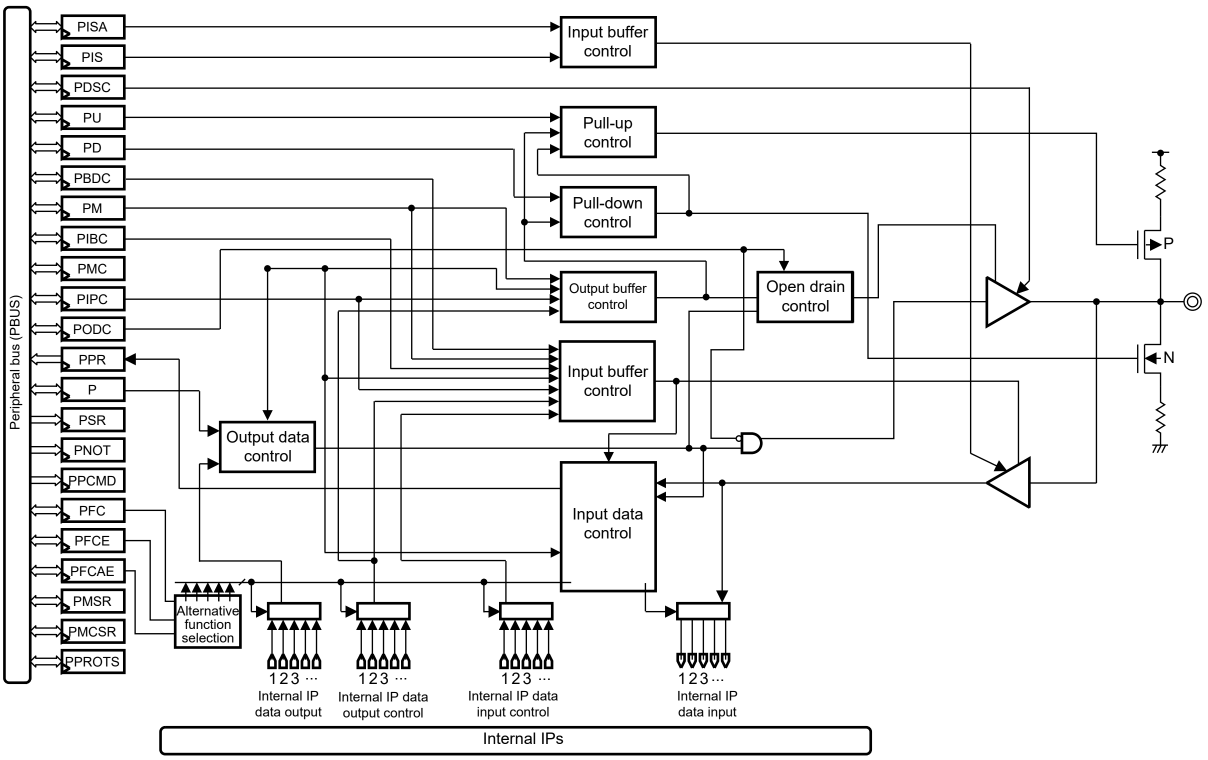


Figure 2B.6 Schematic View of Port Control

CAUTION

Use documented alternative functions only. The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.9 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then describes all registers in detail. The configuration registers are grouped as follows:

- **Section 2B.9.2, Pin Function Configuration**
- **Section 2B.9.3, Pin Data Input/Output**
- **Section 2B.9.4, Configuration of Electrical Characteristics**

2B.9.1 Overview

The following registers are used for setting the individual pins of the port groups.

For details on <PORTn_base> and <JPORT0_base>, see **Section 2B.6.3, Register Base Addresses**.

Table 2B.15 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin function configuration			
PORT	Port mode control register	PMCn	<PORTn_base> + 0400_H + n × 4
JTAG		JPMC0	<JPORT0_base> + 0040_H
PORT	Port mode control set/reset register	PMCSRn	<PORTn_base> + 0900_H + n × 4
JTAG		JPMCSR0	<JPORT0_base> + 0090_H
PORT	Port IP control register	PIPCn	<PORTn_base> + 4200_H + n × 4
PORT	Port mode register	PMn	<PORTn_base> + 0300_H + n × 4
		APMn	<PORTn_base> + $03C8_H$ + n × 4
JTAG		JPM0	<JPORT0_base> + 0030_H
PORT	Port mode set/reset register	PMSRn	<PORTn_base> + 0800_H + n × 4
		APMSRn	<PORTn_base> + $08C8_H$ + n × 4
JTAG		JPMSR0	<JPORT0_base> + 0080_H
PORT	Port input buffer control register	PIBCn	<PORTn_base> + 4000_H + n × 4
		APIBCn	<PORTn_base> + $40C8_H$ + n × 4
JTAG		JPIBC0	<JPORT0_base> + 0400_H
PORT		IPIBC0	<PORTn_base> + $40F0_H$
PORT	Port function control register	PFCn	<PORTn_base> + 0500_H + n × 4
JTAG		JPFC0	<JPORT0_base> + 0050_H
PORT	Port function control expansion register	PFCEn	<PORTn_base> + 0600_H + n × 4
JTAG		JPFCE0	<JPORT0_base> + 0060_H
PORT	Port function control additional expansion register	PFCAEn	<PORTn_base> + $0A00_H$ + n × 4
Pin data input/output			
PORT	Port bidirection control register	PBDCn	<PORTn_base> + 4100_H + n × 4
		APBDCn	<PORTn_base> + $41C8_H$ + n × 4
JTAG		JPBDC0	<JPORT0_base> + 0410_H
PORT	Port pin read register	PPRn	<PORTn_base> + 0200_H + n × 4
		APPRn	<PORTn_base> + $02C8_H$ + n × 4
JTAG		JPPR0	<JPORT0_base> + 0020_H
PORT		IPPR0	<PORTn_base> + $02F0_H$

Table 2B.15 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin data input/output			
PORT	Port register	Pn	<PORTn_base> + 0000$_H$ + n × 4
		APn	<PORTn_base> + 00C8$_H$ + n × 4
JTAG		JP0	<JPORT0_base> + 0000$_H$
PORT	Port NOT register	PNOTn	<PORTn_base> + 0700$_H$ + n × 4
		APNOTn	<PORTn_base> + 07C8$_H$ + n × 4
JTAG		JPNOT0	<JPORT0_base> + 0070$_H$
PORT	Port set/reset register	PSRn	<PORTn_base> + 0100$_H$ + n × 4
		APSRn	<PORTn_base> + 01C8$_H$ + n × 4
JTAG		JPSR0	<JPORT0_base> + 0010$_H$
Configuration of electrical characteristics			
PORT	Pull-up option register	PUn	<PORTn_base> + 4300$_H$ + n × 4
JTAG		JPU0	<JPORT0_base> + 0430$_H$
PORT	Pull-down option register	PDn	<PORTn_base> + 4400$_H$ + n × 4
JTAG		JPD0	<JPORT0_base> + 0440$_H$
PORT	Port drive strength control register	PDSCn	<PORTn_base> + 4600$_H$ + n × 4
JTAG		JPDSC0	<JPORT0_base> + 0460$_H$
PORT	Port open drain control register	PODCn	<PORTn_base> + 4500$_H$ + n × 4
JTAG		JPODC0	<JPORT0_base> + 0450$_H$
PORT	Port input buffer selection register	PISn	<PORTn_base> + 4700$_H$ + n × 4
JTAG		JPIS0	<JPORT0_base> + 0470$_H$
PORT	Port input buffer selection advanced register	PISAn	<PORTn_base> + 4A00$_H$ + n × 4
JTAG		JPISA0	<JPORT0_base> + 04A0$_H$
Port register protection			
PORT	Port protection command register	PPCMDn	<PORTn_base> + 4C00$_H$ + n × 4
JTAG		JPPCMD0	<JPORT0_base> + 04C0$_H$
PORT	Port protection status register	PPROTSn	<PORTn_base> + 4B00$_H$ + n × 4
JTAG		JPPROTS0	<JPORT0_base> + 04B0$_H$

Index n

In **Table 2B.15, Port Group Configuration Registers**, the index "n" in register symbols denotes the actual indices of the individual port groups. For example, PMCn generically indicates a port mode control register for port group n (Pn). The values for n differ according to the number of pins on the device in the way shown in **Table 2B.16, Number of Pins on the Device, Name of Port Groups, and Values for "n" in Register Symbols**.

Table 2B.16 Number of Pins on the Device, Name of Port Groups, and Values for "n" in Register Symbols

Number of Pins on the Device	Port Groups	Values for "n"
100 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
144 pins	P0, P1, P8, P9, P10, P11, P12, P18, P20	0, 1, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1
176 pins	P0, P1, P2, P8, P9, P10, P11, P12, P18, P20	0, 1, 2, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1
233 pins	P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20	0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20
	AP0, AP1	0, 1
272 pins	P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20, P21, P22	0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22
	AP0, AP1	0, 1

JTAG port registers

JTAG port registers are not explicitly described in the following register descriptions.

All descriptions (except for those of the PFCAEn register and PIPCn register) apply to JTAG port registers. Note, however, that the JTAG port register base address differs from that of regular ports.

Value after reset

The values after reset depend on the ports. For the values after reset, see the register descriptions in the following pages.

2B.9.2 Pin Function Configuration

2B.9.2.1 PMCn / JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access:	PMCn: This register can be read or written in 16-bit units. JPMC0: This register can be read or written in 8-bit units.
Address:	PMCn: <PORTn_base> + 0400_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20) JPMC0: <JPORT0_base> + 0040_H*1
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC n_15	PMC n_14	PMC n_13	PMC n_12	PMC n_11	PMC n_10	PMC n_9	PMC n_8	PMC n_7	PMC n_6	PMC n_5	PMC n_4	PMC n_3	PMC n_2	PMC n_1	PMC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.17 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

CAUTIONS

1. I/O is not controlled by only setting alternative mode (PMCn.PMCn_m bit = 1). If the alternative function requires direct I/O control, also set the PIPCn.PIPCn_m bit to 1.
2. If a port is to be used as an input pin in alternative mode, the signals from some pins will pass through a noise filter. These pins may require the setting of the FCLA0CTLm_<*name*>, DNFA<*name*>CTL and the DNFA<*name*>EN register. For details, see **Section 2B.12, Noise Filter & Edge/Level Detector**, and **Section 2B.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].

2B.9.2.2 PMCSRn / JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value of PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access: PMCSRn: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of register PMCn.

JPMCSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JPMC0.

Address: PMCSRn: <PORTn_base> + 0900_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20)

JPMCSR0: <JPORT0_base> + 0090_H^{*1}

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMC SRn_31	PMC SRn_30	PMC SRn_29	PMC SRn_28	PMC SRn_27	PMC SRn_26	PMC SRn_25	PMC SRn_24	PMC SRn_23	PMC SRn_22	PMC SRn_21	PMC SRn_20	PMC SRn_19	PMC SRn_18	PMC SRn_17	PMC SRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC SRn_15	PMC SRn_14	PMC SRn_13	PMC SRn_12	PMC SRn_11	PMC SRn_10	PMC SRn_9	PMC SRn_8	PMC SRn_7	PMC SRn_6	PMC SRn_5	PMC SRn_4	PMC SRn_3	PMC SRn_2	PMC SRn_1	PMC SRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.18 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMCSRn_m (PMCSRn_[15:0]) is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m. 1: PMCn_m is PMCSRn_m. **Example:** If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_[15:0]	Data bits that specify the value of PMCn_m if PMCSRn_m of the corresponding upper bit (PMCSRn_[31:16]) is 1. 0: PMCn_m is 0. 1: PMCn_m is 1.

NOTE

The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].

2B.9.2.3 PIPCn — Port IP Control Register

This register specifies whether the I/O direction of the Pn_m pin is controlled by the port mode register PMn.PMn_m or by an alternative function.

If the Pn_m pin is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn_m setting.

Regarding the alternative functions for which the PIPC register must be set, see **Section 2B.11, Port (Special I/O) Function Overview**.

Access: This register can be read or written in 16-bit units.

Address: PIPCn: <PORTn_base> + 4200_H + n × 4 (n = 0, 10, 11, 12, 20)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.43, Control Registers (P0)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11),** and **Table 2B.59, Control Registers (P12)**.

Table 2B.19 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

2B.9.2.4 PMn / APMn / JPM0 — Port Mode Register

This register specifies whether the individual pins of the port group n are in input mode or in output mode.

Access: PMn, APMn: These registers can be read or written in 16-bit units.
JPM0: This register can be read or written in 8-bit units.

Address: PMn: <PORTn_base> + 0300_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)
APMn: <PORTn_base> + $03C8_H$ + n × 4 (n = 0, 1)
JPM0: <JPORT0_base> + $0030_H{}^{*1}$

Value after reset: $FFFF_H{}^{*2}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	1	1	1	1	1	1	1	1	1	1^{*3}	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Note 2. The PM8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8 register is $FFBF_H$.
When the OPBT0.RESETOUTEN = 0, the PM8 register is $FFFF_H$.

Note 3. The PM8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PM8_6 bit is 1.

Table 2B.20 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

NOTES

1. To use a port in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).
2. By default, PMn.PMn_m specifies the I/O direction in port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m = 1), since PIPCn.PIPCn_m = 0 after reset.
3. The control bits of the analog port register (APMn) are APMn_[15:0].
4. The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].

2B.9.2.5 PMSRn / APMSRn / JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write data to the PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn_m is set by the corresponding bit in the lower 16 bits of PMSRn.

Access: PMSRn, APMSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_H$.

Reading bits 15 to 0 returns the value of registers PMn and APMn.

JPMSR0: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_H$. Bits 15 to 8 are read as FF$_H$. Reading bits 7 to 0 returns the value of register JPM0.

Address: PMSRn: <PORTn_base> + 0800$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

APMSRn: <PORTn_base> + 08C8$_H$ + n × 4 (n = 0, 1)

JPMSR0: <JPORT0_base> + 0080$_H$*1

Value after reset: 0000 FFFF$_H$*2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
Value after reset	1	1	1	1	1	1	1	1	1	1*3	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Note 2. The PMSR8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8 register is 0000 FFBF$_H$.
When the OPBT0.RESETOUTEN = 0, the PMSR8 register is 0000 FFFF$_H$.

Note 3. The PMSR8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PMSR8_6 bit is 1.

Table 2B.21 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMSRn_m (PMSRn_[15:0]) is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m. **Example:** If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Data bits that specify the value of PMn_m if PMSRn_m of the corresponding upper bit (PMSRn_[31:16]) is 1. 0: PMn_m is 0. 1: PMn_m is 1.

NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[31:0].
2. The control bits of the analog port mode set/reset register (APMSRn) are APMSRn_[31:0].

2B.9.2.6 PIBCn / APIBCn / JPIBC0 / IPIBC0 — Port Input Buffer Control Register

In input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables the port pin’s input buffer.

Access: PIBCn, APIBCn, IPIBC0: These registers can be read or written in 16-bit units.

JPIBC0: This register can be read or written in 8-bit units.

Address: PIBCn: <PORTn_base> + 4000$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

APIBCn: <PORTn_base> + 40C8$_H$+ n × 4 (n = 0, 1)

JPIBC0: <JPORT0_base> + 0400$_H$

IPIBC0: <PORTn_base> + 40F0$_H$*1

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBC n_15	PIBC n_14	PIBC n_13	PIBC n_12	PIBC n_11	PIBC n_10	PIBC n_9	PIBC n_8	PIBC n_7	PIBC n_6	PIBC n_5	PIBC n_4	PIBC n_3	PIBC n_2	PIBC n_1	PIBC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, **Table 2B.75, Control Registers (AP1)** and **Table 2B.77, Control Registers (IP0)**.

Table 2B.22 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled

NOTES

1. When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].

CAUTION

Settings in this register are overruled in bidirectional mode (PBDCn.PBDCn_m = 1).

2B.9.2.7 PFCn / JPFC0 — Port Function Control Register

This register, together with register PFCEn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is selected by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCn: This register can be read or written in 16-bit units.
JPFC0: This register can be read or written in 8-bit units.

Address: PFCn: <PORTn_base> + 0500_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 13, 18, 20)
JPFC0: <JPORT0_base> + 0050_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFC n_15	PFC n_14	PFC n_13	PFC n_12	PFC n_11	PFC n_10	PFC n_9	PFC n_8	PFC n_7	PFC n_6	PFC n_5	PFC n_4	PFC n_3	PFC n_2	PFC n_1	PFC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.23 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2B.26, Setting Alternative Functions**.

NOTE

The control bits of the JTAG port function control register (JPFC0) are JPFC0_[7:0].

2B.9.2.8 PFCEn / JPFCE0 — Port Function Control Expansion Register

This register, together with register PFCn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCEn: This register can be read or written in 16-bit units.
JPFCE0: This register can be read or written in 8-bit units.

Address: PFCEn: <PORTn_base> + 0600_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
JPFCE0: <JPORT0_base> + 0060_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn_11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.63, Control Registers (P18)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.24 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2B.26, Setting Alternative Functions**.

2B.9.2.9 PFCAEn — Port Function Control Additional Expansion Register

This register selects the alternative peripheral functions together with PFCEn, PFCn registers.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCAEn: This register can be read or written in 16-bit units.

Address: PFCAEn: <PORTn_base> + $0A00_H$ + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 20)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAEn_15	PFCAEn_14	PFCAEn_13	PFCAEn_12	PFCAEn_11	PFCAEn_10	PFCAEn_9	PFCAEn_8	PFCAEn_7	PFCAEn_6	PFCAEn_5	PFCAEn_4	PFCAEn_3	PFCAEn_2	PFCAEn_1	PFCAEn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.25 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2B.26, Setting Alternative Functions**.

Table 2B.26 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1
			0	Alternative output mode 1
		1	1	Alternative input mode 2
			0	Alternative output mode 2
	1	0	1	Alternative input mode 3
			0	Alternative output mode 3
		1	1	Alternative input mode 4
			0	Alternative output mode 4
1	0	0	1	Alternative input mode 5
			0	Alternative output mode 5
		1	1	Alternative input mode 6
			0	Alternative output mode 6
	1	0	1	Alternative input mode 7
			0	Alternative output mode 7
		1	X	Setting prohibited

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m, or PFCAEn_m bit, set the PMCn_m bit to “1”.
- With this product, the I/O of some functions is assigned to two or more pins, but a specific pin function can only be set to one pin at a time. Setting the same pin function to two or more pins at the same time is prohibited. For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as a pin function other than b.

NOTE

For more details on the assignment of each function, see **Section 2B.10.1.2, Control Registers** to **Section 2B.10.15.2, Control Registers**.

2B.9.3 Pin Data Input/Output

2B.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

This register enables the input buffer in output mode and sets the port to bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

Access: PBDCn, APBDCn: These registers can be read or written in 16-bit units.
JPBDC0: This register can be read or written in 8-bit units.

Address: PBDCn: <PORTn_base> + 4100_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)
APBDCn: <PORTn_base> + $41C8_H$ + n × 4 (n = 0, 1)
JPBDC0: <JPORT0_base> + 0410_H[*1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDC n_15	PBDC n_14	PBDC n_13	PBDC n_12	PBDC n_11	PBDC n_10	PBDC n_9	PBDC n_8	PBDC n_7	PBDC n_6	PBDC n_5	PBDC n_4	PBDC n_3	PBDC n_2	PBDC n_1	PBDC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Table 2B.27 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin. 0: Bidirectional mode disabled 1: Bidirectional mode enabled

CAUTION

- When the Pn_m port is used for the alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read from PPRn.PPRn_m by enabling the bidirectional mode (PBDCn.PBDCn_m = 1).
- However, output of that alternative output function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn_m, PFCEn.PFCEn_m, and PFCAEn.PFCAEn_m). If the alternative input function in question is being used by another pin, the alternative input function is not guaranteed.

NOTE

The control bits of the JTAG port bidirection control register (JPBDC0) are JPBDC0_[7:0].

2B.9.3.2 PPRn / APPRn / JPPR0 / IPPR0 — Port Pin Read Register

This register reflects the actual level of the Pn_m pin, whether it is the value of the Pn.Pn_m bit or the level of an alternative output function.

Access:	PPRn, APPRn, IPPR0: These registers are read-only registers that can be read in 16-bit units. JPPR0: This register is a read-only register that can be read in 8-bit units.
Address:	PPRn: <PORTn_base> + 0200_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22) APPRn: <PORTn_base> + $02C8_H$ + n × 4 (n = 0, 1) JPPR0: <JPORT0_base> + 0020_H IPPR0: <PORTn_base> + $02F0_H$*1
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPR n_15	PPR n_14	PPR n_13	PPR n_12	PPR n_11	PPR n_10	PPR n_9	PPR n_8	PPR n_7	PPR n_6	PPR n_5	PPR n_4	PPR n_3	PPR n_2	PPR n_1	PPR n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, **Table 2B.75, Control Registers (AP1)** and **Table 2B.77, Control Registers (IP0)**.

Table 2B.28 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	The Pn_m Pin, Pn.Pn_m value or alternative function output.

NOTES

1. For the read values of the PPRn register, see **Section 2B.7.4, Pin Data Input/Output**.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].

2B.9.3.3 Pn / APn / JP0 — Port Register

This register holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access: Pn, APn: These registers can be read or written in 16-bit units.
JP0: This register can be read or written in 8-bit units.

Address: Pn: <PORTn_base> + 0000_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)
APn: <PORTn_base> + $00C8_H$ + n × 4 (n = 0, 1)
JP0: <JPORT0_base> + 0000_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Table 2B.29 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

NOTE

The control bits of the JTAG port register (JP0) are JP0_[7:0].

2B.9.3.4 PNOTn / APNOTn / JPNOT0 — Port NOT Register

This register allows the Pn_m bit of the port register Pn to be inverted without directly writing to Pn.

Access: PNOTn, APNOTn: These registers are write-only registers that can be written in 16-bit units. When read, 0000_H is returned.

JPNOT0: This register is a write-only register that can be written in 8-bit units. When read, 00_H is returned.

Address: PNOTn: <PORTn_base> + 0700_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

APNOTn: <PORTn_base> + $07C8_H$ + n × 4 (n = 0, 1)

JPNOT0: <JPORT0_base> + 0070_H[*1]

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOT n_15	PNOT n_14	PNOT n_13	PNOT n_12	PNOT n_11	PNOT n_10	PNOT n_9	PNOT n_8	PNOT n_7	PNOT n_6	PNOT n_5	PNOT n_4	PNOT n_3	PNOT n_2	PNOT n_1	PNOT n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Table 2B.30 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted. 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted ($\overline{\text{Pn\_m}}$ → Pn_m)

NOTE

The control bits of the JTAG port NOT register are JPNOT0_[7:0].

2B.9.3.5 PSRn / APSRn / JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to the Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn_m is set by the corresponding bit in the lower 16 bits of PSRn.

Access: PSRn, APSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_H$. Reading bits 15 to 0 returns the value of registers Pn and APn.

JPSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000$_H$. Reading bits 7 to 0 returns the value of register JP0.

Address: PSRn: <PORTn_base> + 0100$_H$ + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

APSRn: <PORTn_base> + 01C8$_H$ + n × 4 (n = 0, 1)

JPSR0: <JPORT0_base> + 0010$_H^{*1}$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, **Table 2B.71, Control Registers (P22)**, **Table 2B.73, Control Registers (AP0)**, and **Table 2B.75, Control Registers (AP1)**.

Table 2B.31 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m (PSRn_[15:0]) is written to Pn_m. 0: Pn_m is not affected by PSRn_m 1: Pn_m is PSRn_m **Example:** If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit (PSRn_[31:16]) PSRn_m is 1. 0: Pn_m = 0 1: Pn_m = 1

NOTE

The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].

2B.9.4 Configuration of Electrical Characteristics

2B.9.4.1 PUn / JPU0 — Pull-Up Option Register

This register specifies whether an internal pull-up resistor is connected to an input pin.

Access: PUn: This register can be read or written in 16-bit units.

JPU0: This register can be read or written in 8-bit units.

Address: PUn: <PORTn_base> + 4300_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11,12, 13, 18, 19, 20, 21, 22)

JPU0: <JPORT0_base> + 0430_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20)**, **Table 2B.69, Control Registers (P21)**, and **Table 2B.71, Control Registers (P22)**.

Table 2B.32 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an internal pull-up resistor is connected to the corresponding pin. 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The pull-up resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].

2B.9.4.2 PDn / JPD0 — Pull-Down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin.

Access: PDn: This register can be read or written in 16-bit units.
JPD0: This register can be read or written in 8-bit units.

Address: PDn: <PORTn_base> + 4400_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)
JPD0: <JPORT0_base> + 0440_H^{*1}

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20), Table 2B.69, Control Registers (P21)**, and **Table 2B.71, Control Registers (P22).**

Table 2B.33 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin. 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].

2B.9.4.3 PDSCn / JPDSC0 — Port Drive Strength Control Register

This register specifies the output driver strength of the port pin. This function selects the fast mode (high drive strength) or slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**. Regarding the alternative functions for which the PDSC register needs to be set, see **Section 2B.11.3.3, Output Buffer Control (PDSC)**.

Access:	PDSCn, JPDSC0: These registers can be read or written in 32-bit units.
Address:	PDSCn: <PORTn_base> + 4600_H + n × 4 (n = 0, 1, 2, 3, 10, 11, 12, 13, 18, 19, 20) JPDSC0: <JPORT0_base> + 0460_H[*1]
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSC n_15	PDSC n_14	PDSC n_13	PDSC n_12	PDSC n_11	PDSC n_10	PDSC n_9	PDSC n_8	PDSC n_7	PDSC n_6	PDSC n_5	PDSC n_4	PDSC n_3	PDSC n_2	PDSC n_1	PDSC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, and **Table 2B.67, Control Registers (P20)**.

Table 2B.34 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Lower drive strength (when the frequency output from the pin is 10 MHz or below) 1: High drive strength (when the frequency output from the pin is 40 MHz or less).

2B.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: PODCn, JPODC0: These registers can be read or written in 32-bit units.

Address: PODCn: <PORTn_base> + 4500_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

JPODC0: <JPORT0_base> + 0450_H*1

Value after reset: 0000 0000_H*2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0*3	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20), Table 2B.69, Control Registers (P21)**, and **Table 2B.71, Control Registers (P22)**.

Note 2. The PODC8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8 register is 0000 0040_H.
When the OPBT0.RESETOUTEN = 0, the PODC8 register is 0000 0000_H.

Note 3. The PODC8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8_6 bit is 1.
When the OPBT0.RESETOUTEN = 0, the PODC8_6 bit is 0.

Table 2B.35 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

NOTE

The control bits of the JTAG port open drain control register (JPODC0) are JPODC0_[31:0].

2B.9.4.5 PISn/JPIS0 — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

Access: PISn: This register can be read or written in 16-bit units.

JPIS0: This register can be read or written in 8-bit units.

Address: PISn: <PORTn_base> + 4700_H + n × 4 (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22)

JPIS0: <JPORT0_base> + $0470_H{}^{*1}$

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIS n_15	PIS n_14	PIS n_13	PIS n_12	PIS n_11	PIS n_10	PIS n_9	PIS n_8	PIS n_7	PIS n_6	PIS n_5	PIS n_4	PIS n_3	PIS n_2	PIS n_1	PIS n_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.43, Control Registers (P0)**, **Table 2B.45, Control Registers (P1)**, **Table 2B.47, Control Registers (P2)**, **Table 2B.49, Control Registers (P3)**, **Table 2B.51, Control Registers (P8)**, **Table 2B.53, Control Registers (P9)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, **Table 2B.63, Control Registers (P18)**, **Table 2B.65, Control Registers (P19)**, **Table 2B.67, Control Registers (P20), Table 2B.69, Control Registers (P21)**, and **Table 2B.71, Control Registers (P22)**.

Table 2B.36 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer Characteristics: 0: Type 1 (SHMT1) 1: Type 2 (SHMT4)

NOTES

1. Details of the definition of type 1 and type 2 are given in **Section 2B.11.3.2, Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)**. For details, also see **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for input buffer characteristics.
2. The control bits of the JTAG port input buffer selection register (JPIS0) are JPIS0_[7:0].

2B.9.4.6 PISAn / JPISA0 — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics.

Access: PISAn: This register can be read or written in 16-bit units.

JPISA0: This register can be read or written in 8-bit units.

Address: PISAn: <PORTn_base> + $4A00_H$ + n × 4 (n = 10, 11, 12, 13, 18)

JPISA0: <JPORT0_base> + $04A0_H$^{*1}

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISAn_15	PISAn_14	PISAn_13	PISAn_12	PISAn_11	PISAn_10	PISAn_9	PISAn_8	PISAn_7	PISAn_6	PISAn_5	PISAn_4	PISAn_3	PISAn_2	PISAn_1	PISAn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2B.10, Port (General I/O) Function Overview**: **Table 2B.41, Control Registers (JP0)**, **Table 2B.55, Control Registers (P10)**, **Table 2B.57, Control Registers (P11)**, **Table 2B.59, Control Registers (P12)**, **Table 2B.61, Control Registers (P13)**, and **Table 2B.63, Control Registers (P18)**.

Table 2B.37 PISAn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISA_[15:0]	Specifies the input buffer characteristics: 0: Type 2 (SHMT4) 1: Type 5 (TTL)

Table 2B.38 Port Input Selection Advanced Register Contents

PISAn	PISn	Function
0	0	Type 1 input buffer is selected (SHMT1)
	1	Type 2 input buffer is selected (SHMT4)
1	X	Type 5 input buffer is selected (TTL)

NOTE

Details of the definition of type 2 and type 5 are given in **Section 2B.11.3.2, Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)**. For details, also see **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for input buffer characteristics.

2B.9.5 Port Register Protection

RH850/F1KM has Port Protection Command Registers (PPCMDn) and Port Protection Status Registers (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 5, Write-Protected Registers**.

2B.9.6 Flowchart Examples for Port Settings

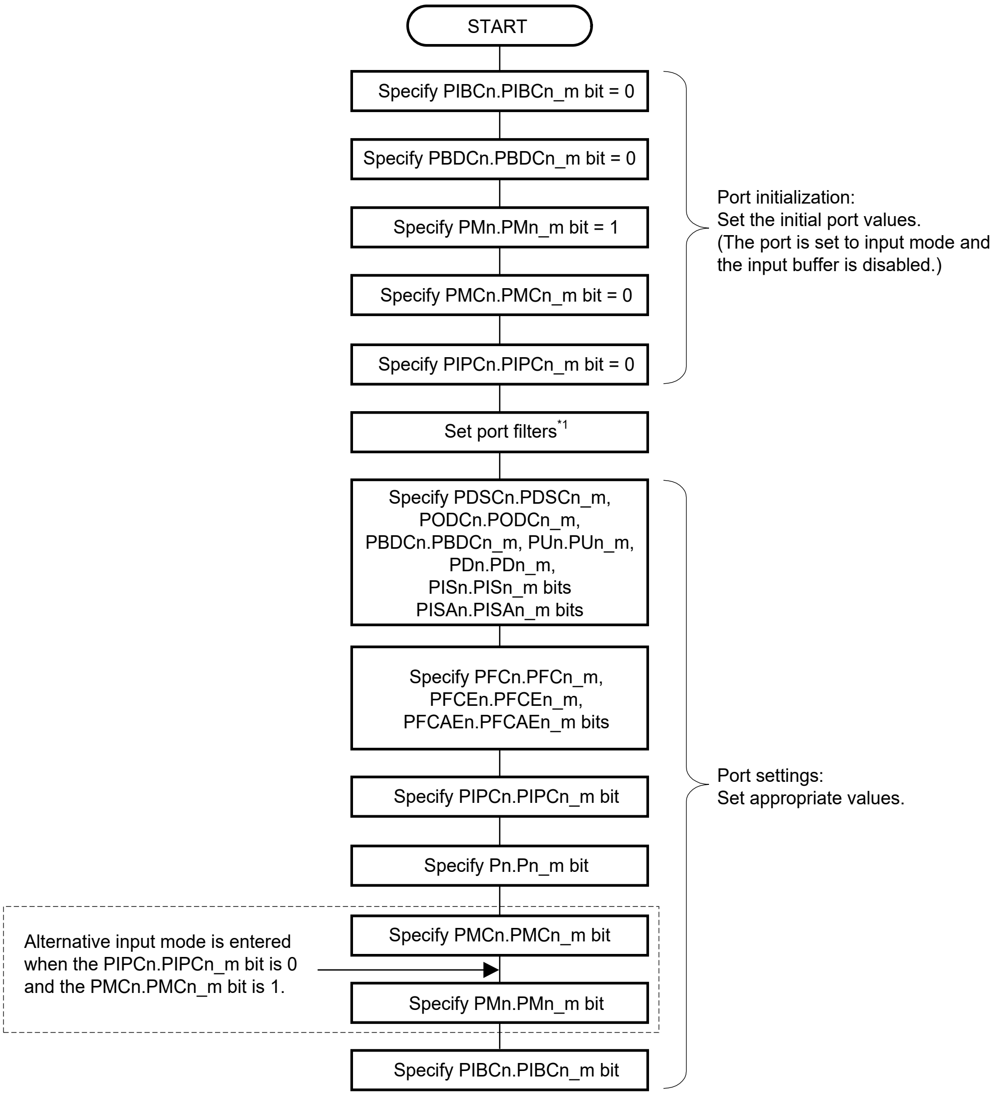
Examples of the port settings are shown in the flowchart below.

CAUTION

If the port is set to the PIPCn.PIPCn_m = 0 and alternative output mode, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCn_m bit is set to 1 and when the PMn.PMn_m bit is set to 0. If an interrupt-related signal is specified as an alternate function of the port, the mode will temporarily become the alternative input mode, so either disable the interrupt in question, or specify that the interrupt is ignored.

2B.9.6.1 Batch Setting

An example of specifying batch port settings is shown in the flowchart below.



Note 1. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-6 and INTP9-23 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2B.7 Example of Port Settings (When Specified in Batch)

2B.9.6.2 Individual Settings

An example of specifying individual port settings is shown in the flowchart below.

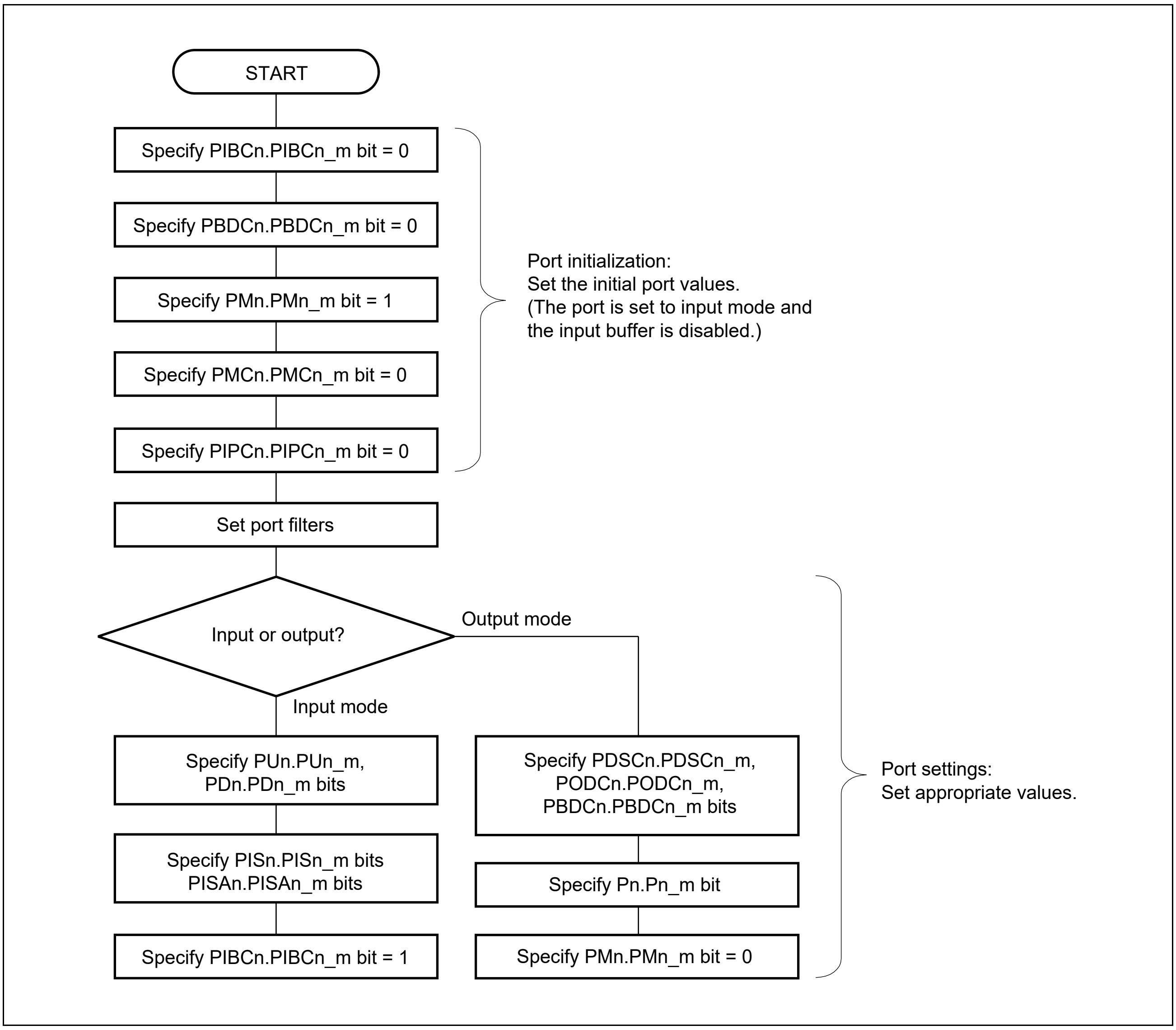


Figure 2B.8 Example of Port Settings (in Port Mode)

(1) With IP Control

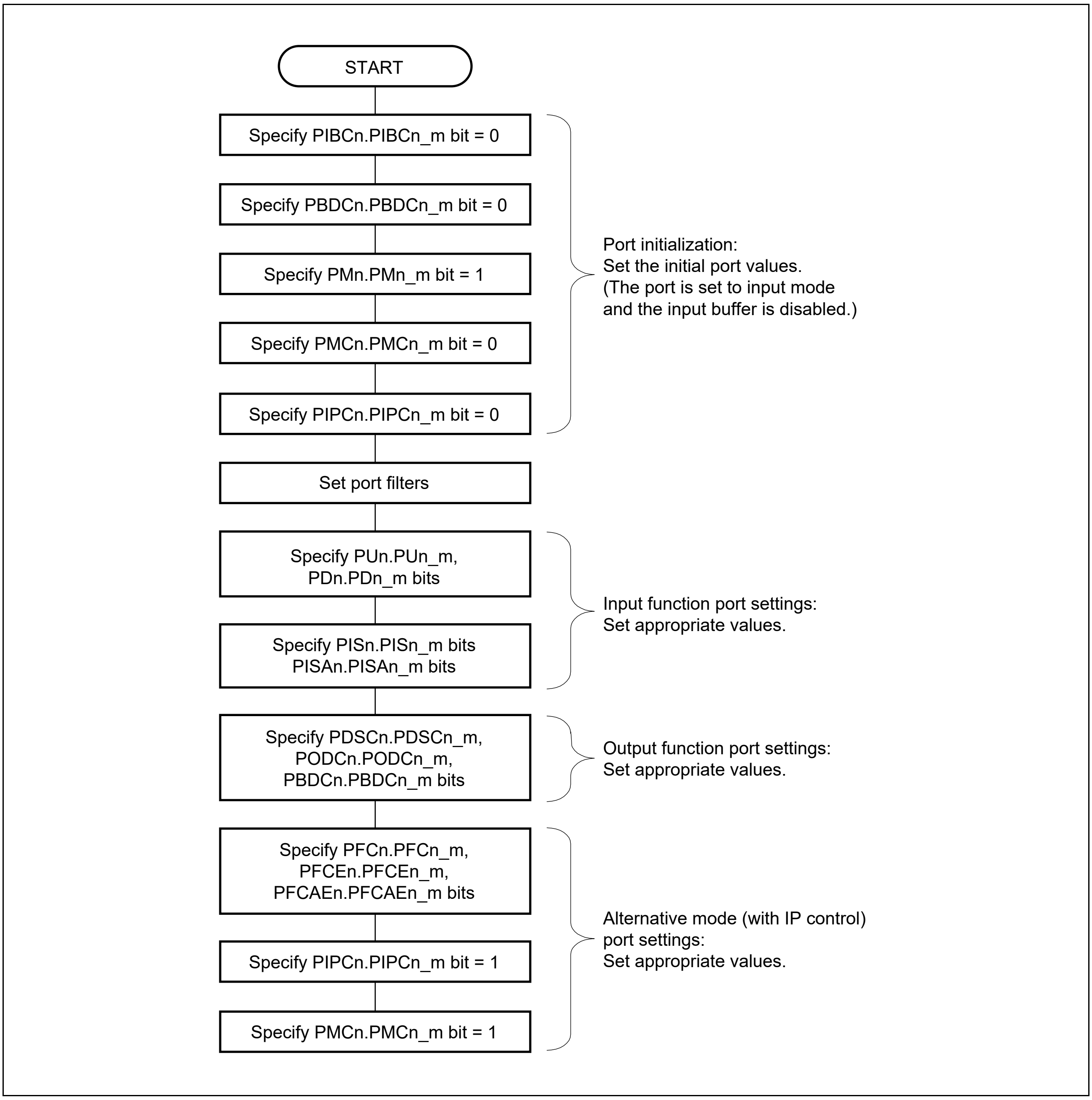
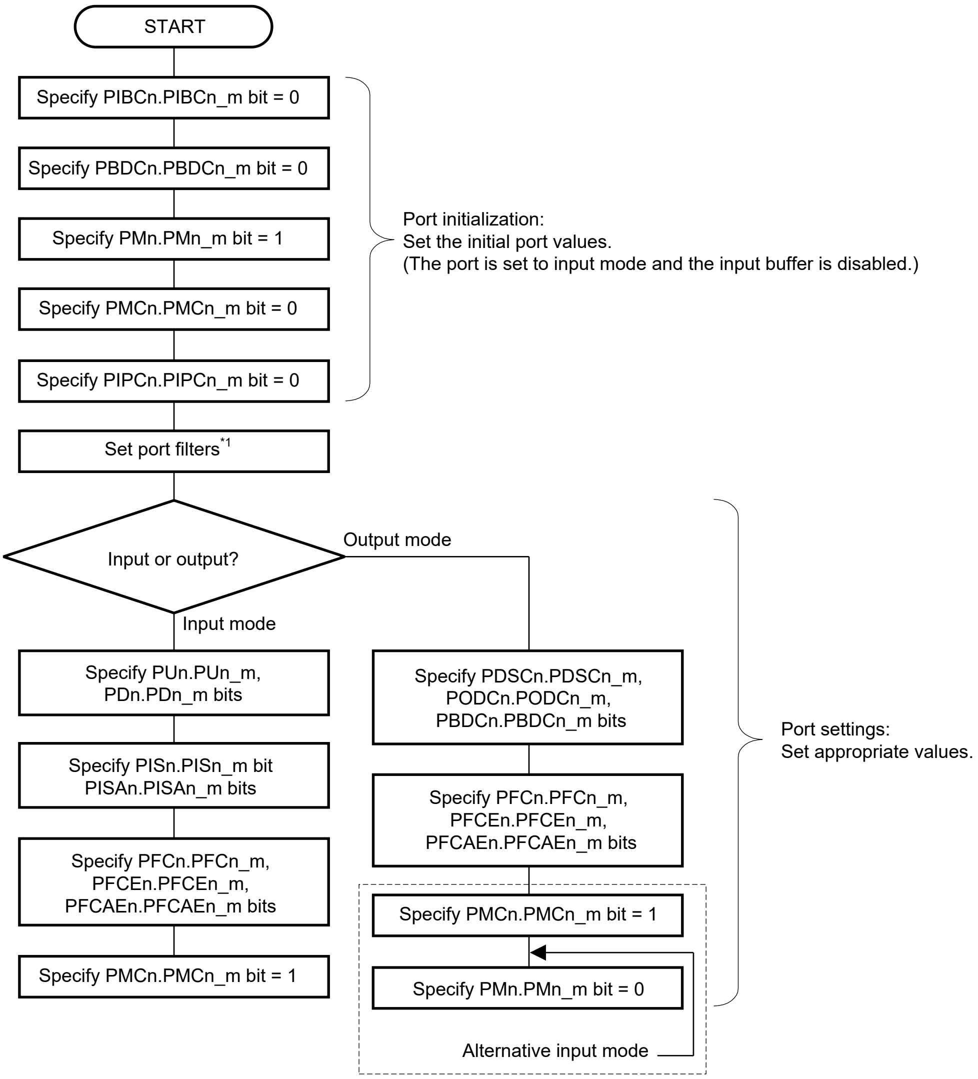


Figure 2B.9 Example of Port Settings (in Alternative Mode)

(2) Without IP Control



Note 1. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-6 and INTP9-23 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2B.10 Example of Port Settings (in Alternative Mode)

2B.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functions and all the functions assigned to the ports. See the following pages for details.

In addition, whether the port mode is alternative mode or not can be selected by PMCn register setting. When PMCn.PMCn_m = 1, alternative functions are selected by the PFCn, PFCEn, and PFCAEn registers.

Table 2B.39 Port Function

Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	Device				
						100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
JTAG Port 0	JP0_0 - 5	6 bits	In/Out	AWO	JTAG, LPD	✓	—	—	—	—
	JP0_0 - 6	7 bits				—	✓	✓	✓	✓
Port 0	P0_0 - 14	15 bits	In/Out	AWO		✓	✓	✓	✓	✓
Port 1	P1_0 - 5, 8-11	10 bits	In/Out	AWO		—	✓	—	—	—
	P1_0 - 5, 8-15	14 bits				—	—	✓	✓	✓
Port 2	P2_0 - 6	7 bits	In/Out	AWO		—	—	✓	—	—
	P2_0 - 15	16 bits				—	—	—	✓	✓
Port 3	P3_0	1 bit	In/Out	AWO		—	—	—	✓	—
	P3_0 - 10	11 bits				—	—	—	—	✓
Port 8	P8_2 - 12	11 bits	In/Out	AWO	ADCA0 (10-bit resolution)	✓	—	—	—	—
	P8_0 - 12	13 bits			$\overline{\text{RESETOUT}}$	—	✓	✓	✓	✓
Port 9	P9_0 - 4	5 bits	In/Out	ISO	ADCA0 (10-bit resolution)	✓	✓	✓	✓	✓
Port 10	P10_0 - 14	15 bits	In/Out	ISO		✓	—	—	—	—
	P10_0 - 15	16 bits				—	✓	✓	✓	✓
Port 11	P11_1 - 7	7 bits	In/Out	ISO		✓	—	—	—	—
	P11_0 - 12, 15	14 bits				—	✓	✓	✓	✓
Port 12	P12_0 - 2	3 bits	In/Out	ISO		—	✓	—	—	—
	P12_0 - 5	6 bits				—	—	✓	✓	✓
Port 13	P13_0 - 7	8 bits	In/Out	ISO		—	—	—	✓	✓
Port 18	P18_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	✓	—	—	—
	P18_0 - 7	8 bits				—	—	✓	—	—
	P18_0 - 15	16 bits				—	—	—	✓	✓
Port 19	P19_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	—	—	✓	✓
Port 20	P20_4 - 5	2 bits	In/Out	ISO		—	✓	—	—	—
	P20_0 - 5	6 bits				—	—	✓	✓	—
	P20_0 - 14	15 bits				—	—	—	—	✓
Port 21	P21_0 - 4	5 bits	In/Out	ISO		—	—	—	—	✓
Port 22	P22_0 - 15	16 bits	In/Out	ISO		—	—	—	—	✓
Analog Port 0	AP0_0 - 15	16 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	✓	✓	✓	✓	✓
Analog Port 1	AP1_0 - 7	8 bits	In/Out	ISO	ADCA1 (12/10-bit resolution)	—	✓	—	—	—
	AP1_0 - 15	16 bits			ADCA1 (12/10-bit resolution)	—	—	✓	✓	✓
Input Port 0	IP0_0	1 bit	In	AWO	SOSC (XT2 pin)	—	✓	✓	✓	✓

2B.10.1 JTAG Port 0 (JP0)

2B.10.1.1 Alternative Function

The following alternative functions are available when JTAG port 0 is configured as a general-purpose I/O port by setting OPJTAG[1:0] on the corresponding option byte to 00_B.

Table 2B.40 JTAG Port 0 (JP0)

Port Mode (JPMC0_m = 0)	Alternative Mode (JPMC0_m = 1)										ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
JP0_0*[1]	INTP0		TAUJ2I0	TAUJ2O0	FPDR	FPDT						DCUTDI/LPDI/LPDIO	28	39	47	U4	V5
JP0_1	INTP1		TAUJ0I0	TAUJ0O0		FPDT						DCUTDO/LPDO	27	38	46	P5	W4
JP0_2	INTP2		TAUJ0I1	TAUJ0O1	FPCK							DCUTCK/LPDCLK	26	37	45	T4	U5
JP0_3	INTP3	CSCXFOUT	TAUJ0I2	TAUJ0O2								DCUTMS	25	36	44	R4	V4
JP0_4												$\overline{\text{DCUTRST}}$	24	35	43	R3	Y3
JP0_5	NMI	RTCA0OUT	TAUJ0I3	TAUJ0O3								$\overline{\text{DCURDY}}$ /LPDCLKOUT	23	34	42	U3	W3
JP0_6												$\overline{\text{EVTO}}$	—	54	66	U11	W11

Note 1. In LPD (1 pin) mode, the JP0_0 output buffer state is Open-drain.

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2B.10.1.2 Control Registers

Table 2B.41 Control Registers (JP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
JP0	JTAG port register 0	8	5-0	R/W	0000$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPSR0	JTAG port set/reset register 0	32	21-16, 5-0	R/W	0010$_H$	0000 0000$_H$	✓	—	—	—	—
			22-16, 6-0				—	✓	✓	✓	✓
JPPR0	JTAG port pin read register 0	8	5-0	R	0020$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPM0	JTAG port mode register 0	8	5-0	R/W	0030$_H$	FF$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPMC0	JTAG port mode control register 0	8	5, 3-0	R/W	0040$_H$	00$_H$	✓	✓	✓	✓	✓
JPFC0	JTAG port function control register 0	8	5, 3-0	R/W	0050$_H$	00$_H$	✓	✓	✓	✓	✓
JPFCE0	JTAG port function control expansion register 0	8	2-0	R/W	0060$_H$	00$_H$	✓	✓	✓	✓	✓
JPNOT0	JTAG port NOT register 0	8	5-0	W	0070$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPMSR0	JTAG port mode set/reset register 0	32	21-16, 5-0	R/W	0080$_H$	0000 FFFF$_H$	✓	—	—	—	—
			22-16, 6-0				—	✓	✓	✓	✓
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	R/W	0090$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
JPIBC0	JTAG port input buffer control register 0	8	5-0	R/W	0400$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPBDC0	JTAG port bidirection control register 0	8	5-0	R/W	0410$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPU0	Pull-up option register 0	8	5-0	R/W	0430$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPD0	Pull-down option register 0	8	5-0	R/W	0440$_H$	00$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPODC0	JTAG port open drain control register 0	32	5-0	R/W	0450$_H$	0000 0000$_H$	✓	—	—	—	—
			6-0				—	✓	✓	✓	✓
JPDSC0	JTAG port drive strength control register 0	32	5, 3-1	R/W	0460$_H$	0000 0000$_H$	✓	—	—	—	—
			6, 5, 3-1				—	✓	✓	✓	✓
JPIS0	JTAG port input buffer selection register 0	8	5, 3-0	R/W	0470$_H$	FF$_H$	✓	—	—	—	—
			6, 5, 3-0				—	✓	✓	✓	✓
JPISA0	JTAG port input buffer selection advanced register 0	8	3, 2, 0	R/W	04A0$_H$	00$_H$	✓	✓	✓	✓	✓
JPPROTS0	JTAG port protection status register 0	32	0	R	04B0$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
JPPCMD0	JTAG port protection command register 0	32	7-0	W	04C0$_H$	xxxx xx00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.2 Port 0 (P0)

2B.10.2.1 Alternative Function

Table 2B.42 Port 0 (P0)

Port Mode (PMC0_m = 0)	Alternative Mode (PMC0_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P0_0	TAUD0I2	TAUD0O2	RLIN20RX	CAN0TX		PWGA10O	$\overline{\text{CSIH0SSI}}$	DPO	TAUJ2I1	TAUJ2O1							6	13	18	J1	M1
P0_1	TAUD0I4	TAUD0O4	CAN0RX/ INTP0	RLIN20TX	INTP0	PWGA11O	CSIH0SI	APO	TAUJ2I2	TAUJ2O2			CAN0RX				7	14	19	J2	M2
P0_2	TAUD0I6	TAUD0O6	CAN1RX/ INTP1	RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO	TAUJ2I3	TAUJ2O3	CAN1RX				8	15	20	J4	N4
P0_3	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10	CAN1TX	DPIN1	PWGA13O		CSIH0SO	INTP10		TAUJ1I0	TAUJ1O0	RLIN30RX				9	16	21	K1	L3
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O	CSIH1SI	SELDP0	DPIN8		TAUB0I12	TAUB0O12			RLIN31RX				11	18	23	K3	N1
P0_5	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO			TAUB0I14	TAUB0O14			CAN2RX				12	19	24	K2	N2
P0_6	INTP2		DPIN10	SELDP2	CSIH1SC					PWGA35O							13	20	25	L3	M3
P0_7	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	CAN3RX/ INTP3				CAN3RX				40	58	70	R11	V11
P0_8	INTP16	RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX							—	57	69	T12	V10
		RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX							39	—	—	—	—
P0_9	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	CAN4RX/ INTP4				CAN4RX				38	56	68	R10	U10
P0_10	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6		CAN4TX							37	55	67	T11	Y12
P0_11	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8	RLIN26RX	PWGA34O									—	—	26	L1	P1
	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8		PWGA34O									14	21	—	—	—
P0_12	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI	RLIN26TX									—	—	27	L2	P2
	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI										15	22	—	—	—
P0_13	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	CAN5RX/ INTP5		RLIN32RX		CAN5RX				16	23	28	M1	R1
P0_14	INTP17	RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX							—	24	29	L4	P3
		RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX							17	—	—	—	—

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.2.2 Control Registers

Table 2B.43 Control Registers (P0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P0	Port register 0	16	14-0	R/W	0000$_H$	0000$_H$	✓	✓	✓	✓	✓
PSR0	Port set/reset register 0	32	30-16, 14-0	R/W	0100$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPR0	Port pin read register 0	16	14-0	R	0200$_H$	0000$_H$	✓	✓	✓	✓	✓
PM0	Port mode register 0	16	14-0	R/W	0300$_H$	FFFF$_H$	✓	✓	✓	✓	✓
PMC0	Port mode control register 0	16	14-0	R/W	0400$_H$	0000$_H$	✓	✓	✓	✓	✓
PFC0	Port function control register 0	16	14-0	R/W	0500$_H$	0000$_H$	✓	✓	✓	✓	✓
PFCE0	Port function control expansion register 0	16	14-0	R/W	0600$_H$	0000$_H$	✓	✓	✓	✓	✓
PNOT0	Port NOT register 0	16	14-0	W	0700$_H$	0000$_H$	✓	✓	✓	✓	✓
PMSR0	Port mode set/reset register 0	32	30-16, 14-0	R/W	0800$_H$	0000 FFFF$_H$	✓	✓	✓	✓	✓
PMCSR0	Port mode control set/reset register 0	32	30-16, 14-0	R/W	0900$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PFCAE0	Port function control additional expansion register 0	16	14, 13, 10-0	R/W	0A00$_H$	0000$_H$	✓	✓	✓	✓	✓
PIBC0	Port input buffer control register 0	16	14-0	R/W	4000$_H$	0000$_H$	✓	✓	✓	✓	✓
PBDC0	Port bidirection control register 0	16	14-0	R/W	4100$_H$	0000$_H$	✓	✓	✓	✓	✓
PIPC0	Port IP control register 0	16	14, 13, 6, 5, 3, 2	R/W	4200$_H$	0000$_H$	✓	✓	✓	✓	✓
PU0	Pull-up option register 0	16	14-0	R/W	4300$_H$	0000$_H$	✓	✓	✓	✓	✓
PD0	Pull-down option register 0	16	14-0	R/W	4400$_H$	0000$_H$	✓	✓	✓	✓	✓
PODC0	Port open drain control register 0	32	14-0	R/W	4500$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PDSC0	Port drive strength control register 0	32	14-0	R/W	4600$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PIS0	Port input buffer selection register 0	16	14-0	R/W	4700$_H$	FFFF$_H$	✓	✓	✓	✓	✓
PPROTS0	Port protection status register 0	32	0	R	4B00$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPCMD0	Port protection command register 0	32	7-0	W	4C00$_H$	XXXX XX00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.3 Port 1 (P1)

2B.10.3.1 Alternative Function

Table 2B.44 Port 1 (P1)

Port Mode (PMC1_m = 0)	Alternative Mode (PMC1_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative								
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P1_0	RLIN33RX/ INTP13		INTP13				TAUJ2I0	TAUJ2O0					RLIN33RX				—	25	30	M2	T1
P1_1	INTP18	RLIN33TX					TAUJ2I1	TAUJ2O1									—	26	31	N2	R2
P1_2	CAN3RX/ INTP3		INTP3		DPIN19		TAUJ2I2	TAUJ2O2					CAN3RX				—	27	32	N1	U1
P1_3	INTP19	CAN3TX	DPIN23				TAUJ2I3	TAUJ2O3									—	28	33	N3	R4
P1_4	RLIN35RX/ INTP15		INTP15		DPIN18								RLIN35RX				—	63	75	U13	W12
P1_5	ADCA1TRG0	RLIN35TX	DPIN17		INTP20												—	62	74	U12	Y13
P1_8																	—	43	53	U5	Y6
P1_9			DPIN20		INTP21												—	42	52	R7	U6
P1_10	RLIN24RX		DPIN21		INTP22		ADCA1TRG1										—	41	51	R6	Y5
P1_11	ADCA1TRG2	RLIN24TX	DPIN22		INTP14												—	40	50	P6	V6
P1_12	CAN4RX/ INTP4		INTP4	RLIN36TX									CAN4RX				—	—	34	P1	T2
P1_13		CAN4TX	RLIN36RX /INTP16										RLIN36RX				—	—	35	P2	V1
P1_14	RLIN23RX		CAN7RX /INTP9										CAN7RX				—	—	78	U14	Y14
P1_15		RLIN23TX		CAN7TX													—	—	79	R13	U13

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.3.2 Control Registers

Table 2B.45 Control Registers (P1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P1	Port register 1	16	11-8, 5-0	R/W	0004$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PSR1	Port set/reset register 1	32	27-24, 21-16, 11-8, 5-0	R/W	0104$_H$	0000 0000$_H$	—	✓	—	—	—
			31-24, 21-16, 15-8, 5-0				—	—	✓	✓	✓
PPR1	Port pin read register 1	16	11-8, 5-0	R	0204$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PM1	Port mode register 1	16	11-8, 5-0	R/W	0304$_H$	FFFF$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PMC1	Port mode control register 1	16	11-9, 5-0	R/W	0404$_H$	0000$_H$	—	✓	—	—	—
			15-9, 5-0				—	—	✓	✓	✓
PFC1	Port function control register 1	16	11-9, 5-0	R/W	0504$_H$	0000$_H$	—	✓	—	—	—
			15-9, 5-0				—	—	✓	✓	✓
PFCE1	Port function control expansion register 1	16	11-9, 5-0	R/W	0604$_H$	0000$_H$	—	✓	—	—	—
			14-9, 5-0				—	—	✓	✓	✓
PNOT1	Port NOT register 1	16	11-8, 5-0	W	0704$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PMSR1	Port mode set/reset register 1	32	27-24, 21-16, 11-8, 5-0	R/W	0804$_H$	0000 FFFF$_H$	—	✓	—	—	—
			31-24, 21-16, 15-8, 5-0				—	—	✓	✓	✓
PMCSR1	Port mode control set/reset register 1	32	27-25, 21-16, 11-9, 5-0	R/W	0904$_H$	0000 0000$_H$	—	✓	—	—	—
			31-25, 21-16, 15-9, 5-0				—	—	✓	✓	✓
PFCAE1	Port function control additional expansion register 1	16	4, 2, 0	R/W	0A04$_H$	0000$_H$	—	✓	—	—	—
			14-12, 4, 2, 0				—	—	✓	✓	✓
PIBC1	Port input buffer control register 1	16	11-8, 5-0	R/W	4004$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PBDC1	Port bidirection control register 1	16	11-8, 5-0	R/W	4104$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PU1	Pull-up option register 1	16	11-8, 5-0	R/W	4304$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PD1	Pull-down option register 1	16	11-8, 5-0	R/W	4404$_H$	0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PODC1	Port open drain control register 1	32	11-8, 5-0	R/W	4504$_H$	0000 0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PDSC1	Port drive strength control register 1	32	11-8, 5-0	R/W	4604$_H$	0000 0000$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PIS1	Port input buffer selection register 1	16	11-8, 5-0	R/W	4704$_H$	FFFF$_H$	—	✓	—	—	—
			15-8, 5-0				—	—	✓	✓	✓
PPROTS1	Port protection status register 1	32	0	R	4B04$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PPCMD1	Port protection command register 1	32	7-0	W	4C04$_H$	xxxx xx00$_H$	—	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.4 Port 2 (P2)

2B.10.4.1 Alternative Function

Table 2B.46 Port 2 (P2)

Port Mode (PMC2_m =0)	Alternative Mode (PMC2_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P2_0	RLIN27RX		CAN6RX/ INTP6										CAN6RX				—	—	49	T5	W5
P2_1		RLIN27TX		CAN6TX													—	—	48	R5	Y4
P2_2	RLIN28RX																—	—	65	T10	W10
P2_3		RLIN28TX															—	—	64	U10	W9
P2_4	RLIN29RX			ADCA0SEL0													—	—	76	T13	W13
P2_5		RLIN29TX		ADCA0SEL1													—	—	77	R12	V12
P2_6		ADCA0SEL2															—	—	36	R1	R3
P2_7	RLIN210RX																—	—	—	M4	U2
P2_8		RLIN210TX															—	—	—	T1	T3
P2_9		PWGA77O															—	—	—	M3	T4
P2_10		PWGA78O															—	—	—	R2	W1
P2_11		PWGA79O															—	—	—	N4	U3
P2_12	RLIN211RX																—	—	—	T2	V2
P2_13		RLIN211TX															—	—	—	P7	V7
P2_14		PWGA74O															—	—	—	T6	W6
P2_15		PWGA75O															—	—	—	P8	W7

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.4.2 Control Registers

Table 2B.47 Control Registers (P2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P2	Port register 2	16	6-0	R/W	0008$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PSR2	Port set/reset register 2	32	22-16, 6-0	R/W	0108$_H$	0000 0000$_H$	—	—	✓	—	—
			31-16, 15-0				—	—	—	✓	✓
PPR2	Port pin read register 2	16	6-0	R	0208$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PM2	Port mode register 2	16	6-0	R/W	0308$_H$	FFFF$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PMC2	Port mode control register 2	16	6-0	R/W	0408$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PFC2	Port function control register 2	16	5, 4, 1, 0	R/W	0508$_H$	0000$_H$	—	—	✓	✓	✓
PFCE2	Port function control expansion register 2	16	0	R/W	0608$_H$	0000$_H$	—	—	✓	✓	✓
PNOT2	Port NOT register 2	16	6-0	W	0708$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PMSR2	Port mode set/reset register 2	32	22-16, 6-0	R/W	0808$_H$	0000 FFFF$_H$	—	—	✓	—	—
			31-16, 15-0				—	—	—	✓	✓
PMCSR2	Port mode control set/reset register 2	32	22-16, 6-0	R/W	0908$_H$	0000 0000$_H$	—	—	✓	—	—
			31-16, 15-0				—	—	—	✓	✓
PFCAE2	Port function control additional expansion register 2	16	0	R/W	0A08$_H$	0000$_H$	—	—	✓	✓	✓
PIBC2	Port input buffer control register 2	16	6-0	R/W	4008$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PBDC2	Port bidirection control register 2	16	6-0	R/W	4108$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PU2	Pull-up option register 2	16	6-0	R/W	4308$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PD2	Pull-down option register 2	16	6-0	R/W	4408$_H$	0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PODC2	Port open drain control register 2	32	6-0	R/W	4508$_H$	0000 0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PDSC2	Port drive strength control register 2	32	6-0	R/W	4608$_H$	0000 0000$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PIS2	Port input buffer selection register 2	16	6-0	R/W	4708$_H$	FFFF$_H$	—	—	✓	—	—
			15-0				—	—	—	✓	✓
PPROTS2	Port protection status register 2	32	0	R	4B08$_H$	0000 0000$_H$	—	—	✓	✓	✓
PPCMD2	Port protection command register 2	32	7-0	W	4C08$_H$	xxxx xx00$_H$	—	—	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.5 Port 3 (P3)

2B.10.5.1 Alternative Function

Table 2B.48 Port 3 (P3)

Port Mode (PMC3_m =0)	Alternative Mode (PMC3_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P3_0		PWGA76O															—	—	—	R8	V8
P3_1		PWGA80O															—	—	—	—	V14
P3_2		PWGA81O															—	—	—	—	W16
P3_3		PWGA82O															—	—	—	—	V16
P3_4		PWGA83O															—	—	—	—	W18
P3_5		PWGA84O															—	—	—	—	Y19
P3_6		PWGA85O															—	—	—	—	V17
P3_7		PWGA86O															—	—	—	—	U16
P3_8		PWGA87O															—	—	—	—	W19
P3_9																	—	—	—	—	V18
P3_10																	—	—	—	—	U17

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.5.2 Control Registers

Table 2B.49 Control Registers (P3)

			Effective Bit				Device				
Register	Function	Register Size	Position	R/W*[1]	Offset Address	Value after Reset	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P3	Port register 3	16	0	R/W	000C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PSR3	Port set/reset register 3	32	16, 0	R/W	010C$_H$	0000 0000$_H$	—	—	—	✓	—
			26-16, 10-0				—	—	—	—	✓
PPR3	Port pin read register 3	16	0	R	020C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PM3	Port mode register 3	16	0	R/W	030C$_H$	FFFF$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PMC3	Port mode control register 3	16	0	R/W	040C$_H$	0000$_H$	—	—	—	✓	—
			8-0				—	—	—	—	✓
PNOT3	Port NOT register 3	16	0	W	070C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PMSR3	Port mode set/reset register 3	32	16, 0	R/W	080C$_H$	0000 FFFF$_H$	—	—	—	✓	—
			26-16, 10-0				—	—	—	—	✓
PMCSR3	Port mode control set/reset register 3	32	16, 0	R/W	090C$_H$	0000 0000$_H$	—	—	—	✓	—
			24-16, 8-0				—	—	—	—	✓
PIBC3	Port input buffer control register 3	16	0	R/W	400C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PBDC3	Port bidirection control register 3	16	0	R/W	410C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PU3	Pull-up option register 3	16	0	R/W	430C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PD3	Pull-down option register 3	16	0	R/W	440C$_H$	0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PODC3	Port open drain control register 3	32	0	R/W	450C$_H$	0000 0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PDSC3	Port drive strength control register 3	32	0	R/W	460C$_H$	0000 0000$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PIS3	Port input buffer selection register 3	16	0	R/W	470C$_H$	FFFF$_H$	—	—	—	✓	—
			10-0				—	—	—	—	✓
PPROTS3	Port protection status register 3	32	0	R	4B0C$_H$	0000 0000$_H$	—	—	—	✓	✓
PPCMD3	Port protection command register 3	32	7-0	W	4C0C$_H$	0000 0000$_H$	—	—	—	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.6 Port 8 (P8)

2B.10.6.1 Alternative Function

Table 2B.50 Port 8 (P8)

Port Mode (PMC8_m = 0)	Alternative Mode (PMC8_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P8_0	TAUJ0I0	TAUJ0O0	DPIN2	PWGA14O	INTP4	CSIH0CSS0	CAN6RX/ INTP6		RIIC1SDA		SENT0RX		CAN6RX		ADCA0I0S		—	64	80	U15	W14
P8_1	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5	CSIH1CSS3		CAN6TX	RIIC1SCL			SENT0SPCO			ADCA0I1S		—	65	81	T14	V13
P8_2	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O		RLIN37TX							ADCA0I4S		—	—	38	U2	W2
	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O									ADCA0I4S		19	30	—	—	—
P8_3	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1	INTP7	PWGA23O		CAN7TX							ADCA0I5S		44	66	82	U16	Y15
P8_4	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8	PWGA36O	CAN7RX/ INTP9						CAN7RX		ADCA0I6S		45	67	83	R14	W15
P8_5	TAUJ0I3	TAUJ0O3	NMI	CSIH0CSS3	INTP9	PWGA37O									ADCA0I7S		46	68	84	T15	Y16
P8_6	NMI	CSIH0CSS4		PWGA38O		RTCA0OUT									ADCA0I8S	$\overline{\text{RESETOUT}}$	47	69	85	P13	Y17
P8_7		CSIH3CSS0		PWGA39O		ADCA0SEL0		RTCA0OUT							ADCA0I14S		48	70	86	R15	V15
P8_8		CSIH3CSS1		PWGA40O		ADCA0SEL1	RLIN34RX /INTP14						RLIN34RX		ADCA0I15S		—	71	87	P14	W17
		CSIH3CSS1		PWGA40O		ADCA0SEL1									ADCA0I15S		49	—	—	—	—
P8_9		CSIH3CSS2		PWGA41O		ADCA0SEL2		RLIN34TX							ADCA0I16S		—	72	88	T16	Y18
		CSIH3CSS2		PWGA41O		ADCA0SEL2									ADCA0I16S		50	—	—	—	—
P8_10		CSIH3CSS3	DPIN14	PWGA42O			RLIN37RX /INTP17						RLIN37RX		ADCA0I17S		—	—	39	P3	V3
		CSIH3CSS3	DPIN14	PWGA42O											ADCA0I17S		20	31	—	—	—
P8_11	TAUJ1I2	TAUJ1O2	DPIN15	PWGA43O		CSIH1CSS4	RLIN25RX								ADCA0I18S		—	32	40	T3	Y2
	TAUJ1I2	TAUJ1O2	DPIN15	PWGA43O		CSIH1CSS4									ADCA0I18S		21	—	—	—	—
P8_12	TAUJ1I3	TAUJ1O3	DPIN16	PWGA44O		CSIH1CSS5	INTP23	RLIN25TX							ADCA0I19S		—	33	41	P4	U4
	TAUJ1I3	TAUJ1O3	DPIN16	PWGA44O		CSIH1CSS5									ADCA0I19S		22	—	—	—	—

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2B.51, Control Registers (P8)**.
3. When the $\overline{\text{RESETOUT}}$ function is selected for the P8_6 pin, the output on the pin is at the low level during a reset and after release from the reset state. For details, see **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** .

2B.10.6.2 Control Registers

Table 2B.51 Control Registers (P8)

			Effective Bit				Device				
Register	Function	Register Size	Position	R/W*1	Offset Address	Value after Reset	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P8	Port register 8	16	12-2	R/W	0020$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PSR8	Port set/reset register 8	32	28-18, 12-2	R/W	0120$_H$	0000 0000$_H$	✓	—	—	—	—
			28-16, 12-0				—	✓	✓	✓	✓
PPR8	Port pin read register 8	16	12-2	R	0220$_H$	0000$_H$	✓	—	—	—	—
			12-0		0220$_H$		—	✓	✓	✓	✓
PM8	Port mode register 8	16	12-2	R/W	0320$_H$	FFBF$_H$	✓	—	—	—	—
			12-0	R/W			—	✓	✓	✓	✓
PMC8	Port mode control register 8	16	12-2	R/W	0420$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PFC8	Port function control register 8	16	12-2	R/W	0520$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PFCE8	Port function control expansion register 8	16	12, 11, 9-2	R/W	0620$_H$	0000$_H$	✓	—	—	—	—
			12, 11, 9-0				—	✓	—	—	—
			12-0				—	—	✓	✓	✓
PNOT8	Port NOT register 8	16	12-2	W	0720$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PMSR8	Port mode set/reset register 8	32	28-18, 12-2	R/W	0820$_H$	0000 FFBF$_H$	✓	—	—	—	—
			28-16, 12-0				—	✓	✓	✓	✓
PMCSR8	Port mode control set/reset register 8	32	28-18, 12-2	R/W	0920$_H$	0000 0000$_H$	✓	—	—	—	—
			28-16, 12-0				—	✓	✓	✓	✓
PFCAE8	Port function control additional expansion register 8	16	4	R/W	0A20$_H$	0000$_H$	✓	—	—	—	—
			8, 4, 1, 0				—	✓	—	—	—
			10, 8, 4, 1, 0				—	—	✓	✓	✓
PIBC8	Port input buffer control register 8	16	12-2	R/W	4020$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PBDC8	Port bidirection control register 8	16	12-2	R/W	4120$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PU8	Pull-up option register 8	16	12-2	R/W	4320$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PD8	Pull-down option register 8	16	12-2	R/W	4420$_H$	0000$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PODC8	Port open drain control register 8	32	12-2	R/W	4520$_H$	0000 0040$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PIS8	Port input buffer selection register 8	16	12-2	R/W	4720$_H$	FFFF$_H$	✓	—	—	—	—
			12-0				—	✓	✓	✓	✓
PPROTS8	Port protection status register 8	32	0	R	4B20$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPCMD8	Port protection command register 8	32	7-0	W	4C20$_H$	xxxx xx00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

CAUTION

P8_6 drives a low level after any kind of reset release, until it is later configured differently by register settings. For details, see **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** .

2B.10.7 Port 9 (P9)

2B.10.7.1 Alternative Function

Table 2B.52 Port 9 (P9)

Port Mode (PMC9_m = 0)	Alternative Mode (PMC9_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P9_0	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0	CSIH2CSS0	KR0I4		TAUJ1I1	TAUJ1O1	SENT1RX		RIIC1SDA		ADCA0I2S		69	92	108	K16	N18
P9_1	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5	CSIH2CSS1			TAUJ1I2	TAUJ1O2		SENT1SPCO	RIIC1SCL		ADCA0I3S		70	93	109	K17	N19
P9_2	KR0I6	PWGA20O	TAPA0ESO	CSIH2CSS2											ADCA0I9S		71	94	110	J17	N20
P9_3	KR0I7	PWGA21O		CSIH2CSS3	TAUJ1I1	TAUJ1O1	INTP16								ADCA0I10S		—	95	111	J15	M18
	KR0I7	PWGA21O		CSIH2CSS3	TAUJ1I1	TAUJ1O1									ADCA0I10S		72	—	—	—	—
P9_4		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0	INTP17								ADCA0I11S		—	96	112	J16	M19
		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0									ADCA0I11S		73	—	—	—	—

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2B.53, Control Registers (P9)**.

2B.10.7.2 Control Registers

Table 2B.53 Control Registers (P9)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P9	Port register 9	16	4-0	R/W	0024$_H$	0000$_H$	✓	✓	✓	✓	✓
PSR9	Port set/reset register 9	32	20-16, 4-0	R/W	0124$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPR9	Port pin read register 9	16	4-0	R	0224$_H$	0000$_H$	✓	✓	✓	✓	✓
PM9	Port mode register 9	16	4-0	R/W	0324$_H$	FFFF$_H$	✓	✓	✓	✓	✓
PMC9	Port mode control register 9	16	4-0	R/W	0424$_H$	0000$_H$	✓	✓	✓	✓	✓
PFC9	Port function control register 9	16	4-0	R/W	0524$_H$	0000$_H$	✓	✓	✓	✓	✓
PFCE9	Port function control expansion register 9	16	4, 3, 1, 0	R/W	0624$_H$	0000$_H$	✓	✓	✓	✓	✓
PNOT9	Port NOT register 9	16	4-0	W	0724$_H$	0000$_H$	✓	✓	✓	✓	✓
PMSR9	Port mode set/reset register 9	32	20-16, 4-0	R/W	0824$_H$	0000 FFFF$_H$	✓	✓	✓	✓	✓
PMCSR9	Port mode control set/reset register 9	32	20-16, 4-0	R/W	0924$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PFCAE9	Port function control additional expansion register 9	16	1, 0	R/W	0A24$_H$	0000$_H$	✓	✓	✓	✓	✓
PIBC9	Port input buffer control register 9	16	4-0	R/W	4024$_H$	0000$_H$	✓	✓	✓	✓	✓
PBDC9	Port bidirection control register 9	16	4-0	R/W	4124$_H$	0000$_H$	✓	✓	✓	✓	✓
PU9	Pull-up option register 9	16	4-0	R/W	4324$_H$	0000$_H$	✓	✓	✓	✓	✓
PD9	Pull-down option register 9	16	4-0	R/W	4424$_H$	0000$_H$	✓	✓	✓	✓	✓
PODC9	Port open drain control register 9	32	4-0	R/W	4524$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PIS9	Port input buffer selection register 9	16	4-0	R/W	4724$_H$	FFFF$_H$	✓	✓	✓	✓	✓
PPROTS9	Port protection status register 9	32	0	R	4B24$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPCMD9	Port protection command register 9	32	7-0	W	4C24$_H$	xxxx xx00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.8 Port 10 (P10)

2B.10.8.1 Alternative Function

Table 2B.54 Port 10 (P10)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P10_0	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O	TAUJ1I3	TAPA0UP	CSIH1SI	MEMC0A19	ETNB0RXCLK	TAUJ1O3	CAN0RX				—	—	174	A2	C4
	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O	TAUJ1I3	TAPA0UP	CSIH1SI			TAUJ1O3	CAN0RX				98	142	—	—	—
P10_1	TAUD0I3	TAUD0O3	INTP18	CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN	CSIH1SC		ETNB0RXD0	MEMC0A20		TAUJ3O0		MODE0	—	—	175	B2	B3
	TAUD0I3	TAUD0O3	INTP18	CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN	CSIH1SC					TAUJ3O0		MODE0	—	143	—	—	—
	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN	CSIH1SC					TAUJ3O0		MODE0	99	—	—	—	—
P10_2	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO	ETNB0RXD1	MEMC0A21		RLIN37TX		MODE1	—	—	176	C3	C3
	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO						MODE1	100	144	—	—	—
P10_3	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$	MEMC0CLK	RLIN37RX/ INTP17		RLIN37RX				—	—	1	B1	B2
	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$								1	1	—	—	—
P10_4	TAUD0I9	TAUD0O9	RLIN21RX	CAN6TX	KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$	PWGA53O	ETNB0RXD2	MEMC0A22					—	—	2	D3	E3
	TAUD0I9	TAUD0O9	RLIN21RX	CAN6TX	KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$	PWGA53O							—	2	—	—	—
	TAUD0I9	TAUD0O9	RLIN21RX	CAN6TX	KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$								2	—	—	—	—
P10_5	TAUD0I11	TAUD0O11	CAN6RX/ INTP6	RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO	ETNB0RXD3	PWGA54O	CAN6RX				—	—	3	C2	D3
	TAUD0I11	TAUD0O11	CAN6RX/ INTP6	RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO		PWGA54O	CAN6RX				—	3	—	—	—
	TAUD0I11	TAUD0O11	CAN6RX/ INTP6	RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO			CAN6RX				3	—	—	—	—
P10_6	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX /INTP1		MEMC0AD0		RLIN24RX		CAN1RX			MODE2	—	—	152	A9	A12
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX /INTP1				RLIN24RX		CAN1RX			MODE2	—	120	—	—	—
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX /INTP1						CAN1RX			MODE2	80	—	—	—	—

Table 2B.54 Port 10 (P10)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P10_7	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX	MEMC0AD1			RLIN24TX	TAUJ3I1	TAUJ3O1			—	—	153	A8	C11
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX				RLIN24TX	TAUJ3I1	TAUJ3O1			—	121	—	—	—
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX					TAUJ3I1	TAUJ3O1			81	—	—	—	—
P10_8	TAUD0I10	TAUD0O10	CSIG0SI	FLXA0TXDB	ENCA0EC	PWGA5O	MEMC0AD2						TAUJ3I2	TAUJ3O2		FLMD1	—	—	154	D8	B11
	TAUD0I10	TAUD0O10	CSIG0SI	FLXA0TXDB	ENCA0EC	PWGA5O							TAUJ3I2	TAUJ3O2		FLMD1	82	122	—	—	—
P10_9	TAUD0I12	TAUD0O12	RLIN30RX /INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO	MEMC0AD3		FLXA0RXDB		RLIN30RX				—	—	155	B8	C12
	TAUD0I12	TAUD0O12	RLIN30RX /INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO			FLXA0RXDB		RLIN30RX				83	123	—	—	—
P10_10	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1	MEMC0AD4				TAUJ3I3	TAUJ3O3			—	—	156	A7	A11
	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1					TAUJ3I3	TAUJ3O3			84	124	—	—	—
P10_11		PWGA16O	RLIN31RX /INTP11	FLXA0TXENA		CSIH1CSS0	TAUB0I1	TAUB0O1	MEMC0AD5				RLIN31RX				—	—	157	C8	D11
		PWGA16O	RLIN31RX /INTP11	FLXA0TXENA		CSIH1CSS0	TAUB0I1	TAUB0O1					RLIN31RX				85	125	—	—	—
P10_12		PWGA17O	FLXA0STPWT	RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3	MEMC0AD6								—	—	158	D7	A10
		PWGA17O	FLXA0STPWT	RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3									86	126	—	—	—
P10_13	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX /INTP12	FLXA0TXENB			TAUB0I5	TAUB0O5	MEMC0AD7			CAN7TX	RLIN32RX				—	—	159	A6	C10
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX /INTP12	FLXA0TXENB			TAUB0I5	TAUB0O5				CAN7TX	RLIN32RX				87	127	—	—	—

Table 2B.54 Port 10 (P10)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P10_14	ADCA1TRG0	PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7	MEMC0AD8		CAN7RX/INTP9		CAN7RX				—	—	160	B7	B10
	ADCA1TRG0	PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7			CAN7RX/INTP9		CAN7RX				—	128	—	—	—
		PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7			CAN7RX/INTP9		CAN7RX				88	—	—	—	—
P10_15	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9		$\overline{\text{MEMC0RD}}$							—	—	6	C1	D1
	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9									—	4	—	—	—

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.8.2 Control Registers

Table 2B.55 Control Registers (P10)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P10	Port register 10	16	14-0	R/W	0028$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	0028$_H$	0000$_H$	—	✓	✓	✓	✓
PSR10	Port set/reset register 10	32	30-16, 14-0	R/W	0128$_H$	0000 0000$_H$	✓	—	—	—	—
			31-16, 15-0	R/W	0128$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PPR10	Port pin read register 10	16	14-0	R	0228$_H$	0000$_H$	✓	—	—	—	—
			15-0	R	0228$_H$	0000$_H$	—	✓	✓	✓	✓
PM10	Port mode register 10	16	14-0	R/W	0328$_H$	FFFF$_H$	✓	—	—	—	—
			15-0	R/W	0328$_H$	FFFF$_H$	—	✓	✓	✓	✓
PMC10	Port mode control register 10	16	14-0	R/W	0428$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	0428$_H$	0000$_H$	—	✓	✓	✓	✓
PFC10	Port function control register 10	16	14-0	R/W	0528$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	0528$_H$	0000$_H$	—	✓	✓	✓	✓
PFCE10	Port function control expansion register 10	16	14-0	R/W	0628$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	0628$_H$	0000$_H$	—	✓	✓	✓	✓
PNOT10	Port NOT register 10	16	14-0	W	0728$_H$	0000$_H$	✓	—	—	—	—
			15-0	W	0728$_H$	0000$_H$	—	✓	✓	✓	✓
PMSR10	Port mode set/reset register 10	32	30-16, 14-0	R/W	0828$_H$	0000 FFFF$_H$	✓	—	—	—	—
			31-16, 15-0	R/W	0828$_H$	0000 FFFF$_H$	—	✓	✓	✓	✓
PMCSR10	Port mode control set/reset register 10	32	30-16, 14-0	R/W	0928$_H$	0000 0000$_H$	✓	—	—	—	—
			31-16, 15-0	R/W	0928$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PFCAE10	Port function control additional expansion register 10	16	14-0	R/W	0A28$_H$	0000$_H$	✓	✓	—	—	—
			15-0				—	—	✓	✓	✓
PIBC10	Port input buffer control register 10	16	14-0	R/W	4028$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	4028$_H$	0000$_H$	—	✓	✓	✓	✓
PBDC10	Port bidirection control register 10	16	14-0	R/W	4128$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	4128$_H$	0000$_H$	—	✓	✓	✓	✓
PIPC10	Port IP control register 10	16	7-0	R/W	4228$_H$	0000$_H$	✓	✓	—	—	—
			14-0				—	—	✓	✓	✓
PU10	Pull-up option register 10	16	14-0	R/W	4328$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	4328$_H$	0000$_H$	—	✓	✓	✓	✓
PD10	Pull-down option register 10	16	14-0	R/W	4428$_H$	0000$_H$	✓	—	—	—	—
			15-0	R/W	4428$_H$	0000$_H$	—	✓	✓	✓	✓
PODC10	Port open drain control register 10	32	14-0	R/W	4528$_H$	0000 0000$_H$	✓	—	—	—	—
			15-0	R/W	4528$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PDSC10	Port drive strength control register 10	32	14-0	R/W	4628$_H$	0000 0000$_H$	✓	—	—	—	—
			15-0	R/W	4628$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PIS10	Port input buffer selection register 10	16	14-0	R/W	4728$_H$	FFFF$_H$	✓	—	—	—	—
			15-0	R/W	4728$_H$	FFFF$_H$	—	✓	✓	✓	✓
PISA10	Port input buffer selection advanced register 10	16	5, 4, 2-0	R/W	4A28$_H$	0000$_H$	—	—	✓	✓	✓
PPROTS10	Port protection status register 10	32	0	R	4B28$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPCMD10	Port protection command register 10	32	7-0	W	4C28$_H$	xxxx xx00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset..

2B.10.9 Port 11 (P11)

2B.10.9.1 Alternative Function

Table 2B.56 Port 11 (P11)

Port mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P11_0	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11		$\overline{\text{MEMC0WR}}$							—	—	7	D2	E1
	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11									—	5	—	—	—
P11_1	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7	INTP20	PWGA26O	TAUB0I13	TAUB0O13	MEMC0AD9								—	—	161	A5	A9
	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7	INTP20	PWGA26O	TAUB0I13	TAUB0O13									—	129	—	—	—
	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7		PWGA26O	TAUB0I13	TAUB0O13									89	—	—	—	—
P11_2		CSIH2SO	RLIN32RX /INTP12	RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15	MEMC0AD10		SFMA0IO3		RLIN32RX				—	—	162	C7	D9
		CSIH2SO	RLIN32RX /INTP12	RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15			SFMA0IO3		RLIN32RX				—	130	—	—	—
		CSIH2SO	RLIN32RX /INTP12	RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15					RLIN32RX				90	—	—	—	—
P11_3	CSIH2SC		CAN3RX /INTP3	PWGA28O	TAUB1I1	TAUB1O1	MEMC0AD11			RLIN32TX	SFMA0IO2		CAN3RX				—	—	163	B6	C9
	CSIH2SC		CAN3RX /INTP3	PWGA28O						RLIN32TX	SFMA0IO2		CAN3RX				—	131	—	—	—
	CSIH2SC		CAN3RX /INTP3	PWGA28O						RLIN32TX			CAN3RX				91	—	—	—	—
P11_4	CSIH2SI	CAN3TX	INTP21	PWGA29O	TAUB1I3	TAUB1O3	MEMC0AD12				SFMA0IO1						—	—	164	B5	B8
	CSIH2SI	CAN3TX	INTP21	PWGA29O							SFMA0IO1						—	132	—	—	—
	CSIH2SI	CAN3TX		PWGA29O													92	—	—	—	—
P11_5	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI		TAUB1I5	TAUB1O5	MEMC0AD13		SFMA0IO0		CAN5RX				—	—	165	A4	B7
	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI						SFMA0IO0		CAN5RX				—	133	—	—	—
	CAN5RX/ INTP5			PWGA30O	CSIH3SI								CAN5RX				93	—	—	—	—

Table 2B.56 Port 11 (P11)

Port mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P11_6	RLIN33RX/INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO	TAUB1I7	TAUB1O7	MEMC0AD14			SFMA0SSL	RLIN33RX				—	—	166	D6	C7
	RLIN33RX/INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO						SFMA0SSL	RLIN33RX				—	134	—	—	—
	INTP13	CAN5TX		PWGA31O		CSIH3SO											94	—	—	—	—
P11_7	INTP5			PWGA32O	CSIH3SC		TAUB1I9	TAUB1O9	MEMC0AD15			SFMA0CLK					—	—	167	C6	D8
	INTP5			PWGA32O	CSIH3SC							SFMA0CLK					—	135	—	—	—
	INTP5			PWGA32O	CSIH3SC												95	—	—	—	—
P11_8	$\overline{\text{CSIG1SSI}}$	RLIN35TX		PWGA48O	TAUB1I11	TAUB1O11		$\overline{\text{MEMC0CS0}}$									—	—	8	E3	E2
	$\overline{\text{CSIG1SSI}}$	RLIN35TX		PWGA48O													—	6	—	—	—
P11_9		CSIG1SO	RLIN35RX/INTP15	PWGA49O	TAUB1I13	TAUB1O13		$\overline{\text{MEMC0CS1}}$					RLIN35RX				—	—	9	D1	F3
		CSIG1SO	RLIN35RX/INTP15	PWGA49O									RLIN35RX				—	7	—	—	—
P11_10	CSIG1SC			PWGA50O	TAUB1I15	TAUB1O15		$\overline{\text{MEMC0CS2}}$									—	—	10	E2	G3
	CSIG1SC			PWGA50O													—	8	—	—	—
P11_11	CSIG1SI	RLIN25TX		PWGA51O	TAUB1I0	TAUB1O0		$\overline{\text{MEMC0CS3}}$	ETNB0RXDV								—	—	11	F3	G2
	CSIG1SI	RLIN25TX		PWGA51O													—	9	—	—	—
P11_12	RLIN25RX			PWGA52O	TAUB1I2	TAUB1O2	$\overline{\text{MEMC0WAIT}}$										—	—	12	E1	H2
	RLIN25RX			PWGA52O													—	10	—	—	—
P11_15	CAN2RX/INTP2	CSIH2CSS4		PWGA55O	TAUB1I8	TAUB1O8		$\overline{\text{MEMC0ASTB}}$	ETNB0RXERR	RLIN36TX			CAN2RX				—	—	168	D5	A5
	CAN2RX/INTP2	CSIH2CSS4		PWGA55O									CAN2RX				—	136	—	—	—

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.9.2 Control Registers

Table 2B.57 Control Registers (P11)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P11	Port register 11	16	7-1	R/W	002C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PSR11	Port set/reset register 11	32	23-17, 7-1	R/W	012C$_H$	0000 0000$_H$	✓	—	—	—	—
			31, 28-16, 15, 12-0				—	✓	✓	✓	✓
PPR11	Port pin read register 11	16	7-1	R	022C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PM11	Port mode register 11	16	7-1	R/W	032C$_H$	FFFF$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PMC11	Port mode control register 11	16	7-1	R/W	042C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PFC11	Port function control register 11	16	7-1	R/W	052C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PFCE11	Port function control expansion register 11	16	7-5, 3-1	R/W	062C$_H$	0000$_H$	✓	—	—	—	—
			15, 9, 7-5, 3-0				—	✓	—	—	—
			15, 12-0				—	—	✓	✓	✓
PNOT11	Port NOT register 11	16	7-1	W	072C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PMSR11	Port mode set/reset register 11	32	23-17, 7-1	R/W	082C$_H$	0000 FFFF$_H$	✓	—	—	—	—
			31, 28-16, 15, 12-0				—	✓	✓	✓	✓
PMCSR11	Port mode control set/reset register 11	32	23-17, 7-1	R/W	092C$_H$	0000 0000$_H$	✓	—	—	—	—
			31, 28-16, 15, 12-0				—	✓	✓	✓	✓
PFCAE11	Port function control additional expansion register 11	16	5, 3, 2	R/W	0A2C$_H$	0000$_H$	✓	—	—	—	—
			15, 9, 7-2				—	✓	—	—	—
			15, 11, 9, 7-0				—	—	✓	✓	✓
PIBC11	Port input buffer control register 11	16	7-1	R/W	402C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PBDC11	Port bidirection control register 11	16	7-1	R/W	412C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PIPC11	Port IP control register 11	16	7, 6, 3, 2	R/W	422C$_H$	0000$_H$	✓	—	—	—	—
			10, 9, 7-2				—	✓	—	—	—
			10, 9, 7-1				—	—	✓	✓	✓
PU11	Pull-up option register 11	16	7-1	R/W	432C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PD11	Pull-down option register 11	16	7-1	R/W	442C$_H$	0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PODC11	Port open drain control register 11	32	7-1	R/W	452C$_H$	0000 0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PDSC11	Port drive strength control register 11	32	7-1	R/W	462C$_H$	0000 0000$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PIS11	Port input buffer selection register 11	16	7-1	R/W	472C$_H$	FFFF$_H$	✓	—	—	—	—
			15, 12-0				—	✓	✓	✓	✓
PISA11	Port input buffer selection advanced register 11	16	15, 12-10	R/W	4A2C$_H$	0000$_H$	—	—	✓	✓	✓
PPROTS11	Port protection status register 11	32	0	R	4B2C$_H$	0000 0000$_H$	✓	✓	✓	✓	✓
PPCMD11	Port protection command register 11	32	7-0	W	4C2C$_H$	xxxx xx00$_H$	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.10 Port 12 (P12)

2B.10.10.1 Alternative Function

Table 2B.58 Port 12 (P12)

Port Mode (PMC12_m =0)	Alternative Mode (PMC12_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P12_0		CAN2TX		PWGA56O	TAUB1I10	TAUB1O10	$\overline{\text{CSIG2SSI}}$	MEMC0A16	RLIN36RX/INTP16				RLIN36RX				—	—	169	B4	D7
		CAN2TX		PWGA56O													—	137	—	—	—
P12_1	RLIN34RX/ INTP14	CSIH2CSS5		PWGA57O	TAUB1I12	TAUB1O12		MEMC0A17					RLIN34RX				—	—	170	C5	B5
	RLIN34RX/ INTP14	CSIH2CSS5		PWGA57O									RLIN34RX				—	138	—	—	—
P12_2	INTP19	RLIN34TX		PWGA58O	TAUB1I14	TAUB1O14		MEMC0A18	CSIG2RYI	CSIG2RYO							—	—	171	A3	C5
	INTP19	RLIN34TX		PWGA58O													—	139	—	—	—
P12_3	RLIN27RX			PWGA68O	CSIG2SI	$\overline{\text{MEMC0BEN0}}$	TAUB1I6	TAUB1O6									—	—	15	G1	L1
P12_4		RLIN27TX		PWGA69O	CSIG2SC		ETNB0MDIO			$\overline{\text{MEMC0BEN1}}$							—	—	16	H1	N3
P12_5		PWGA70O		ETNB0MDC		CSIG2SO	TAUB1I4	TAUB1O4									—	—	17	J3	L2

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.10.2 Control Registers

Table 2B.59 Control Registers (P12)

			Effective Bit				Device				
Register	Function	Register Size	Position	R/W*1	Offset Address	Value after Reset	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P12	Port register 12	16	2-0	R/W	0030$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PSR12	Port set/reset register 12	32	18-16, 2-0	R/W	0130$_H$	0000 0000$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	✓
PPR12	Port pin read register 12	16	2-0	R	0230$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PM12	Port mode register 12	16	2-0	R/W	0330$_H$	FFFF$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PMC12	Port mode control register 12	16	2-0	R/W	0430$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PFC12	Port function control register 12	16	2-0	R/W	0530$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PFCE12	Port function control expansion register 12	16	1	R/W	0630$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PNOT12	Port NOT register 12	16	2-0	W	0730$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PMSR12	Port mode set/reset register 12	32	18-16, 2-0	R/W	0830$_H$	0000 FFFF$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	✓
PMCSR12	Port mode control set/reset register 12	32	18-16, 2-0	R/W	0930$_H$	0000 0000$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	✓
PFCAE12	Port function control expansion register 12	16	1	R/W	0A30$_H$	0000$_H$	—	✓	—	—	—
			4, 2-0				—	—	✓	✓	✓
PIBC12	Port input buffer control register 12	16	2-0	R/W	4030$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PBDC12	Port bidirection control register 12	16	2-0	R/W	4130$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PIPC12	Port IP control register 12	16	5, 4	R/W	4230$_H$	0000$_H$	—	—	✓	✓	✓
PU12	Pull-up option register 12	16	2-0	R/W	4330$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PD12	Pull-down option register 12	16	2-0	R/W	4430$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PODC12	Port open drain control register 12	32	2-0	R/W	4530$_H$	0000 0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PDSC12	Port drive strength control register 12	32	2-0	R/W	4630$_H$	0000 0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PIS12	Port input buffer selection register 12	16	2-0	R/W	4730$_H$	FFFF$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PISA12	Port input buffer selection advanced register 12	16	4	R/W	4A30$_H$	0000$_H$	—	—	✓	✓	✓
PPROTS12	Port protection status register 12	32	0	R	4B30$_H$	0000 0000$_H$	—	✓	✓	✓	✓
PPCMD12	Port protection command register 12	32	7-0	W	4C30$_H$	xxxx xx00$_H$	—	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.11 Port 13 (P13)

2B.10.11.1 Alternative Function

Table 2B.60 Port 13 (P13)

Port Mode (PMC13_m =0)	Alternative Mode (PMC13_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative								
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P13_0		MEMC0A19															—	—	—	C4	D6
P13_1		MEMC0A20															—	—	—	B3	A3
P13_2	ETNB0RXDV																—	—	—	F2	K3
P13_3	ETNB0RXERR																—	—	—	F1	J1
P13_4																	—	—	—	G2	K1
P13_5		MEMC0A21															—	—	—	G3	L4
P13_6		MEMC0A22		PWGA72O													—	—	—	H3	K2
P13_7		MEMC0A23		PWGA73O													—	—	—	—	M4
				PWGA73O													—	—	—	H2	—

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.11.2 Control Registers

Table 2B.61 Control Registers (P13)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P13	Port register 13	16	7-0	R/W	0034_H	0000_H	—	—	—	✓	✓
PSR13	Port set/reset register 13	32	23-16, 7-0	R/W	0134_H	$0000\ 0000_H$	—	—	—	✓	✓
PPR13	Port pin read register 13	16	7-0	R	0234_H	0000_H	—	—	—	✓	✓
PM13	Port mode register 13	16	7-0	R/W	0334_H	$FFFF_H$	—	—	—	✓	✓
PMC13	Port mode control register 13	16	7-5, 3-0	R/W	0434_H	0000_H	—	—	—	✓	✓
PFC13	Port function control register 13	16	7, 6	R/W	0534_H	0000_H	—	—	—	✓	✓
PNOT13	Port NOT register 13	16	7-0	W	0734_H	0000_H	—	—	—	✓	✓
PMSR13	Port mode set/reset register 13	32	23-16, 7-0	R/W	0834_H	$0000\ FFFF_H$	—	—	—	✓	✓
PMCSR13	Port mode control set/reset register 13	32	23-21, 19-16, 7-5, 3-0	R/W	0934_H	$0000\ 0000_H$	—	—	—	✓	✓
PIBC13	Port input buffer control register 13	16	7-0	R/W	4034_H	0000_H	—	—	—	✓	✓
PBDC13	Port bidirection control register 13	16	7-0	R/W	4134_H	0000_H	—	—	—	✓	✓
PU13	Pull-up option register 13	16	7-0	R/W	4334_H	0000_H	—	—	—	✓	✓
PD13	Pull-down option register 13	16	7-0	R/W	4434_H	0000_H	—	—	—	✓	✓
PODC13	Port open drain control register 13	32	7-0	R/W	4534_H	$0000\ 0000_H$	—	—	—	✓	✓
PDSC13	Port drive strength control register 13	32	7-0	R/W	4634_H	$0000\ 0000_H$	—	—	—	✓	✓
PIS13	Port input buffer selection register 13	16	7-0	R/W	4734_H	$FFFF_H$	—	—	—	✓	✓
PISA13	Port input buffer selection advanced register 13	16	5, 3, 2	R/W	$4A34_H$	0000_H	—	—	—	✓	✓
PPROTS13	Port protection status register 13	32	0	R	$4B34_H$	$0000\ 0000_H$	—	—	—	✓	✓
PPCMD13	Port protection command register 13	32	7-0	W	$4C34_H$	$0000\ 0000_H$	—	—	—	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.12 Port 18 (P18)

2B.10.12.1 Alternative Function

Table 2B.62 Port 18 (P18)

Port Mode (PMC18_m = 0)	Alternative Mode (PMC18_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P18_0	CSIG1RYI	CSIG1RYO	ETNB0LINK	PWGA61O			TAUJ3I0	TAUJ3O0							ADCA1I0S		—	—	143	C14	A19
	CSIG1RYI	CSIG1RYO		PWGA61O			TAUJ3I0	TAUJ3O0							ADCA1I0S		—	115	—	—	—
P18_1		PWGA62O		ETNB0TXD0			TAUJ3I1	TAUJ3O1							ADCA1I1S		—	—	144	B15	B18
		PWGA62O					TAUJ3I1	TAUJ3O1							ADCA1I1S		—	116	—	—	—
P18_2		PWGA63O		ETNB0TXD1			TAUJ3I2	TAUJ3O2							ADCA1I2S		—	—	145	B14	B17
		PWGA63O					TAUJ3I2	TAUJ3O2							ADCA1I2S		—	117	—	—	—
P18_3		PWGA71O		ETNB0TXD2			TAUJ3I3	TAUJ3O3							ADCA1I3S		—	—	146	B13	A17
							TAUJ3I3	TAUJ3O3							ADCA1I3S		—	118	—	—	—
P18_4		CSIH1CSS4		ETNB0TXD3											ADCA1I4S		—	—	147	C11	B16
P18_5		CSIH1CSS5		ETNB0TXEN											ADCA1I5S		—	—	148	A14	D14
P18_6															ADCA1I6S		—	—	149	A13	A16
P18_7	ETNB0TXCLK														ADCA1I7S		—	—	150	B11	B14
P18_8															ADCA1I8S		—	—	—	A16	C17
P18_9															ADCA1I9S		—	—	—	C13	A18
P18_10															ADCA1I10S		—	—	—	A15	C16
P18_11															ADCA1I11S		—	—	—	B12	C15
P18_12															ADCA1I12S		—	—	—	C12	C14
P18_13															ADCA1I13S		—	—	—	A12	B15
P18_14															ADCA1I14S		—	—	—	C9	A15
P18_15															ADCA1I15S		—	—	—	A11	C13

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2B.63, Control Registers (P18)**.

2B.10.12.2 Control Registers

Table 2B.63 Control Registers (P18)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P18	Port register 18	16	3-0	R/W	0048$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PSR18	Port set/reset register 18	32	19-16, 3-0	R/W	0148$_H$	0000 0000$_H$	—	✓	—	—	—
			23-16, 7-0				—	—	✓	—	—
			31-16, 15-0				—	—	—	✓	✓
PPR18	Port pin read register 18	16	3-0	R	0248$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PM18	Port mode register 18	16	3-0	R/W	0348$_H$	FFFF$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PMC18	Port mode control register 18	16	3-0	R/W	0448$_H$	0000$_H$	—	✓	—	—	—
			7, 5-0				—	—	✓	✓	✓
PFC18	Port function control register 18	16	3-0	R/W	0548$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PFCE18	Port function control expansion register 18	16	3-0	R/W	0648$_H$	0000$_H$	—	✓	✓	✓	✓
PNOT18	Port NOT register 18	16	3-0	W	0748$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PMSR18	Port mode set/reset register 18	32	19-16, 3-0	R/W	0848$_H$	0000 FFFF$_H$	—	✓	—	—	—
			23-16, 7-0				—	—	✓	—	—
			31-16, 15-0				—	—	—	✓	✓
PMCSR18	Port mode control set/reset register 18	32	18-16, 3-0	R/W	0948$_H$	0000 0000$_H$	—	✓	—	—	—
			23, 21-16, 7, 5-0				—	—	✓	✓	✓
PIBC18	Port input buffer control register 18	16	3-0	R/W	4048$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PBDC18	Port bidirection control register 18	16	3-0	R/W	4148$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PU18	Pull-up option register 18	16	3-0	R/W	4348$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PD18	Pull-down option register 18	16	3-0	R/W	4448$_H$	0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PODC18	Port open drain control register 18	32	3-0	R/W	4548$_H$	0000 0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PDSC18	Port drive strength control register 18	32	3-0	R/W	4648$_H$	0000 0000$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓
PIS18	Port input buffer selection register 18	16	3-0	R/W	4748$_H$	FFFF$_H$	—	✓	—	—	—
			7-0				—	—	✓	—	—
			15-0				—	—	—	✓	✓

Table 2B.63 Control Registers (P18)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
PISA18	Port input buffer selection advanced register 18	16	7, 0	R/W	$4A48_H$	0000_H	—	—	✓	—	—
			9-7, 0				—	—	—	✓	✓
PPROTS18	Port protection status register 18	32	0	R	$4B48_H$	$0000\ 0000_H$	—	✓	✓	✓	✓
PPCMD18	Port protection command register 18	32	7-0	W	$4C48_H$	xxxx xx00_H	—	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.13 Port 19 (P19)

2B.10.13.1 Alternative Function

Table 2B.64 Port 19 (P19)

Port Mode (PMC19_m =0)	Alternative Mode (PMC19_m =1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P19_0															ADCA1I16S		—	—	—	C10	A14
P19_1															ADCA1I17S		—	—	—	B10	B13
P19_2															ADCA1I18S		—	—	—	A10	B12
P19_3															ADCA1I19S		—	—	—	B9	A13

CAUTION

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use ADC functions with their initial settings. For details, see **Table 2B.65, Control Registers (P19)**.

2B.10.13.2 Control Registers

Table 2B.65 Control Registers (P19)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P19	Port register 19	16	3-0	R/W	004C$_H$	0000$_H$	—	—	—	✓	✓
PSR19	Port set/reset register 19	32	19-16, 3-0	R/W	014C$_H$	0000 0000$_H$	—	—	—	✓	✓
PPR19	Port pin read register 19	16	3-0	R	024C$_H$	0000$_H$	—	—	—	✓	✓
PM19	Port mode register 19	16	3-0	R/W	034C$_H$	FFFF$_H$	—	—	—	✓	✓
PNOT19	Port NOT register 19	16	3-0	W	074C$_H$	0000$_H$	—	—	—	✓	✓
PMSR19	Port mode set/reset register 19	32	19-16, 3-0	R/W	084C$_H$	0000 FFFF$_H$	—	—	—	✓	✓
PIBC19	Port input buffer control register 19	16	3-0	R/W	404C$_H$	0000$_H$	—	—	—	✓	✓
PBDC19	Port bidirection control register 19	16	3-0	R/W	414C$_H$	0000$_H$	—	—	—	✓	✓
PU19	Pull-up option register 19	16	3-0	R/W	434C$_H$	0000$_H$	—	—	—	✓	✓
PD19	Pull-down option register 19	16	3-0	R/W	444C$_H$	0000$_H$	—	—	—	✓	✓
PODC19	Port open drain control register 19	32	3-0	R/W	454C$_H$	0000 0000$_H$	—	—	—	✓	✓
PDSC19	Port drive strength control register 19	32	3-0	R/W	464C$_H$	0000 0000$_H$	—	—	—	✓	✓
PIS19	Port input buffer selection register 19	16	3-0	R/W	474C$_H$	FFFF$_H$	—	—	—	✓	✓
PPROTS19	Port protection status register 19	32	0	R	4B4C$_H$	0000 0000$_H$	—	—	—	✓	✓
PPCMD19	Port protection command register 19	32	7-0	W	4C4C$_H$	0000 0000$_H$	—	—	—	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.14 Port 20 (P20)

2B.10.14.1 Alternative Function

Table 2B.66 Port 20 (P20)

Port Mode (PMC20_m =0)	Alternative Mode (PMC20_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P20_0	RLIN26RX			PWGA64O	CAN6RX/INTP6		CSIG3SI						CAN6RX				—	—	118	G17	J19
P20_1		RLIN26TX		PWGA65O		CAN6TX		CSIG3SO									—	—	117	H15	J20
P20_2	CAN4RX/INTP4			PWGA66O	RLIN29RX		CSIG3SC						CAN4RX				—	—	116	H16	K20
P20_3		CAN4TX		PWGA67O		RLIN29TX	CSIG3RYI	CSIG3RYO									—	—	115	H17	K18
P20_4	RLIN23RX		INTP22	PWGA59O	CAN7RX/INTP9		$\overline{\text{CSIG3SSI}}$						CAN7RX				—	—	120	F17	H20
	RLIN23RX		INTP22	PWGA59O	CAN7RX/INTP9								CAN7RX				—	100	—	—	—
P20_5		RLIN23TX	INTP23	PWGA60O		CAN7TX											—	99	119	G16	J18
P20_6		PWGA88O															—	—	—	—	G19
P20_7		PWGA89O															—	—	—	—	G20
P20_8		PWGA90O															—	—	—	—	H18
P20_9		PWGA91O															—	—	—	—	H19
P20_10		PWGA92O															—	—	—	—	K19
P20_11		PWGA93O															—	—	—	—	L20
P20_12		PWGA94O															—	—	—	—	L19
P20_13		PWGA95O															—	—	—	—	L18
P20_14																	—	—	—	—	M20

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.14.2 Control Registers

Table 2B.67 Control Registers (P20)

			Effective Bit				Device				
Register	Function	Register Size	Position	R/W^{*1}	Offset Address	Value after Reset	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P20	Port register 20	16	5, 4	R/W	0050$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PSR20	Port set/reset register 20	32	21-20, 5, 4	R/W	0150$_H$	0000 0000$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	—
			30-16, 14-0				—	—	—	—	✓
PPR20	Port pin read register 20	16	5, 4	R	0250$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PM20	Port mode register 20	16	5, 4	R/W	0350$_H$	FFFF$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PMC20	Port mode control register 20	16	5, 4	R/W	0450$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			13-0				—	—	—	—	✓
PFC20	Port function control register 20	16	5, 4	R/W	0550$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PFCE20	Port function control expansion register 20	16	5, 4	R/W	0650$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	✓
PNOT20	Port NOT register 20	16	5, 4	W	0750$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PMSR20	Port mode set/reset register 20	32	21, 20, 5, 4	R/W	0850$_H$	0000 FFFF$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	—
			30-16, 14-0				—	—	—	—	✓
PMCSR20	Port mode control set/reset register 20	32	21, 20, 5, 4	R/W	0950$_H$	0000 0000$_H$	—	✓	—	—	—
			21-16, 5-0				—	—	✓	✓	—
			29-16, 13-0				—	—	—	—	✓
PFCAE20	Port function control additional expansion register 20	16	4	R/W	0A50$_H$	0000$_H$	—	✓	—	—	—
			4, 2, 0				—	—	✓	✓	✓
PIBC20	Port input buffer control register 20	16	5, 4	R/W	4050$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PBDC20	Port bidirection control register 20	16	5, 4	R/W	4150$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PIPC20	Port IP control register 20	16	2, 1	R/W	4250$_H$	0000$_H$	—	—	✓	✓	✓
PU20	Pull-up option register 20	16	5, 4	R/W	4350$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PD20	Pull-down option register 20	16	5, 4	R/W	4450$_H$	0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PODC20	Port open drain control register 20	32	5, 4	R/W	4550$_H$	0000 0000$_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓

Table 2B.67 Control Registers (P20)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
PDSC20	Port drive strength control register 20	32	5, 4	R/W	4650_H	0000 0000_H	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PIS20	Port input buffer selection register 20	16	5, 4	R/W	4750_H	$FFFF_H$	—	✓	—	—	—
			5-0				—	—	✓	✓	—
			14-0				—	—	—	—	✓
PPROTS20	Port protection status register 20	32	0	R	$4B50_H$	0000 0000_H	—	✓	✓	✓	✓
PPCMD20	Port protection command register 20	32	7-0	W	$4C50_H$	xxxx $xx00_H$	—	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.15 Port 21 (P21)

2B.10.15.1 Alternative Function

Table 2B.68 Port 21 (P21)

Port Mode (PMC21_m =0)	Alternative Mode (PMC21_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P21_0																	—	—	—	—	H3
P21_1																	—	—	—	—	B9
P21_2																	—	—	—	—	J3
P21_3																	—	—	—	—	J2
P21_4																	—	—	—	—	H1

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.15.2 Control Registers

Table 2B.69 Control Registers (P21)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P21	Port register 21	16	4-0	R/W	0054_H	0000_H	—	—	—	—	✓
PSR21	Port set/reset register 21	32	20-16, 4-0	R/W	0154_H	$0000\ 0000_H$	—	—	—	—	✓
PPR21	Port pin read register 21	16	4-0	R	0254_H	0000_H	—	—	—	—	✓
PM21	Port mode register 21	16	4-0	R/W	0354_H	$FFFF_H$	—	—	—	—	✓
PNOT21	Port NOT register 21	16	4-0	W	0754_H	0000_H	—	—	—	—	✓
PMSR21	Port mode set/reset register 21	32	20-16, 4-0	R/W	0854_H	$0000\ FFFF_H$	—	—	—	—	✓
PIBC21	Port input buffer control register 21	16	4-0	R/W	4054_H	0000_H	—	—	—	—	✓
PBDC21	Port bidirection control register 21	16	4-0	R/W	4154_H	0000_H	—	—	—	—	✓
PU21	Pull-up option register 21	16	4-0	R/W	4354_H	0000_H	—	—	—	—	✓
PD21	Pull-down option register 21	16	4-0	R/W	4454_H	0000_H	—	—	—	—	✓
PODC21	Port open drain control register 21	32	4-0	R/W	4554_H	$0000\ 0000_H$	—	—	—	—	✓
PIS21	Port input buffer selection register 21	16	4-0	R/W	4754_H	$FFFF_H$	—	—	—	—	✓
PPROTS21	Port protection status register 21	32	0	R	$4B54_H$	$0000\ 0000_H$	—	—	—	—	✓
PPCMD21	Port protection command register 21	32	7-0	W	$4C54_H$	$0000\ 0000_H$	—	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.16 Port 22 (P22)

2B.10.16.1 Alternative Function

Table 2B.70 Port 22 (P22)

Port Mode (PMC22_m =0)	Alternative Mode (PMC22_m = 1)														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P22_0																	—	—	—	—	G1
P22_1																	—	—	—	—	F2
P22_2																	—	—	—	—	F1
P22_3																	—	—	—	—	D2
P22_4																	—	—	—	—	C1
P22_5																	—	—	—	—	C2
P22_6																	—	—	—	—	B1
P22_7																	—	—	—	—	A2
P22_8																	—	—	—	—	B4
P22_9																	—	—	—	—	A4
P22_10																	—	—	—	—	C6
P22_11																	—	—	—	—	B6
P22_12																	—	—	—	—	A6
P22_13																	—	—	—	—	A7
P22_14																	—	—	—	—	C8
P22_15																	—	—	—	—	A8

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2B.10.16.2 Control Registers

Table 2B.71 Control Registers (P22)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P22	Port register 22	16	15-0	R/W	0058$_{H}$	0000$_{H}$	—	—	—	—	✓
PSR22	Port set/reset register 22	32	31-16, 15-0	R/W	0158$_{H}$	0000 0000$_{H}$	—	—	—	—	✓
PPR22	Port pin read register 22	16	15-0	R	0258$_{H}$	0000$_{H}$	—	—	—	—	✓
PM22	Port mode register 22	16	15-0	R/W	0358$_{H}$	FFFF$_{H}$	—	—	—	—	✓
PNOT22	Port NOT register 22	16	15-0	W	0758$_{H}$	0000$_{H}$	—	—	—	—	✓
PMSR22	Port mode set/reset register 22	32	31-16, 15-0	R/W	0858$_{H}$	0000 FFFF$_{H}$	—	—	—	—	✓
PIBC22	Port input buffer control register 22	16	15-0	R/W	4058$_{H}$	0000$_{H}$	—	—	—	—	✓
PBDC22	Port bidirection control register 22	16	15-0	R/W	4158$_{H}$	0000$_{H}$	—	—	—	—	✓
PU22	Pull-up option register 22	16	15-0	R/W	4358$_{H}$	0000$_{H}$	—	—	—	—	✓
PD22	Pull-down option register 22	16	15-0	R/W	4458$_{H}$	0000$_{H}$	—	—	—	—	✓
PODC22	Port open drain control register 22	32	15-0	R/W	4558$_{H}$	0000 0000$_{H}$	—	—	—	—	✓
PIS22	Port input buffer selection register 22	16	15-0	R/W	4758$_{H}$	FFFF$_{H}$	—	—	—	—	✓
PPROTS22	Port protection status register 22	32	0	R	4B58$_{H}$	0000 0000$_{H}$	—	—	—	—	✓
PPCMD22	Port protection command register 22	32	7-0	W	4C58$_{H}$	0000 0000$_{H}$	—	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.17 Analog Port 0 (AP0)

2B.10.17.1 Alternative Function

Table 2B.72 Analog Port 0 (AP0)

	Alternative Mode														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative								
Port Mode	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	ADC	Special Function	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
AP0_0															ADCA0I0		68	90	106	K15	P20
AP0_1															ADCA0I1		67	89	105	L17	P19
AP0_2															ADCA0I2		66	88	104	L16	R20
AP0_3															ADCA0I3		65	87	103	M17	P18
AP0_4															ADCA0I4		64	86	102	L15	R19
AP0_5															ADCA0I5		63	85	101	M16	T20
AP0_6															ADCA0I6		62	84	100	N17	P17
AP0_7															ADCA0I7		61	83	99	N16	R18
AP0_8															ADCA0I8		60	82	98	M15	T19
AP0_9															ADCA0I9		59	81	97	P17	U20
AP0_10															ADCA0I10		58	80	96	P16	T18
AP0_11															ADCA0I11		57	79	95	N15	U19
AP0_12															ADCA0I12		56	78	94	R17	V20
AP0_13															ADCA0I13		55	77	93	P15	U18
AP0_14															ADCA0I14		54	76	92	R16	V19
AP0_15															ADCA0I15		53	75	91	T17	W20

CAUTION

Use ADC functions with their initial settings. For details, see **Table 2B.73, Control Registers (AP0)**.

2B.10.17.2 Control Registers

Table 2B.73 Control Registers (AP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
AP0	Analog port register 0	16	15-0	R/W	$00C8_H$	0000_H	✓	✓	✓	✓	✓
APSR0	Analog port set/reset register 0	32	31-16, 15-0	R/W	$01C8_H$	$0000\ 0000_H$	✓	✓	✓	✓	✓
APPR0	Analog port pin read register 0	16	15-0	R	$02C8_H$	0000_H	✓	✓	✓	✓	✓
APM0	Analog port mode register 0	16	15-0	R/W	$03C8_H$	$FFFF_H$	✓	✓	✓	✓	✓
APNOT0	Analog port NOT register 0	16	15-0	W	$07C8_H$	0000_H	✓	✓	✓	✓	✓
APMSR0	Analog port mode set/reset register 0	32	31-16, 15-0	R/W	$08C8_H$	$0000\ FFFF_H$	✓	✓	✓	✓	✓
APIBC0	Analog port input buffer control register 0	16	15-0	R/W	$40C8_H$	0000_H	✓	✓	✓	✓	✓
APBDC0	Analog port bidirection control register 0	16	15-0	R/W	$41C8_H$	0000_H	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.18 Analog Port 1 (AP1)

2B.10.18.1 Alternative Function

Table 2B.74 Analog Port 1 (AP1)

Port Mode	Alternative Mode														ADC	Special Function	PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
AP1_0															ADCA1I0		—	109	133	C17	C20
AP1_1															ADCA1I1		—	108	132	D15	D18
AP1_2															ADCA1I2		—	107	131	D16	D19
AP1_3															ADCA1I3		—	106	130	D17	D20
AP1_4															ADCA1I4		—	105	129	F14	E18
AP1_5															ADCA1I5		—	104	128	E15	E19
AP1_6															ADCA1I6		—	103	127	E16	E20
AP1_7															ADCA1I7		—	102	126	F15	F18
AP1_8															ADCA1I8		—	—	125	E17	G17
AP1_9															ADCA1I9		—	—	124	F16	F19
AP1_10															ADCA1I10		—	—	123	G14	F20
AP1_11															ADCA1I11		—	—	122	G15	G18
AP1_12															ADCA1I12		—	—	137	B16	B19
AP1_13															ADCA1I13		—	—	136	C15	B20
AP1_14															ADCA1I14		—	—	135	B17	C18
AP1_15															ADCA1I15		—	—	134	C16	C19

CAUTION

Use ADC functions with their initial settings. For details, see **Table 2B.75, Control Registers (AP1)**.

2B.10.18.2 Control Registers

Table 2B.75 Control Registers (AP1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
AP1	Analog port register 1	16	7-0	R/W	00CC$_H$	0000$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓
APSR1	Analog port set/reset register 1	32	23-16, 7-0	R/W	01CC$_H$	0000 0000$_H$	—	✓	—	—	—
			31-16, 15-0				—	—	✓	✓	✓
APPR1	Analog port pin read register 1	16	7-0	R	02CC$_H$	0000$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓
APM1	Analog port mode register 1	16	7-0	R/W	03CC$_H$	FFFF$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓
APNOT1	Analog port NOT register 1	16	7-0	W	07CC$_H$	0000$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓
APMSR1	Analog port mode Set/reset register 1	32	23-16, 7-0	R/W	08CC$_H$	0000 FFFF$_H$	—	✓	—	—	—
			31-16, 15-0				—	—	✓	✓	✓
APIBC1	Analog port input buffer control register 1	16	7-0	R/W	40CC$_H$	0000$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓
APBDC1	Analog port bidirection control register 1	16	7-0	R/W	41CC$_H$	0000$_H$	—	✓	—	—	—
			15-0				—	—	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2B.10.19 Input Port 0 (IP0)

2B.10.19.1 Alternative Function

Table 2B.76 Input Port 0 (IP0)

	Alternative Mode															PKG No.				
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative							
Port Mode	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
IP0_0															XT2	—	47	57	T7	W8

2B.10.19.2 Control Registers

Table 2B.77 Control Registers (IP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device				
			Position	R/W*1			100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
IPPR0	Input port pin read register 0	16	0	R	$02F0_H$	0000_H	—	✓	✓	✓	✓
IPIBC0	Port input buffer control register 0	16	0	R/W	$40F0_H$	0000_H	—	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset

CAUTION

When the IP0_0/XT2 pin is used as an input port, set the IPIBC0.0 bit to 1 and stop the SOSC operation. For details on the settings for SOSC operations, see **Section 12AB.4.2.7, SOSCE — SubOSC Enable Register**. When the IP0_0/XT2 pin is used for the SubOSC (SOSC) not as an input port, set the IPIBC0.0 bit to 0.

2B.11 Port (Special I/O) Function Overview

This section describes the port (special I/O) functions.

2B.11.1 Special I/O after Reset

The special port function after reset is deasserted is shown below.

2B.11.1.1 P8_6: $\overline{\text{RESETOUT}}$

The P8_6 pin ($\overline{\text{RESETOUT}}$ signal) changes PM8.PM8_6 and PODC8.PODC8_6 registers value after reset by OPBT0.RESETOUTEN setting.

The P8_6 pin outputs a low level while a reset is asserted, and pin status of after the reset is different.

(Case 1): OPBT0.RESETOUTEN = 1

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 0: Output mode
- PODC8.PODC8_6 = 1: Open-drain

(Case 2): OPBT0.RESETOUTEN = 0

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 1: Input mode
- PODC8.PODC8_6 = 0: Push-pull

For detail of OPBT0.RESETOUTEN register, see **Section 44.9.2, OPBT0 — Option Byte 0**, also see **Section 9BC.1.3, Reset Output ($\overline{\text{RESETOUT}}$)**.

When the P8_6 pin setting is updated with another value, the pin operates by new setting.

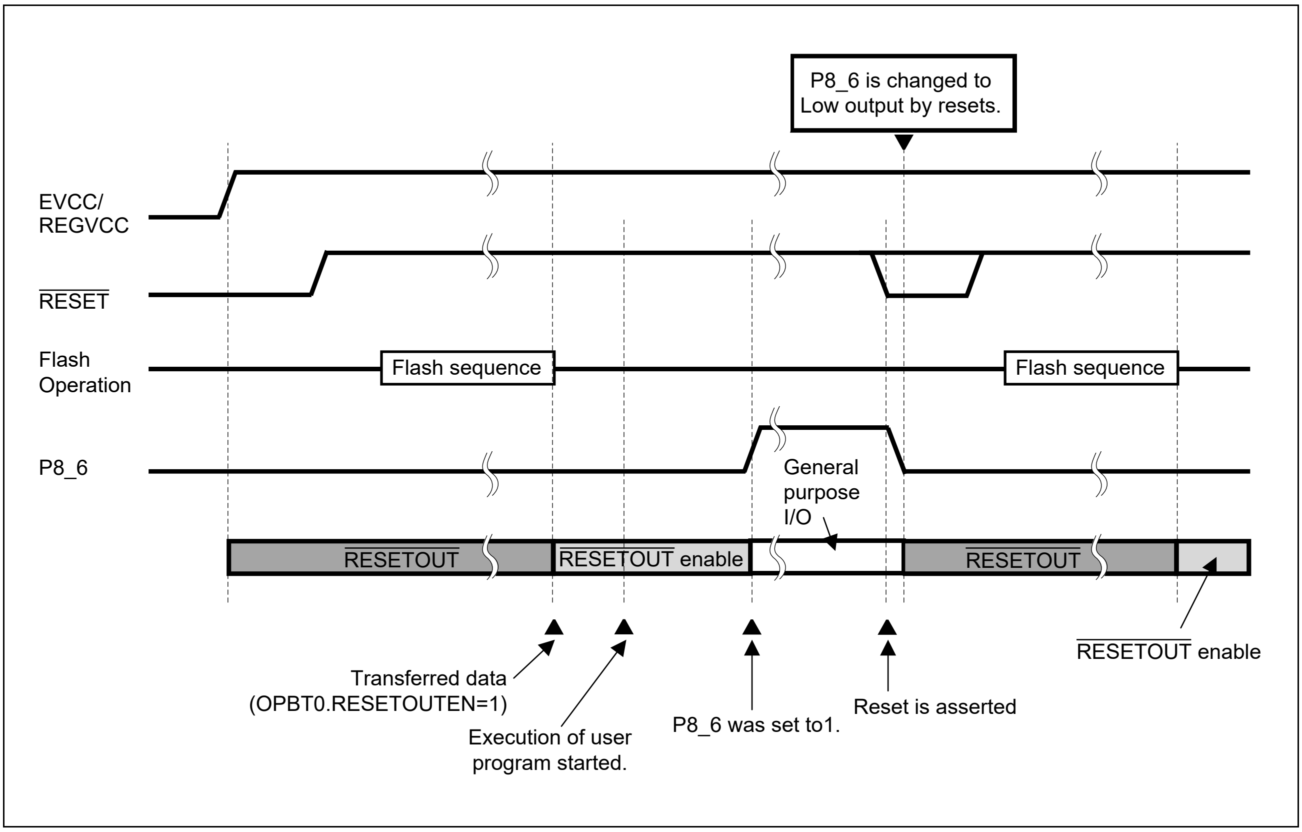
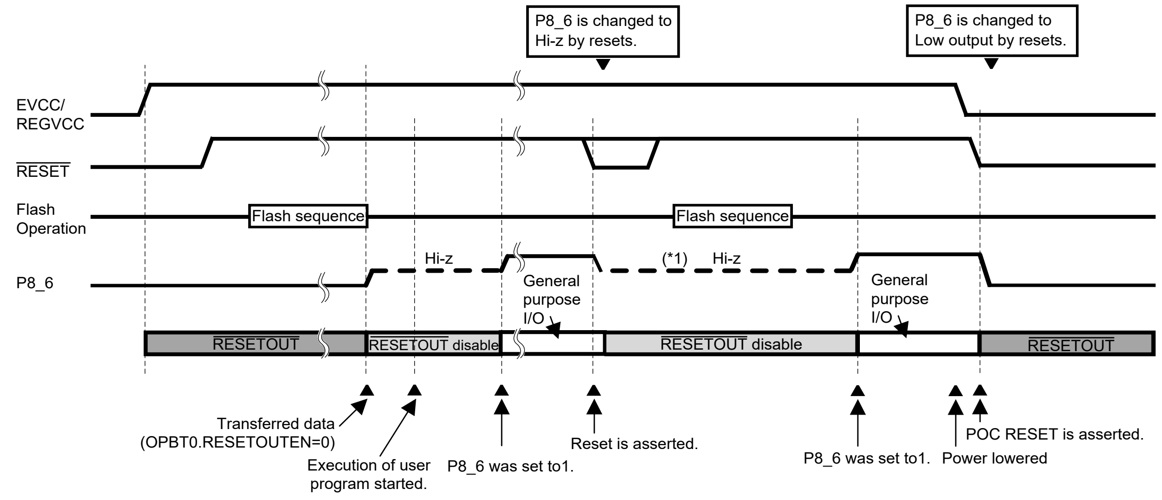


Figure 2B.11 P8_6 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation While a Reset is asserted and released: (Case 1) OPBT0.RESETOUTEN setting is 1



Note 1. When a reset except POC reset occurs with $\overline{\text{RESETOUT}}$ disable (OPBT0.RESETOUTEN = 0), P8_6 pin ($\overline{\text{RESETOUT}}$ signal) will be changed to Hi-z.

Figure 2B.12 P8_6 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation While a Reset is asserted and released: (Case 2) OPBT0.RESETOUTEN setting is 0

2B.11.1.2 JP0_0 to JP0_6: Debug Interface

If the OPJTAG[1:0] setting is the combination below, the pins of the JTAG port group can be used as a debug interface after reset release.

Table 2B.78 Debug Interface

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5	JP0_6
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	$\overline{\text{DCUTRST}}$ input	$\overline{\text{DCURDY}}$ output	$\overline{\text{EVTO}}$ output
0	1	LPD (4 pins)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output	Port/ alternative function
1	0	LPD (1 pin)	LPDIO input/output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

NOTE

For the OPJTAG[1:0] settings, see **Section 44.9.2, OPBT0 — Option Byte 0**.

2B.11.1.3 FPDR(JP0_0), FPDT(JP0_1), FPCK(JP0_2): Flash Programmer

These pins are used for connecting a flash programmer. See *Flash Programmer's Manual* for details.

2B.11.1.4 Mode Pins

The FLMD0 pin in combination with the P10_8: FLMD1 pin can select serial programming mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_2: MODE1 and the P10_1: MODE0 pins can select boundary scan mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_6: MODE2, the P10_2: MODE1 and the P10_1: MODE0 pins can select user boot mode.

For details on the mode selection, see **Section 6, Operating Mode**.

2B.11.1.5 IP0_0: XT2

This pin is the SubOSC (SOSC) input pin. When the IPIBC0_0 bit = 1, the IP0_0/XT2 pin is used as an input port. If you make this setting, stop SOSC operation at the same time.

2B.11.2 A/D Input Alternative I/O

The following ports are permanently connected to A/D input functions. (However, an analog input to the A/D is controlled by the A/D module.)

Table 2B.79 A/D Input Alternative Pins

Port	A/D Input	Device				
		100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P8_0	ADCA0I0S	—	✓	✓	✓	✓
P8_1	ADCA0I1S	—	✓	✓	✓	✓
P8_2	ADCA0I4S	✓	✓	✓	✓	✓
P8_3	ADCA0I5S	✓	✓	✓	✓	✓
P8_4	ADCA0I6S	✓	✓	✓	✓	✓
P8_5	ADCA0I7S	✓	✓	✓	✓	✓
P8_6	ADCA0I8S	✓	✓	✓	✓	✓
P8_7	ADCA0I14S	✓	✓	✓	✓	✓
P8_8	ADCA0I15S	✓	✓	✓	✓	✓
P8_9	ADCA0I16S	✓	✓	✓	✓	✓
P8_10	ADCA0I17S	✓	✓	✓	✓	✓
P8_11	ADCA0I18S	✓	✓	✓	✓	✓
P8_12	ADCA0I19S	✓	✓	✓	✓	✓
P9_0	ADCA0I2S	✓	✓	✓	✓	✓
P9_1	ADCA0I3S	✓	✓	✓	✓	✓
P9_2	ADCA0I9S	✓	✓	✓	✓	✓
P9_3	ADCA0I10S	✓	✓	✓	✓	✓
P9_4	ADCA0I11S	✓	✓	✓	✓	✓
P18_0	ADCA1I0S	—	✓	✓	✓	✓
P18_1	ADCA1I1S	—	✓	✓	✓	✓
P18_2	ADCA1I2S	—	✓	✓	✓	✓
P18_3	ADCA1I3S	—	✓	✓	✓	✓
P18_4	ADCA1I4S	—	—	✓	✓	✓
P18_5	ADCA1I5S	—	—	✓	✓	✓
P18_6	ADCA1I6S	—	—	✓	✓	✓
P18_7	ADCA1I7S	—	—	✓	✓	✓
P18_8	ADCA1I8S	—	—	—	✓	✓
P18_9	ADCA1I9S	—	—	—	✓	✓
P18_10	ADCA1I10S	—	—	—	✓	✓
P18_11	ADCA1I11S	—	—	—	✓	✓
P18_12	ADCA1I12S	—	—	—	✓	✓
P18_13	ADCA1I13S	—	—	—	✓	✓
P18_14	ADCA1I14S	—	—	—	✓	✓
P18_15	ADCA1I15S	—	—	—	✓	✓
P19_0	ADCA1I16S	—	—	—	✓	✓
P19_1	ADCA1I17S	—	—	—	✓	✓
P19_2	ADCA1I18S	—	—	—	✓	✓
P19_3	ADCA1I19S	—	—	—	✓	✓

Table 2B.79 A/D Input Alternative Pins

Port	A/D Input	Device				
		100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
AP0_0	ADCA0I0	✓	✓	✓	✓	✓
AP0_1	ADCA0I1	✓	✓	✓	✓	✓
AP0_2	ADCA0I2	✓	✓	✓	✓	✓
AP0_3	ADCA0I3	✓	✓	✓	✓	✓
AP0_4	ADCA0I4	✓	✓	✓	✓	✓
AP0_5	ADCA0I5	✓	✓	✓	✓	✓
AP0_6	ADCA0I6	✓	✓	✓	✓	✓
AP0_7	ADCA0I7	✓	✓	✓	✓	✓
AP0_8	ADCA0I8	✓	✓	✓	✓	✓
AP0_9	ADCA0I9	✓	✓	✓	✓	✓
AP0_10	ADCA0I10	✓	✓	✓	✓	✓
AP0_11	ADCA0I11	✓	✓	✓	✓	✓
AP0_12	ADCA0I12	✓	✓	✓	✓	✓
AP0_13	ADCA0I13	✓	✓	✓	✓	✓
AP0_14	ADCA0I14	✓	✓	✓	✓	✓
AP0_15	ADCA0I15	✓	✓	✓	✓	✓
AP1_0	ADCA1I0	—	✓	✓	✓	✓
AP1_1	ADCA1I1	—	✓	✓	✓	✓
AP1_2	ADCA1I2	—	✓	✓	✓	✓
AP1_3	ADCA1I3	—	✓	✓	✓	✓
AP1_4	ADCA1I4	—	✓	✓	✓	✓
AP1_5	ADCA1I5	—	✓	✓	✓	✓
AP1_6	ADCA1I6	—	✓	✓	✓	✓
AP1_7	ADCA1I7	—	✓	✓	✓	✓
AP1_8	ADCA1I8	—	—	✓	✓	✓
AP1_9	ADCA1I9	—	—	✓	✓	✓
AP1_10	ADCA1I10	—	—	✓	✓	✓
AP1_11	ADCA1I11	—	—	✓	✓	✓
AP1_12	ADCA1I12	—	—	✓	✓	✓
AP1_13	ADCA1I13	—	—	✓	✓	✓
AP1_14	ADCA1I14	—	—	✓	✓	✓
AP1_15	ADCA1I15	—	—	✓	✓	✓

2B.11.3 Special I/O Control

2B.11.3.1 Direct I/O Control (PIPC)

Some alternative functions take over the input and output control of the ports.

The following table lists all alternative functions where PIPCn.PIPCn_m must be set to 1.

For details, see **Section 2B.9.2.3, PIPCn — Port IP Control Register**.

Table 2B.80 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1

Function	Alternative functions Name	Port Name	Power Supply Area	Control	Reference Section
MEMC	MEMC0AD0	P10_6	ISO		**Section 16**
	MEMC0AD1	P10_7	ISO		
	MEMC0AD2	P10_8	ISO		
	MEMC0AD3	P10_9	ISO		
	MEMC0AD4	P10_10	ISO		
	MEMC0AD5	P10_11	ISO		
	MEMC0AD6	P10_12	ISO		
	MEMC0AD7	P10_13	ISO		
	MEMC0AD8	P10_14	ISO		
	MEMC0AD9	P11_1	ISO		
	MEMC0AD10	P11_2	ISO		
	MEMC0AD11	P11_3	ISO		
	MEMC0AD12	P11_4	ISO		
	MEMC0AD13	P11_5	ISO		
	MEMC0AD14	P11_6	ISO		
	MEMC0AD15	P11_7	ISO		
TAPA	TAPA0UP	P10_0	ISO	U phase Hi-Z control	**Section 36**
	TAPA0UN	P10_1	ISO		
	TAPA0VP	P10_2	ISO	V phase Hi-Z control	
	TAPA0VN	P10_3	ISO		
	TAPA0WP	P10_4	ISO	W phase Hi-Z control	
	TAPA0WN	P10_5	ISO		
CSIG	CSIG0SO	P0_13	AWO	Serial data output control signal	**Section 19**
		P10_6	ISO		
	CSIG0SC	P0_14	AWO	Master (1) / slave (0) mode signal	
		P10_7	ISO		
	CSIG1SO	P11_9	ISO	Serial data output control signal	
	CSIG1SC	P11_10	ISO	Master (1) / slave (0) mode signal	
	CSIG2SO	P12_5	ISO	Serial data output control signal	
	CSIG2SC	P12_4	ISO	Master (1) / slave (0) mode signal	
	CSIG3SO	P20_1	ISO	Serial data output control signal	
	CSIG3SC	P20_2	ISO	Master (1) / slave (0) mode signal	

Table 2B.80 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1

Function	Alternative functions Name	Port Name	Power Supply Area	Control	Reference Section
CSIH	CSIH0SO	P0_3	AWO	Serial data output control signal	**Section 20**
	CSIH0SC	P0_2	AWO	Master (1) / slave (0) mode signal	
	CSIH1SO	P0_5	AWO	Serial data output control signal	
		P10_2	ISO		
	CSIH1SC	P0_6	AWO	Master (1) / slave (0) mode signal	
		P10_1	ISO		
	CSIH2SO	P11_2	ISO	Serial data output control signal	
	CSIH2SC	P11_3	ISO	Master (1) / slave (0) mode signal	
	CSIH3SO	P11_6	ISO	Serial data output control signal	
	CSIH3SC	P11_7	ISO	Master (1) / slave (0) mode signal	
SFMA	SFMA0IO0	P11_5	ISO	SPIch.0 MOSI0_IO00 output enable	**Section 17**
	SFMA0IO1	P11_4	ISO	SPIch.0 MOSI0_IO10 output enable	
	SFMA0IO2	P11_3	ISO	SPIch.0 IO20 output enable	
	SFMA0IO3	P11_2	ISO	SPIch.0 IO30 output enable	
ETNB	ETNB0MDIO	P12_4	ISO	MDIO output enable	**Section 26**

2B.11.3.2 Input Buffer Control (PISn/JPIS0, PISAn/JPISA0)

The port input buffer characteristics (Type 1 or Type 2) of this device can be selected using the PISn/PISAn/JPIS0 register. The applicable pins are shown in the following table.

The JTAG port input buffer characteristics (Type 1/2 or Type 5) of this device can be selected using the JPISA0 register. The applicable pins are shown in **Table 2B.82, JTAG Port Input Buffer Characteristics Selection**.

Table 2B.81 Port Input Buffer Characteristics Selection

	Input Buffer Selection			Device				
Port Name	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P0_0	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_1	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_2	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_4	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_5	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_6	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_7	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_8	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_9	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_10	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_11	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_12	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_13	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P0_14	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P1_0	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_1	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_2	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_3	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_4	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_5	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_8	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_9	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_10	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_11	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P1_12	SHMT1	SHMT4	—	—	—	✓	✓	✓
P1_13	SHMT1	SHMT4	—	—	—	✓	✓	✓
P1_14	SHMT1	SHMT4	—	—	—	✓	✓	✓
P1_15	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_0	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_1	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_2	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_3	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_4	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_5	SHMT1	SHMT4	—	—	—	✓	✓	✓
P2_6	SHMT1	SHMT4	—	—	—	✓	✓	✓

Table 2B.81 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device				
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P2_7	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_8	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_9	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_10	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_11	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_12	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_13	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_14	SHMT1	SHMT4	—	—	—	—	✓	✓
P2_15	SHMT1	SHMT4	—	—	—	—	✓	✓
P3_0	SHMT1	SHMT4	—	—	—	—	✓	✓
P3_1	SHMT1	SHMT4	—	—	—	—	—	✓
P3_2	SHMT1	SHMT4	—	—	—	—	—	✓
P3_3	SHMT1	SHMT4	—	—	—	—	—	✓
P3_4	SHMT1	SHMT4	—	—	—	—	—	✓
P3_5	SHMT1	SHMT4	—	—	—	—	—	✓
P3_6	SHMT1	SHMT4	—	—	—	—	—	✓
P3_7	SHMT1	SHMT4	—	—	—	—	—	✓
P3_8	SHMT1	SHMT4	—	—	—	—	—	✓
P3_9	SHMT1	SHMT4	—	—	—	—	—	✓
P3_10	SHMT1	SHMT4	—	—	—	—	—	✓
P8_0	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P8_1	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P8_2	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_4	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_5	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_6	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_7	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_8	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_9	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_10	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_11	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P8_12	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P9_0	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P9_1	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P9_2	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P9_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P9_4	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_0	SHMT1	SHMT4	—	✓	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P10_1	SHMT1	SHMT4	—	✓	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓

Table 2B.81 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device				
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P10_2	SHMT1	SHMT4	—	✓	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P10_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_4	SHMT1	SHMT4	—	✓	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P10_5	SHMT1	SHMT4	—	✓	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P10_6	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_7	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_8	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_9	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_10	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_11	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_12	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_13	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_14	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P10_15	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P11_0	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P11_1	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_2	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_4	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_5	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_6	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_7	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
P11_8	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P11_9	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P11_10	SHMT1	SHMT4	—	—	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P11_11	SHMT1	SHMT4	—	—	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P11_12	SHMT1	SHMT4	—	—	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P11_15	SHMT1	SHMT4	—	—	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P12_0	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P12_1	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P12_2	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P12_3	SHMT1	SHMT4	—	—	—	✓	✓	✓
P12_4	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P12_5	SHMT1	SHMT4	—	—	—	✓	✓	✓
P13_0	SHMT1	SHMT4	—	—	—	—	✓	✓
P13_1	SHMT1	SHMT4	—	—	—	—	✓	✓

Table 2B.81 Port Input Buffer Characteristics Selection

	Input Buffer Selection			Device				
Port Name	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P13_2	SHMT1	SHMT4	TTL	—	—	—	✓	✓
P13_3	SHMT1	SHMT4	TTL	—	—	—	✓	✓
P13_4	SHMT1	SHMT4	—	—	—	—	✓	✓
P13_5	SHMT1	SHMT4	TTL	—	—	—	✓	✓
P13_6	SHMT1	SHMT4	—	—	—	—	✓	✓
P13_7	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_0	SHMT1	SHMT4	—	—	✓	—	—	—
	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P18_1	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P18_2	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P18_3	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P18_4	SHMT1	SHMT4	—	—	—	✓	✓	✓
P18_5	SHMT1	SHMT4	—	—	—	✓	✓	✓
P18_6	SHMT1	SHMT4	—	—	—	✓	✓	✓
P18_7	SHMT1	SHMT4	TTL	—	—	✓	✓	✓
P18_8	SHMT1	SHMT4	TTL	—	—	—	✓	✓
P18_9	SHMT1	SHMT4	TTL	—	—	—	✓	✓
P18_10	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_11	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_12	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_13	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_14	SHMT1	SHMT4	—	—	—	—	✓	✓
P18_15	SHMT1	SHMT4	—	—	—	—	✓	✓
P19_0	SHMT1	SHMT4	—	—	—	—	✓	✓
P19_1	SHMT1	SHMT4	—	—	—	—	✓	✓
P19_2	SHMT1	SHMT4	—	—	—	—	✓	✓
P19_3	SHMT1	SHMT4	—	—	—	—	✓	✓
P20_0	SHMT1	SHMT4	—	—	—	✓	✓	✓
P20_1	SHMT1	SHMT4	—	—	—	✓	✓	✓
P20_2	SHMT1	SHMT4	—	—	—	✓	✓	✓
P20_3	SHMT1	SHMT4	—	—	—	✓	✓	✓
P20_4	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P20_5	SHMT1	SHMT4	—	—	✓	✓	✓	✓
P20_6	SHMT1	SHMT4	—	—	—	—	—	✓
P20_7	SHMT1	SHMT4	—	—	—	—	—	✓
P20_8	SHMT1	SHMT4	—	—	—	—	—	✓
P20_9	SHMT1	SHMT4	—	—	—	—	—	✓
P20_10	SHMT1	SHMT4	—	—	—	—	—	✓
P20_11	SHMT1	SHMT4	—	—	—	—	—	✓
P20_12	SHMT1	SHMT4	—	—	—	—	—	✓
P20_13	SHMT1	SHMT4	—	—	—	—	—	✓
P20_14	SHMT1	SHMT4	—	—	—	—	—	✓
P21_0	SHMT1	SHMT4	—	—	—	—	—	✓

Table 2B.81 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Device				
	Type 1 (PISn_m = 0 & PISAn_m = 0)	Type 2 (PISn_m = 1 & PISAn_m = 0)	Type 5 (PISAn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P21_1	SHMT1	SHMT4	—	—	—	—	—	✓
P21_2	SHMT1	SHMT4	—	—	—	—	—	✓
P21_3	SHMT1	SHMT4	—	—	—	—	—	✓
P21_4	SHMT1	SHMT4	—	—	—	—	—	✓
P22_0	SHMT1	SHMT4	—	—	—	—	—	✓
P22_1	SHMT1	SHMT4	—	—	—	—	—	✓
P22_2	SHMT1	SHMT4	—	—	—	—	—	✓
P22_3	SHMT1	SHMT4	—	—	—	—	—	✓
P22_4	SHMT1	SHMT4	—	—	—	—	—	✓
P22_5	SHMT1	SHMT4	—	—	—	—	—	✓
P22_6	SHMT1	SHMT4	—	—	—	—	—	✓
P22_7	SHMT1	SHMT4	—	—	—	—	—	✓
P22_8	SHMT1	SHMT4	—	—	—	—	—	✓
P22_9	SHMT1	SHMT4	—	—	—	—	—	✓
P22_10	SHMT1	SHMT4	—	—	—	—	—	✓
P22_11	SHMT1	SHMT4	—	—	—	—	—	✓
P22_12	SHMT1	SHMT4	—	—	—	—	—	✓
P22_13	SHMT1	SHMT4	—	—	—	—	—	✓
P22_14	SHMT1	SHMT4	—	—	—	—	—	✓
P22_15	SHMT1	SHMT4	—	—	—	—	—	✓

Table 2B.82 JTAG Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Devices				
	Type 1 (JPIS0_m = 0 & JPISA0_m = 0)	Type 2 (JPIS0_m = 1 & JPISA0_m = 0)	Type 5 (JPISA0_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
JP0_0	SHMT1	SHMT4	TTL*1,*2,*3,*4	✓	✓	✓	✓	✓
JP0_1	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
JP0_2	SHMT1	SHMT4	TTL*1,*2,*3	✓	✓	✓	✓	✓
JP0_3	SHMT1	SHMT4	TTL*1,*2	✓	✓	✓	✓	✓
JP0_4	—	SHMT4	—*1,*2	✓	✓	✓	✓	✓
JP0_5	SHMT1	SHMT4	—	✓	✓	✓	✓	✓
JP0_6	SHMT1	SHMT4	—	—	✓	✓	✓	✓

Note 1. TTL is selected for Boundary scan mode without JPISA0 register setting.

Note 2. TTL is selected for Nexus in normal operating mode without JPISA0 register setting.

Note 3. TTL is selected for LPD (4 pins) in normal operating mode without JPISA0 register setting.

Note 4. TTL is selected for LPD (1 pin) in normal operating mode without JPISA0 register setting.

NOTES

1. For the SHMT1, SHMT4 and TTL pin characteristics, see **Section 47B, Electrical Characteristics of RH850/F1KM-S4**.
2. For the input buffer after reset, Type 2 (SHMT4) is selected.

2B.11.3.3 Output Buffer Control (PDSC)

The port output driver strength (slow mode/fast mode) can be selected using the PDSCn register. The applicable pins are shown in the following table. Only slow mode is supported for ports other than those listed below.

Table 2B.83 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device				
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
JP0_1	10 MHz	40 MHz	✓	✓	✓	✓	✓
JP0_2	10 MHz	40 MHz	✓	✓	✓	✓	✓
JP0_3	10 MHz	40 MHz	✓	✓	✓	✓	✓
JP0_5	10 MHz	40 MHz	✓	✓	✓	✓	✓
JP0_6	10 MHz	40 MHz	—	✓	✓	✓	✓
P0_0	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_1	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_2	10 MHz	40 MHz*1	✓	✓	✓	✓	✓
P0_3	10 MHz	40 MHz*1	✓	✓	✓	✓	✓
P0_4	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_5	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P0_6	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P0_7	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_8	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_9	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_10	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_11	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_12	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_13	10 MHz	40 MHz	✓	✓	✓	✓	✓
P0_14	10 MHz	40 MHz	✓	✓	✓	✓	✓
P1_0	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_1	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_2	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_3	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_4	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_5	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_8	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_9	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_10	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_11	10 MHz	40 MHz	—	✓	✓	✓	✓
P1_12	10 MHz	40 MHz	—	—	✓	✓	✓
P1_13	10 MHz	40 MHz	—	—	✓	✓	✓
P1_14	10 MHz	40 MHz	—	—	✓	✓	✓
P1_15	10 MHz	40 MHz	—	—	✓	✓	✓
P2_0	10 MHz	40 MHz	—	—	✓	✓	✓
P2_1	10 MHz	40 MHz	—	—	✓	✓	✓
P2_2	10 MHz	40 MHz	—	—	✓	✓	✓
P2_3	10 MHz	40 MHz	—	—	✓	✓	✓

Table 2B.83 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device				
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P2_4	10 MHz	40 MHz	—	—	✓	✓	✓
P2_5	10 MHz	40 MHz	—	—	✓	✓	✓
P2_6	10 MHz	40 MHz	—	—	✓	✓	✓
P2_7	10 MHz	40 MHz	—	—	—	✓	✓
P2_8	10 MHz	40 MHz	—	—	—	✓	✓
P2_9	10 MHz	40 MHz	—	—	—	✓	✓
P2_10	10 MHz	40 MHz	—	—	—	✓	✓
P2_11	10 MHz	40 MHz	—	—	—	✓	✓
P2_12	10 MHz	40 MHz	—	—	—	✓	✓
P2_13	10 MHz	40 MHz	—	—	—	✓	✓
P2_14	10 MHz	40 MHz	—	—	—	✓	✓
P2_15	10 MHz	40 MHz	—	—	—	✓	✓
P3_0	10 MHz	40 MHz	—	—	—	✓	✓
P3_1	10 MHz	40 MHz	—	—	—	—	✓
P3_2	10 MHz	40 MHz	—	—	—	—	✓
P3_3	10 MHz	40 MHz	—	—	—	—	✓
P3_4	10 MHz	40 MHz	—	—	—	—	✓
P3_5	10 MHz	40 MHz	—	—	—	—	✓
P3_6	10 MHz	40 MHz	—	—	—	—	✓
P3_7	10 MHz	40 MHz	—	—	—	—	✓
P3_8	10 MHz	40 MHz	—	—	—	—	✓
P3_9	10 MHz	40 MHz	—	—	—	—	✓
P3_10	10 MHz	40 MHz	—	—	—	—	✓
P10_0	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_1	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P10_2	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P10_3	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_4	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_5	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_6	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_7	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_8	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_9	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_10	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_11	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_12	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_13	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_14	10 MHz	40 MHz	✓	✓	✓	✓	✓
P10_15	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_0	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_1	10 MHz	40 MHz	✓	✓	✓	✓	✓
P11_2	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P11_3	10 MHz	40 MHz*2	✓	✓	✓	✓	✓

Table 2B.83 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device				
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P11_4	10 MHz	40 MHz	✓	✓	✓	✓	✓
P11_5	10 MHz	40 MHz	✓	✓	✓	✓	✓
P11_6	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P11_7	10 MHz	40 MHz*2	✓	✓	✓	✓	✓
P11_8	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_9	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_10	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_11	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_12	10 MHz	40 MHz	—	✓	✓	✓	✓
P11_15	10 MHz	40 MHz	—	✓	✓	✓	✓
P12_0	10 MHz	40 MHz	—	✓	✓	✓	✓
P12_1	10 MHz	40 MHz	—	✓	✓	✓	✓
P12_2	10 MHz	40 MHz	—	✓	✓	✓	✓
P12_3	10 MHz	40 MHz	—	—	✓	✓	✓
P12_4	10 MHz	40 MHz	—	—	✓	✓	✓
P12_5	10 MHz	40 MHz	—	—	✓	✓	✓
P13_0	10 MHz	40 MHz	—	—	—	✓	✓
P13_1	10 MHz	40 MHz	—	—	—	✓	✓
P13_2	10 MHz	40 MHz	—	—	—	✓	✓
P13_3	10 MHz	40 MHz	—	—	—	✓	✓
P13_4	10 MHz	40 MHz	—	—	—	✓	✓
P13_5	10 MHz	40 MHz	—	—	—	✓	✓
P13_6	10 MHz	40 MHz	—	—	—	✓	✓
P13_7	10 MHz	40 MHz	—	—	—	✓	✓
P18_0	10 MHz	40 MHz	—	✓	✓	✓	✓
P18_1	10 MHz	40 MHz	—	✓	✓	✓	✓
P18_2	10 MHz	40 MHz	—	✓	✓	✓	✓
P18_3	10 MHz	40 MHz	—	✓	✓	✓	✓
P18_4	10 MHz	40 MHz	—	—	✓	✓	✓
P18_5	10 MHz	40 MHz	—	—	✓	✓	✓
P18_6	10 MHz	40 MHz	—	—	✓	✓	✓
P18_7	10 MHz	40 MHz	—	—	✓	✓	✓
P18_8	10 MHz	40 MHz	—	—	—	✓	✓
P18_9	10 MHz	40 MHz	—	—	—	✓	✓
P18_10	10 MHz	40 MHz	—	—	—	✓	✓
P18_11	10 MHz	40 MHz	—	—	—	✓	✓
P18_12	10 MHz	40 MHz	—	—	—	✓	✓
P18_13	10 MHz	40 MHz	—	—	—	✓	✓
P18_14	10 MHz	40 MHz	—	—	—	✓	✓
P18_15	10 MHz	40 MHz	—	—	—	✓	✓
P19_0	10 MHz	40 MHz	—	—	—	✓	✓
P19_1	10 MHz	40 MHz	—	—	—	✓	✓
P19_2	10 MHz	40 MHz	—	—	—	✓	✓
P19_3	10 MHz	40 MHz	—	—	—	✓	✓

Table 2B.83 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device				
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
P20_0	10 MHz	40 MHz	—	—	✓	✓	✓
P20_1	10 MHz	40 MHz	—	—	✓	✓	✓
P20_2	10 MHz	40 MHz	—	—	✓	✓	✓
P20_3	10 MHz	40 MHz	—	—	✓	✓	✓
P20_4	10 MHz	40 MHz	—	✓	✓	✓	✓
P20_5	10 MHz	40 MHz	—	✓	✓	✓	✓
P20_6	10 MHz	40 MHz	—	—	—	—	✓
P20_7	10 MHz	40 MHz	—	—	—	—	✓
P20_8	10 MHz	40 MHz	—	—	—	—	✓
P20_9	10 MHz	40 MHz	—	—	—	—	✓
P20_10	10 MHz	40 MHz	—	—	—	—	✓
P20_11	10 MHz	40 MHz	—	—	—	—	✓
P20_12	10 MHz	40 MHz	—	—	—	—	✓
P20_13	10 MHz	40 MHz	—	—	—	—	✓
P20_14	10 MHz	40 MHz	—	—	—	—	✓

Note 1. Supports Cload: 100 pF (The load capacitance of CSIH0 is 100 pF.)

Note 2. Supports Cload: 50 pF (The load capacitance of CSIH1 to CSIH3 are 50 pF.)

Note 3. In some of the functions, Fast mode or Slow mode is specified. For details, see **Section 47B.5, AC Characteristics**.

2B.12 Noise Filter & Edge/Level Detector

The input signals at some pins are passed through a filter to remove noise and glitches. The RH850/F1KM supports both analog and digital filters.

It also supports the function for edge and level detection after the signals have passed through a filter.

The first part of this section provides an overview of port input pins that are equipped with a filter and the filter type, noise filter & edge/level detection control registers and control bits, and register addresses.

For details on the digital/analog filter function and noise filter & edge/level detection control registers, see **Section 2B.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2B.12.1 Port Filter Assignment

A list of the input pins that incorporate an analog or digital filter is provided below.

2B.12.1.1 Input Pins that Incorporate Analog Filter Type A

The input pins of analog filter type A incorporate an analog filter and edge/level detection function. Edge/level detection is controlled by the following registers.

- Filter control register FCLA0CTLm_<*name*> (m = 0 to 7)
  A dedicated FCLA0CTLm_<*name*> register is provided for each pin in a port that incorporates an analog filter.

Table 2B.84 Input Pins that Incorporate Analog Filter Type A

		FCLA0CTL Register Configuration		Device				
Module Name	Input Pin	Register	Address	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
FCLA0	NMI	FCLA0CTL0_NMI	FFC3 4000_H	✓	✓	✓	✓	✓
	INTP0	FCLA0CTL0_INTPL	FFC3 4020_H	✓	✓	✓	✓	✓
	INTP1	FCLA0CTL1_INTPL	FFC3 4024_H	✓	✓	✓	✓	✓
	INTP2	FCLA0CTL2_INTPL	FFC3 4028_H	✓	✓	✓	✓	✓
	INTP3	FCLA0CTL3_INTPL	FFC3 $402C_H$	✓	✓	✓	✓	✓
	INTP4	FCLA0CTL4_INTPL	FFC3 4030_H	✓	✓	✓	✓	✓
	INTP5	FCLA0CTL5_INTPL	FFC3 4034_H	✓	✓	✓	✓	✓
	INTP6	FCLA0CTL6_INTPL	FFC3 4038_H	✓	✓	✓	✓	✓
	INTP7	FCLA0CTL7_INTPL	FFC3 $403C_H$	✓	✓	✓	✓	✓
	INTP8	FCLA0CTL0_INTPH	FFC3 4040_H	✓	✓	✓	✓	✓
	INTP9	FCLA0CTL1_INTPH	FFC3 4044_H	✓	✓	✓	✓	✓
	INTP10	FCLA0CTL2_INTPH	FFC3 4048_H	✓	✓	✓	✓	✓
	INTP11	FCLA0CTL3_INTPH	FFC3 $404C_H$	✓	✓	✓	✓	✓
	INTP12	FCLA0CTL4_INTPH	FFC3 4050_H	✓	✓	✓	✓	✓
	INTP13	FCLA0CTL5_INTPH	FFC3 4054_H	✓	✓	✓	✓	✓
	INTP14	FCLA0CTL6_INTPH	FFC3 4058_H	—	✓	✓	✓	✓
	INTP15	FCLA0CTL7_INTPH	FFC3 $405C_H$	—	✓	✓	✓	✓
	INTP16	FCLA0CTL0_INTPU	FFC3 $40A0_H$	—	✓	✓	✓	✓
	INTP17	FCLA0CTL1_INTPU	FFC3 $40A4_H$	—	✓	✓	✓	✓
	INTP18	FCLA0CTL2_INTPU	FFC3 $40A8_H$	—	✓	✓	✓	✓
	INTP19	FCLA0CTL3_INTPU	FFC3 $40AC_H$	—	✓	✓	✓	✓

Table 2B.84 Input Pins that Incorporate Analog Filter Type A

		FCLA0CTL Register Configuration		Device				
Module Name	Input Pin	Register	Address	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
FCLA0	INTP20	FCLA0CTL4_INTPU	FFC3 $40B0_H$	—	✓	✓	✓	✓
	INTP21	FCLA0CTL5_INTPU	FFC3 $40B4_H$	—	✓	✓	✓	✓
	INTP22	FCLA0CTL6_INTPU	FFC3 $40B8_H$	—	✓	✓	✓	✓
	INTP23	FCLA0CTL7_INTPU	FFC3 $40BC_H$	—	✓	✓	✓	✓

2B.12.1.2 Input Pins that Incorporate Analog Filter Type B

The input pins of analog filter type B incorporate an analog filter. Edge/level detection is controlled by the registers for individual peripheral functions.

Table 2B.85 Input Pins that Incorporate Analog Filter Type B

		Device				
Input Pin	Edge/Level Detection	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
TAUJ0I0	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ0I1	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ0I2	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ0I3	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ1I0	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ1I1	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ1I2	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ1I3	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ2I0	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ2I1	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ2I2	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ2I3	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ3I0	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ3I1	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ3I2	Edge detection*[1]	✓	✓	✓	✓	✓
TAUJ3I3	Edge detection*[1]	✓	✓	✓	✓	✓
TAPA0ESO	Edge detection*[2]	—	✓	✓	✓	✓
KR0I0	Low level detection	✓	✓	✓	✓	✓
KR0I1	Low level detection	✓	✓	✓	✓	✓
KR0I2	Low level detection	✓	✓	✓	✓	✓
KR0I3	Low level detection	✓	✓	✓	✓	✓
KR0I4	Low level detection	✓	✓	✓	✓	✓
KR0I5	Low level detection	✓	✓	✓	✓	✓
KR0I6	Low level detection	✓	✓	✓	✓	✓
KR0I7	Low level detection	✓	✓	✓	✓	✓

Note 1. For details on edge detection for TAUJ, see **Section 33.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register**.

Note 2. For details on edge detection for TAPA, see **Section 36.3.2, TAPAnCTL0 — TAPA Control Register 0**.

2B.12.1.3 Input Pins that Incorporate Analog Filter Type C

The input pins of analog filter type C only incorporate an analog filter function.

Table 2B.86 Input Pins that Incorporate Analog Filter Type C

Input Pin
FLMD0
FLMD1
MODE0
MODE1
MODE2
$\overline{\text{RESET}}$
$\overline{\text{DCUTRST}}$

2B.12.1.4 Input Pins that Incorporate Digital Filter Type D

The input pins of digital filter type D incorporate a digital filter and edge detection function. The digital filter and edge detection are controlled by the following registers.

- Filter control register FCLA0CTLm_<*name*> (m = 0 to 2)
  Each port with a digital filter has a special FCLA0CTLm_<*name*> register.
- Digital noise elimination control register DNFA<*name*>CTL
  Each DNFA<*name*>CTL control register controls digital filter processing for three input signals per group.
- Digital noise elimination enable register DNFA<*name*>EN
  The setting of the DNFA<*name*>ENL[2:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for three input signals per group.

Table 2B.87 Input Pins that Incorporate Digital Filter Type D

	Device					Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Filter Control Register	
Input Pin	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Control Register	Address	Control Register	Control Bit	Address	Control Register	Address
ADCA0TRG0	✓	✓	✓	✓	✓	DNFAADCTL0CTL	FFC3 00A0$_H$	DNFAADCTL0EN (DNFAADCTL0ENL)	DNFAADCTL0ENL0	FFC3 00A4$_H$ (FFC3 00AC$_H$)	FCLA0CTL0_ADC0	FFC3 4060$_H$
ADCA0TRG1	✓	✓	✓	✓	✓				DNFAADCTL0ENL1		FCLA0CTL1_ADC0	FFC3 4064$_H$
ADCA0TRG2	✓	✓	✓	✓	✓				DNFAADCTL0ENL2		FCLA0CTL2_ADC0	FFC3 4068$_H$
ADCA1TRG0	—	✓	✓	✓	✓	DNFAADCTL1CTL	FFC3 00C0$_H$	DNFAADCTL1EN (DNFAADCTL1ENL)	DNFAADCTL1ENL0	FFC3 00C4$_H$ (FFC3 00CC$_H$)	FCLA0CTL0_ADC1	FFC3 4080$_H$
ADCA1TRG1	—	✓	✓	✓	✓				DNFAADCTL1ENL1		FCLA0CTL1_ADC1	FFC3 4084$_H$
ADCA1TRG2	—	✓	✓	✓	✓				DNFAADCTL1ENL2		FCLA0CTL2_ADC1	FFC3 4088$_H$

2B.12.1.5 Input Pins that Incorporate Digital Filter Type E

The input pins of digital filter type E incorporate a digital filter. The digital filter is controlled by the following registers. Edge detection is controlled by the registers for individual peripheral functions.

- Digital noise elimination control register DNFA<*name*>CTL
  Each DNFA<*name*>CTL control register controls digital filter processing for up to 16 input signals per group.
- Digital noise elimination enable register DNFA<*name*>EN
  The setting of the DNFA<*name*>ENL[7:0] and DNFA<*name*>ENH[7:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for up to 16 input signals per group.

Table 2B.88 Input Pins that Incorporate Digital Filter Type E

	Devises					Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
Input Pin	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUD0I0	✓	✓	✓	✓	✓	DNFATAUD0ICTL	FFC3 0000$_H$	DNFATAUD0IEN (DNFATAUD0IENH/ DNFATAUD0IENL)	DNFATAUD0IENL0	FFC3 0004$_H$ (FFC3 0008$_H$/ FFC3 000C$_H$)	*1
TAUD0I1	✓	✓	✓	✓	✓				DNFATAUD0IENL1		
TAUD0I2	✓	✓	✓	✓	✓				DNFATAUD0IENL2		
TAUD0I3	✓	✓	✓	✓	✓				DNFATAUD0IENL3		
TAUD0I4	✓	✓	✓	✓	✓				DNFATAUD0IENL4		
TAUD0I5	✓	✓	✓	✓	✓				DNFATAUD0IENL5		
TAUD0I6	✓	✓	✓	✓	✓				DNFATAUD0IENL6		
TAUD0I7	✓	✓	✓	✓	✓				DNFATAUD0IENL7		
TAUD0I8	✓	✓	✓	✓	✓				DNFATAUD0IENH0		
TAUD0I9	✓	✓	✓	✓	✓				DNFATAUD0IENH1		
TAUD0I10	✓	✓	✓	✓	✓				DNFATAUD0IENH2		
TAUD0I11	✓	✓	✓	✓	✓				DNFATAUD0IENH3		
TAUD0I12	✓	✓	✓	✓	✓				DNFATAUD0IENH4		
TAUD0I13	✓	✓	✓	✓	✓				DNFATAUD0IENH5		
TAUD0I14	✓	✓	✓	✓	✓				DNFATAUD0IENH6		
TAUD0I15	✓	✓	✓	✓	✓				DNFATAUD0IENH7		
TAUB0I0	✓	✓	✓	✓	✓	DNFATAUB0ICTL	FFC3 0020$_H$	DNFATAUB0IEN (DNFATAUB0IENH/ DNFATAUB0IENL)	DNFATAUB0IENL0	FFC3 0024$_H$ (FFC3 0028$_H$/ FFC3 002C$_H$)	*2
TAUB0I1	✓	✓	✓	✓	✓				DNFATAUB0IENL1		
TAUB0I2	✓	✓	✓	✓	✓				DNFATAUB0IENL2		
TAUB0I3	✓	✓	✓	✓	✓				DNFATAUB0IENL3		
TAUB0I4	✓	✓	✓	✓	✓				DNFATAUB0IENL4		
TAUB0I5	✓	✓	✓	✓	✓				DNFATAUB0IENL5		
TAUB0I6	✓	✓	✓	✓	✓				DNFATAUB0IENL6		
TAUB0I7	✓	✓	✓	✓	✓				DNFATAUB0IENL7		
TAUB0I8	✓	✓	✓	✓	✓				DNFATAUB0IENH0		
TAUB0I9	—	✓	✓	✓	✓				DNFATAUB0IENH1		
TAUB0I10	✓	✓	✓	✓	✓				DNFATAUB0IENH2		
TAUB0I11	—	✓	✓	✓	✓				DNFATAUB0IENH3		
TAUB0I12	✓	✓	✓	✓	✓				DNFATAUB0IENH4		
TAUB0I13	✓	✓	✓	✓	✓				DNFATAUB0IENH5		
TAUB0I14	✓	✓	✓	✓	✓				DNFATAUB0IENH6		
TAUB0I15	✓	✓	✓	✓	✓				DNFATAUB0IENH7		

Table 2B.88 Input Pins that Incorporate Digital Filter Type E

	Devises					Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
Input Pin	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUB1I0	—	—	✓	✓	✓	DNFATAUB1ICTL	FFC3 0040$_H$	DNFATAUB1IEN (DNFATAUB1IENH/ DNFATAUB1IENL)	DNFATAUB1IENL0	FFC3 0044$_H$ (FFC3 0048$_H$/ FFC3 004C$_H$)	*2
TAUB1I1	—	—	✓	✓	✓				DNFATAUB1IENL1		
TAUB1I2	—	—	✓	✓	✓				DNFATAUB1IENL2		
TAUB1I3	—	—	✓	✓	✓				DNFATAUB1IENL3		
TAUB1I4	—	—	✓	✓	✓				DNFATAUB1IENL4		
TAUB1I5	—	—	✓	✓	✓				DNFATAUB1IENL5		
TAUB1I6	—	—	✓	✓	✓				DNFATAUB1IENL6		
TAUB1I7	—	—	✓	✓	✓				DNFATAUB1IENL7		
TAUB1I8	—	—	✓	✓	✓				DNFATAUB1IENH0		
TAUB1I9	—	—	✓	✓	✓				DNFATAUB1IENH1		
TAUB1I10	—	—	✓	✓	✓				DNFATAUB1IENH2		
TAUB1I11	—	—	✓	✓	✓				DNFATAUB1IENH3		
TAUB1I12	—	—	✓	✓	✓				DNFATAUB1IENH4		
TAUB1I13	—	—	✓	✓	✓				DNFATAUB1IENH5		
TAUB1I14	—	—	✓	✓	✓				DNFATAUB1IENH6		
TAUB1I15	—	—	✓	✓	✓				DNFATAUB1IENH7		
ENCA0TIN0	✓	✓	✓	✓	✓	DNFAENCA0ICTL	FFC3 0060$_H$	DNFAENCA0IEN (DNFAENCA0IENL)	DNFAENCA0IENL0	FFC3 0064$_H$ (FFC3 006C$_H$)	*3
ENCA0TIN1	✓	✓	✓	✓	✓				DNFAENCA0IENL1		
ENCA0E0	✓	✓	✓	✓	✓				DNFAENCA0IENL2		
ENCA0E1	✓	✓	✓	✓	✓				DNFAENCA0IENL3		
ENCA0EC	✓	✓	✓	✓	✓				DNFAENCA0IENL4		
SENT0RX	—	✓	✓	✓	✓	DNFASENTICTL	FFC3 00E0$_H$	DNFASENTIEN (DNFASENTIENL)	DNFASENTIENL0	FFC3 00E4$_H$ (FFC3 00EC$_H$)	—*4
SENT1RX	✓	✓	✓	✓	✓				DNFASENTIENL1		

Note 1. For the setting for TAUD edge detection, see **Section 32.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register**.

Note 2. For the setting for TAUB edge detection, see **Section 31.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register**.

Note 3. For the setting for ENCA edge detection, see **Section 35.3.3, ENCAnIOC0 — ENCAn I/O Control Register 0**.

Note 4. RSENT does not have the edge detection.

2B.12.2 Clock Supply for Port Filters

The following table shows the clock supply for each filter type in each port domain.

Table 2B.89 Clock Supply for Port Filters

				Setting Register	
Peripheral Function	Port Domain*[1]	Filter Type	Filter Clock	Source Clock Selection	Clock Selection
ADCA0	Always-On area (AWO area)	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
ADCA1	Isolated area (ISO area)	Digital filter type D	DNFATCKI	CKSC_IADCAS_CTL	CKSC_IADCAD_CTL
TAUD0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
TAUB0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
TAUB1	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
ENCA0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
RSENTn	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—

Note 1. Power Domain

NOTE

For the Setting Register, see **Section 12AB.4.3, Clock Selector Control Register.**

2B.13 Description of Port Noise Filter & Edge/Level Detection

External signals pass through different types of filters according to the use of each external input signal.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2B.13.1 Overview

2B.13.1.1 Analog Filter Types

Analog filters have fixed characteristics.

- Type A: An analog filter with edge detection or level detection.
  Used for external interrupt signals.
- Type B: An analog filter
  Edge detection is performed by each peripheral function. Used for the timer input signals, asynchronous Hi-Z control input signals, and key return input signals.
- Type C: An analog filter only
  Used for the external RESET input and mode signals.

2B.13.1.2 Digital Filter Types

The digital filter characteristics can be adjusted to suit the application.

- Type D: A digital filter with edge detection.
  Used for the A/D converter external trigger pin.
- Type E: A digital filter. Edge detection is performed by each peripheral function.
  Used for the timer input signals and encoder input signals.

2B.13.2 Analog Filters

2B.13.2.1 Analog Filter Characteristic

See **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for the input conditions for signals input to pins that incorporate an analog filter.

2B.13.2.2 Analog Filter Control Registers

A dedicated FCLA0CTLm_<*name*> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Table 2B.84, Input Pins that Incorporate Analog Filter Type A**.

2B.13.2.3 Analog Filter in Standby Mode

Analog filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Analog filters in the Always-On area (AWO area) always operate.

The analog filter in standby mode and its wake-up capability depend on the filter types. See the description of the analog filter types below.

(1) Analog Filter Type A

A block diagram of analog filter type A is shown below.

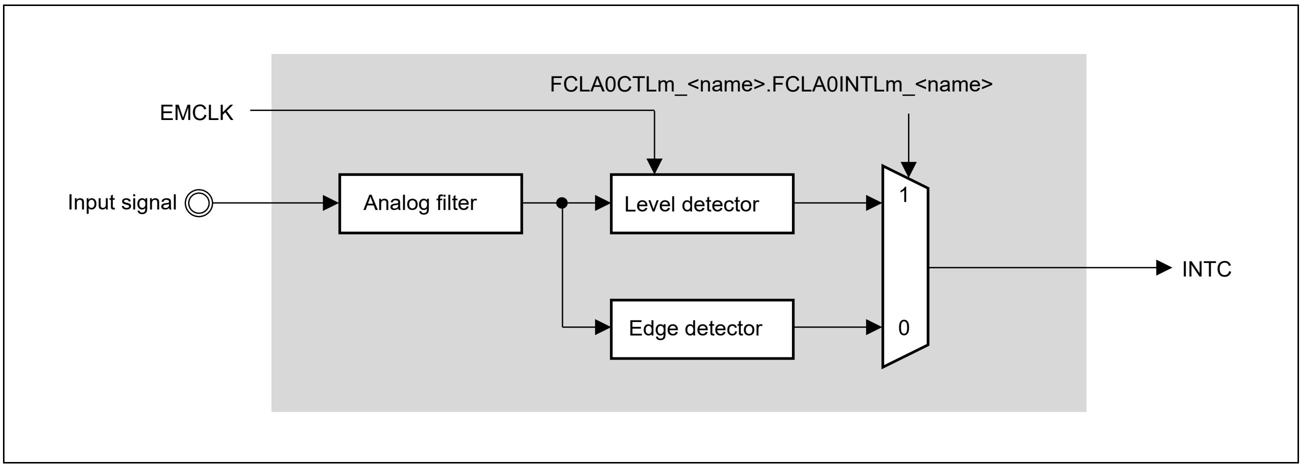


Figure 2B.13 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and glitches, an output signal is generated according to whether an event is detected; that is whether a specified level is detected or whether a change in the level (an edge) occurs.

Whether a level or an edge is detected is selected by the control bit FCLA0CTLm_<*name*>.FCLA0INTLm_<*name*>.

- FCLA0INTLm_<*name*> bit = 0: Edge detection

  Whether a rising or falling edge is detected can be specified by setting the FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> and FCLA0CTLm_<*name*>.FCLA0INTFm_<*name*> bits.

- FCLA0INTLm_<*name*> bit = 1: Level detection

  The detection of a high level or low level can be specified by setting FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> bit.

The table below summarizes the detection conditions of the analog filter.

Table 2B.90 Analog Filter Event Detection Conditions

FCLA0INTLm_<name>	FCLA0INTFm_<*name*>	FCLA0INTRm_<*name*>	Edge Detection	Level Detection
0	0	0	No edge detected	Disabled
	0	1	Rising edge	
	1	0	Falling edge	
	1	1	Rising and falling edges	
1	X	0	Disabled	Low level
	X	1		High level

Analog filter type A in Standby mode

The output signal of an analog filter type A can always be used as a standby mode wake-up signal.

(2) Analog filter type B

A block diagram of analog filter type B is shown below.

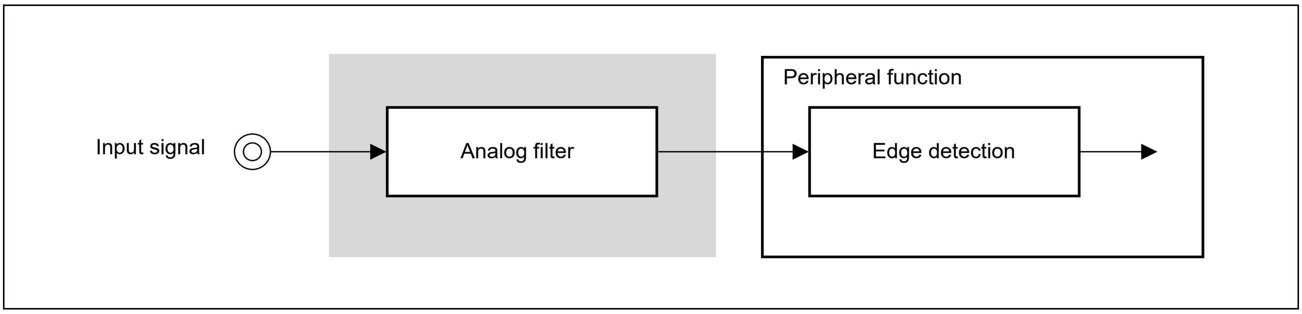


Figure 2B.14 Block Diagram of Analog Filter Type B

Analog filter type B in Standby mode

The output signal of an analog filter type B can always be used as a standby mode wake-up signal.

(3) Analog filter type C

A block diagram of analog filter type C is shown below.

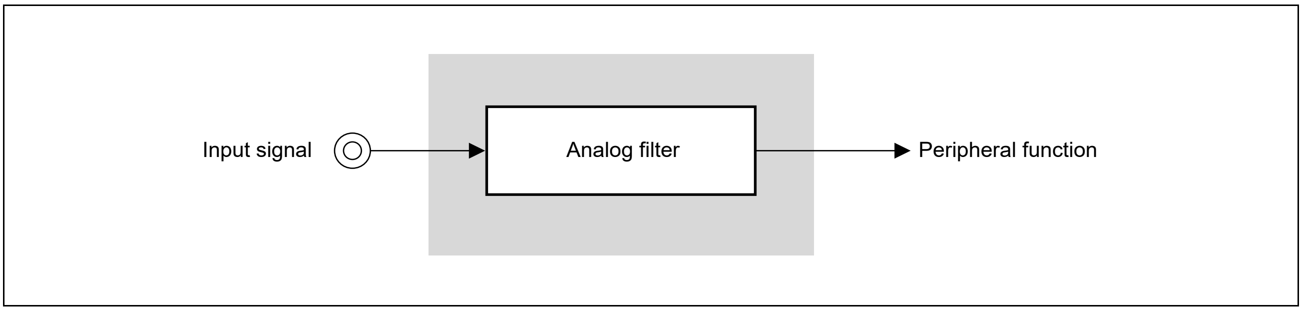


Figure 2B.15 Block Diagram of Analog Filter Type C

The generated signals are always input signals that have passed through an analog filter.

Analog filter type C in Standby mode

Pins equipped with type C analog filters in this product do not support the input of event signals to trigger wake-up from standby.

2B.13.3 Digital Filters

2B.13.3.1 Digital Filter Characteristic

The digital filters allow the filter characteristics to be adjusted accordingly to the needs of the application.

The input signal is sampled with the sampling frequency fs.

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] select the sampling frequency based on

  $f_s = f_{DNFATCKI} / 2^{DNFA<name>PRS[2:0]}$
  where fDNFATCKI is the frequency of the DNFATCKI clock.

- DNFA<*name*>CTL.DNFA<*name*>NFSTS[1:0] determines the number of same level samples, “s”, (2 to 5):

  s = DNFA<*name*>NFSTS[1:0] + 2

External signal pulses shorter than the following are suppressed at all times.

$s \times 1/f_s$

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

$(s + 1) \times 1/f_s$

External signal pulses in the following range may be suppressed or judged as valid.

$s \times 1/f_s$ to $(s + 1) \times 1/f_s$

The filter operation is illustrated in the figure below with DNFA<*name*>NFSTS[1:0] = 01_B, i.e. s = 3 same level samples.

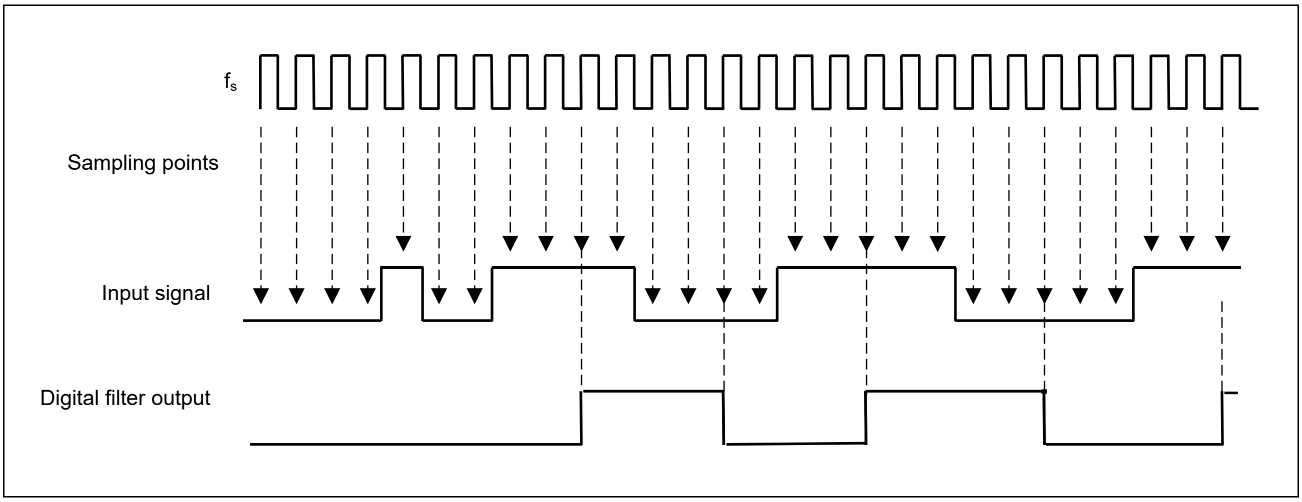


Figure 2B.16 Digital Filter Function

2B.13.3.2 Digital Filter Groups

The input signals processed through digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] and DNFA<*name*>NFSTS[1:0] apply to the signals.

However, the digital filter for each signal can be enabled or disabled separately by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

CAUTIONS

1. When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) for the port pin to switch to the alternative function.
   s = DNFA<*name*>NFSTS[1:0] + 2
   $s \times 1/f_s + 2 \times 1/f_{DNFATCKI}$
2. When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.
   $s \times 1/f_s + 3 \times 1/f_{DNFATCKI}$

2B.13.3.3 Digital Filters in Standby Mode

Digital filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Digital filters on the Always-On area (AWO area) are always operating.

Digital noise elimination requires the clock supply DNFATCKI to operate.
Pins equipped with digital filters in this product do not support the input of event signals to trigger wake-up from standby.

2B.13.3.4 Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFA<*name*>CTL and digital noise elimination enable register DNFA<*name*>EN are used to set all the filters in the same group (<*name*> = peripheral function group).

The DNFA<*name*>CTL register specifies the characteristics of the digital noise elimination filter for the digital filter of <*name*>.

The DNFA<*name*>EN register enables/disables each filter by setting the corresponding bit in DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The FCLA0CTLm_ADCn registers are ordered in groups of 3 registers with the same index n. The register index n is in 0 or 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2B.87, Input Pins that Incorporate Digital Filter Type D** and **Table 2B.88, Input Pins that Incorporate Digital Filter Type E** in **Section 2B.12.1, Port Filter Assignment**.

CAUTION

Do not change any control register settings while the corresponding digital filter is enabled by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1. Otherwise an unintended filter output may be generated.

(1) Digital filter type D

A block diagram of digital filter type D is shown below.

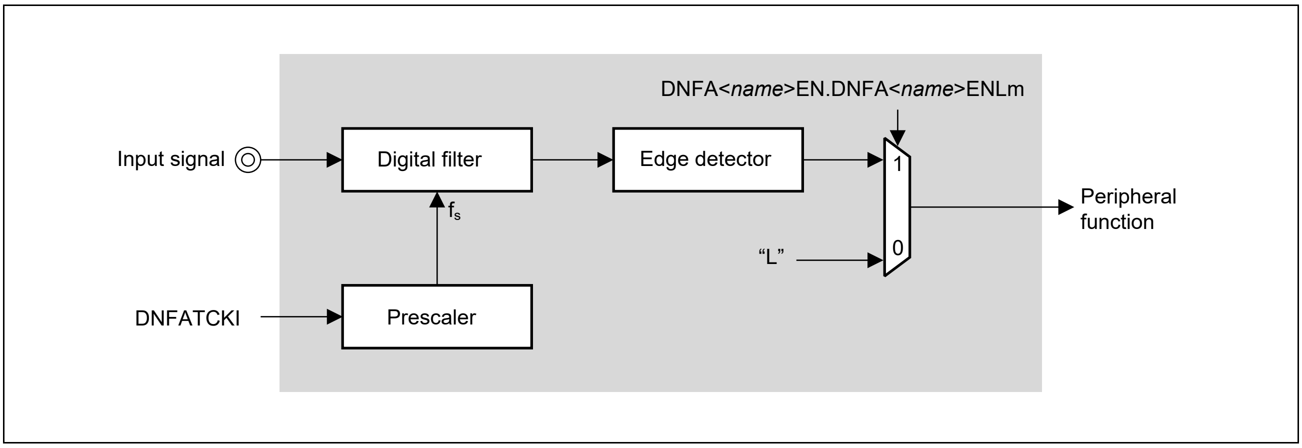


Figure 2B.17 Block Diagram of Digital Filter Type D

The generated signal depends on the register setting, as shown in the following table.

Table 2B.91 Output Options for Digital Filter Type D

DNFA<*name*>EN.DNFA<*name*>ENLm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

(2) Digital filter type E

A block diagram of digital filter type E is shown below.

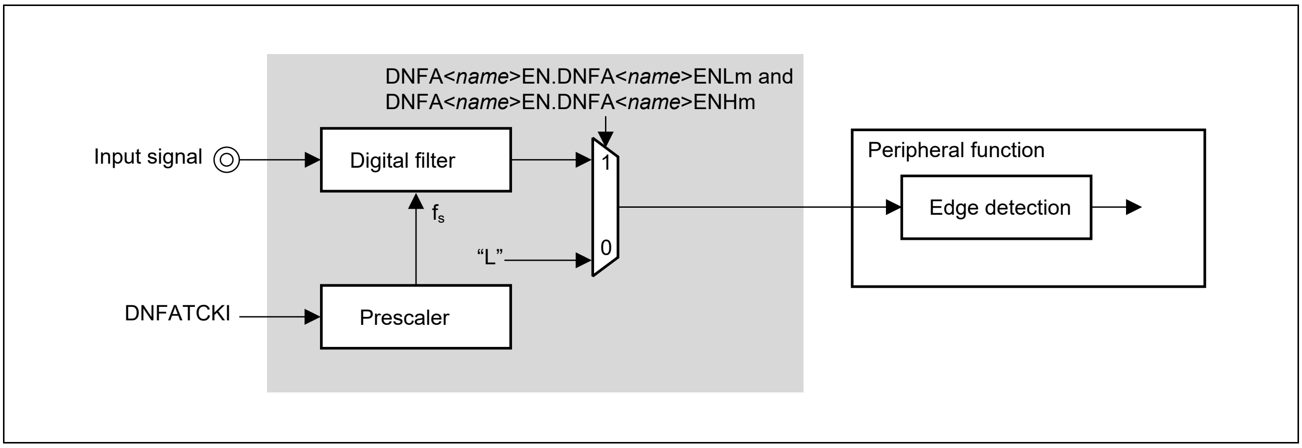


Figure 2B.18 Block Diagram of Digital Filter Type E

The generated signal depends on the register setting, as shown in the following table.

Table 2B.92 Output Options for Digital Filter Type E

DNFA<*name*>EN.DNFA<*name*>ENLm and DNFA<*name*>EN.DNFA<*name*>ENHm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

2B.13.4 Filter Control Registers

The analog and digital filters are controlled and operated by the following registers:

Table 2B.93 List of Filter Registers

Module Name	Register Name	Symbol	Address
FCLA0	Filter control register m	FCLA0CTLm_<name>	The addresses are shown in the tables in **Section 2B.12.1, Port Filter Assignment**.
DNF	Digital noise elimination control register	DNFA<name>CTL	
	Digital noise elimination enable register	DNFA<name>EN	
	Digital noise elimination enable H register	DNFA<name>ENH	
	Digital noise elimination enable L register	DNFA<name>ENL	

2B.13.4.1 FCLA0CTLm_<name> — Filter Control Register

This register controls the analog and digital filter operation.

Access: This register can be read or written in 8-bit units.

Address: The allocation of input signals to FCLA0CTLm_<name> registers and the address of each register are shown in the tables in **Section 2B.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLA0INTLm _<name>	FCLA0INTFm _<name>	FCLA0INTRm _<name>
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 2B.94 FCLA0CTLm_<name> Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	FCLA0INTLm _<name>	Detection Mode Selection 0: Edge detection 1: Level detection **NOTE:** This bit is only valid for analog filter type A.
1	FCLA0INTFm _<name>	● In level detection mode (FCLA0INTLm_<name> = 1): This bit has no effect. ● In edge detection mode (FCLA0INTLm_<name> = 0): Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.
0	FCLA0INTRm _<name>	● In level detection mode (FCLA0INTLm_<name> = 1): Detected level selection 0: Low level detection 1: High level detection ● In edge detection mode (FCLA0INTLm_<name> = 0): Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.

CAUTION

Digital filter type D: Always set bit 2 to “0”.

2B.13.4.2 DNFA<name>CTL — Digital Noise Elimination Control Register

This register is used to specify the filter characteristics of the digital noise elimination filter.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit units.

Address: For the correspondence between the DNFA<name>CTL register and input signals, and the addresses of individual registers, see **Table 2B.87, Input Pins that Incorporate Digital Filter Type D** and **Table 2B.88, Input Pins that Incorporate Digital Filter Type E** in **Section 2B.12.1, Port Filter Assignment.**

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	DNFA<name>NFSTS[1:0]		—	—	DNFA<name>PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2B.95 DNFA<name>CTL Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>6, 5</td><td>DNFA<name>
NFSTS[1:0]</td><td>The DNFA<name>NFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid.
<table>
<tr><th>DNFA<name>NFSTS[1:0]</th><th>Number of Samples</th></tr>
<tr><td>00_B</td><td>2</td></tr>
<tr><td>01_B</td><td>3</td></tr>
<tr><td>10_B</td><td>4</td></tr>
<tr><td>11_B</td><td>5</td></tr>
</table></td></tr>
<tr><td>4, 3</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>2 to 0</td><td>DNFA<name>
PRS[2:0]</td><td>Digital filter sampling clock selection
<table>
<tr><th>DNFA<name>PRS[2:0]</th><th>Sampling Clock Frequency</th></tr>
<tr><td>000_B</td><td>DNFATCKI/1</td></tr>
<tr><td>001_B</td><td>DNFATCKI/2</td></tr>
<tr><td>010_B</td><td>DNFATCKI/4</td></tr>
<tr><td>011_B</td><td>DNFATCKI/8</td></tr>
<tr><td>100_B</td><td>DNFATCKI/16</td></tr>
<tr><td>101_B</td><td>DNFATCKI/32</td></tr>
<tr><td>110_B</td><td>DNFATCKI/64</td></tr>
<tr><td>111_B</td><td>DNFATCKI/128</td></tr>
</table></td></tr>
</table>

2B.13.4.3 DNFA<name>EN — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 16-bit units.

The upper- and lower-order bytes (DNFA<name>ENH[7:0] and DNFA<name>ENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH. and DNFA<name>ENL.

Address: For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Table 2B.87, Input Pins that Incorporate Digital Filter Type D** and **Table 2B.88, Input Pins that Incorporate Digital Filter Type E** in **Section 2B.12.1, Port Filter Assignment.**

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFA<name>ENH7	DNFA<name>ENH6	DNFA<name>ENH5	DNFA<name>ENH4	DNFA<name>ENH3	DNFA<name>ENH2	DNFA<name>ENH1	DNFA<name>ENH0	DNFA<name>ENL7	DNFA<name>ENL6	DNFA<name>ENL5	DNFA<name>ENL4	DNFA<name>ENL3	DNFA<name>ENL2	DNFA<name>ENL1	DNFA<name>ENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2B.96 DNFA<name>EN Register Contents

Bit Position	Bit Name	Function
15 to 0	DNFA<name>ENH[7:0] DNFA<name>ENL[7:0]	Digital Noise Elimination Enable/Disable Control 0: Fixed to low level 1: Input signal passed through filter

2B.13.4.4 DNFA<name>ENH – Digital Noise Elimination Enable H Register

Setting in this register correspond to those of the 8 upper-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENH register and input signals, and the addresses of individual registers, see **Table 2B.88, Input Pins that Incorporate Digital Filter Type E** in **Section 2B.12.1, Port Filter Assignment.**

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA<name> ENH7	DNFA<name> ENH6	DNFA<name> ENH5	DNFA<name> ENH4	DNFA<name> ENH3	DNFA<name> ENH2	DNFA<name> ENH1	DNFA<name> ENH0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2B.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

2B.13.4.5 DNFA<name>ENL – Digital Noise Elimination Enable L Register

Setting in this register correspond to those of the 8 lower-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENL register and input signals, and the addresses of individual registers, see **Table 2B.87, Input Pins that Incorporate Digital Filter Type D** and **Table 2B.88, Input Pins that Incorporate Digital Filter Type E** in **Section 2B.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA<name> ENL7	DNFA<name> ENL6	DNFA<name> ENL5	DNFA<name> ENL4	DNFA<name> ENL3	DNFA<name> ENL2	DNFA<name> ENL1	DNFA<name> ENL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2B.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

Section 2C Pin Function of RH850/F1KM-S1

This section describes the pin and port functions.

Section 2C.1, Pin Connection Diagram to **Section 2C.5, Recommended Connection of Unused Pins** describe the pin connections and respective pins.

Section 2C.6, Features of RH850/F1KM Port to **Section 2C.13, Description of Port Noise Filter & Edge/Level Detection** describe the general port functions.

2C.1 Pin Connection Diagram

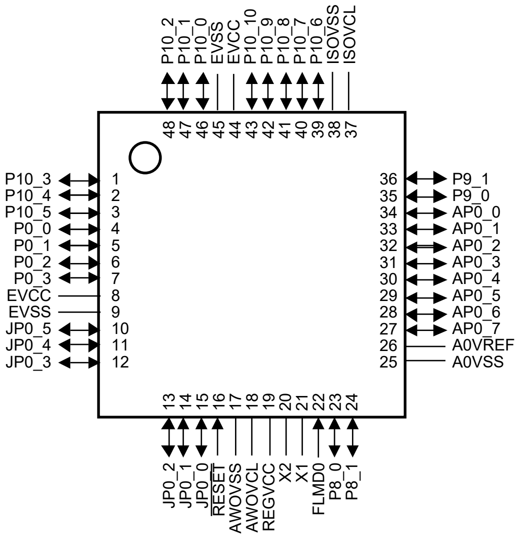


Figure 2C.1 Pin Connection Diagram (48-Pin LQFP)

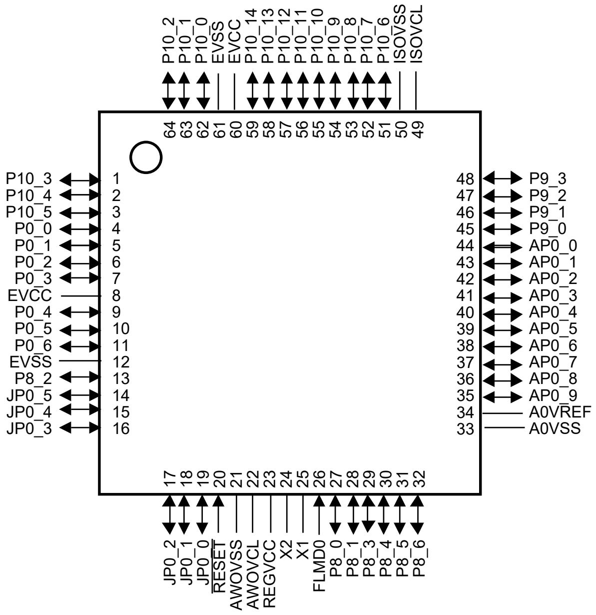


Figure 2C.2 Pin Connection Diagram (64-Pin LQFP)

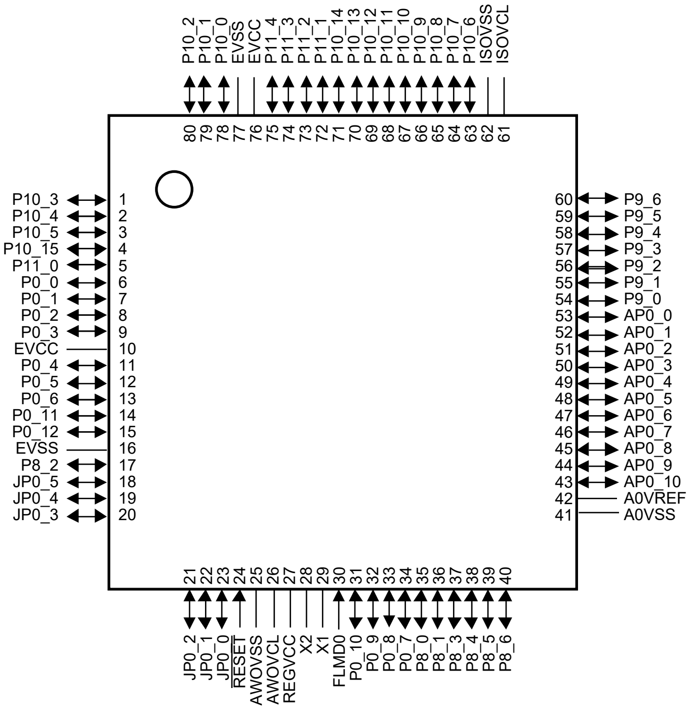


Figure 2C.3 Pin Connection Diagram (80-Pin LQFP)

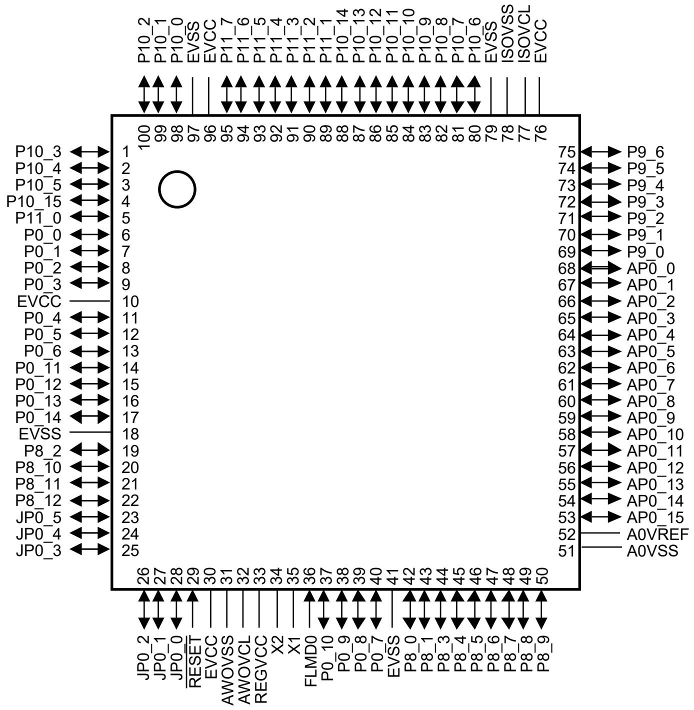


Figure 2C.4 Pin Connection Diagram (100-Pin LQFP)

Table 2C.1 Pin Assignment 48-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
6	P0_2 / TAUD0I6 / TAUD0O6 / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / DPIN1 / CSIH0SO / TAUJ1I0 / TAUJ1O0
8	EVCC
9	EVSS
10	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
11	JP0_4 / $\overline{\text{DCUTRST}}$
12	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
13	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
14	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
15	JP0_0 / INTP0 / FPDR / FPDT / TAUJ2I0 / TAUJ2O0 / DCUTDI / LPDI/ LPDIO
16	$\overline{\text{RESET}}$
17	AWOVSS
18	AWOVCL
19	REGVCC
20	X2
21	X1
22	FLMD0
23	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / INTP4 / CSIH0CSS0 / SENT0RX / ADCA0I0S / RIIC1SDA
24	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / INTP5 / SENT0SPCO / ADCA0I1S / RIIC1SCL
25	A0VSS
26	A0VREF
27	AP0_7 / ADCA0I7
28	AP0_6 / ADCA0I6
29	AP0_5 / ADCA0I5
30	AP0_4 / ADCA0I4
31	AP0_3 / ADCA0I3
32	AP0_2 / ADCA0I2
33	AP0_1 / ADCA0I1
34	AP0_0 / ADCA0I0
35	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / ADCA0I2S / RIIC1SDA
36	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / ADCA0I3S / RIIC1SCL
37	ISOVCL
38	ISOVSS
39	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / MODE2
40	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / TAUJ3I1 / TAUJ3O1
41	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / TAUJ3I2 / TAUJ3O2 / FLMD1
42	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
43	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
44	EVCC

Table 2C.1 Pin Assignment 48-Pin LQFP

Pin No.	Pin Name
45	EVSS
46	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / TAUJ1O3
47	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / TAUJ3O0 / MODE0
48	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP /MODE1

Table 2C.2 Pin Assignment 64-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
6	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
8	EVCC
9	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / SELDP0 / DPIN8
10	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1
11	P0_6 / INTP2 / DPIN10 / SELDP2
12	EVSS
13	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / PWGA22O / ADCA0I4S
14	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
15	JP0_4 / $\overline{\text{DCUTRST}}$
16	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
17	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
18	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
19	JP0_0 / INTP0 / FPDR / FPDT / TAUJ2I0 / TAUJ2O0 / DCUTDI / LPDI/ LPDIO
20	$\overline{\text{RESET}}$
21	AWOVSS
22	AWOVCL
23	REGVCC
24	X2
25	X1
26	FLMD0
27	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / SENT0RX / ADCA0I0S / RIIC1SDA
28	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / SENT0SPCO / ADCA0I1S / RIIC1SCL
29	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / PWGA23O / ADCA0I5S
30	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / ADCA0I6S
31	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / ADCA0I7S
32	P8_6 / NMI / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
33	A0VSS
34	A0VREF
35	AP0_9 / ADCA0I9
36	AP0_8 / ADCA0I8
37	AP0_7 / ADCA0I7
38	AP0_6 / ADCA0I6
39	AP0_5 / ADCA0I5
40	AP0_4 / ADCA0I4
41	AP0_3 / ADCA0I3
42	AP0_2 / ADCA0I2
43	AP0_1 / ADCA0I1
44	AP0_0 / ADCA0I0

Table 2C.2 Pin Assignment 64-Pin LQFP

Pin No.	Pin Name
45	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / ADCA0I2S / RIIC1SDA
46	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / ADCA0I3S / RIIC1SCL
47	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / ADCA0I9S
48	P9_3 / KR0I7 / PWGA21O / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
49	ISOVCL
50	ISOVSS
51	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MODE2
52	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / TAUJ3I1 / TAUJ3O1
53	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / TAUJ3I2 / TAUJ3O2 / FLMD1
54	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
55	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
56	P10_11 / PWGA16O / RLIN31RX / INTP11
57	P10_12 / PWGA17O / RLIN31TX
58	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O
59	P10_14 / PWGA19O
60	EVCC
61	EVSS
62	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / TAUJ1O3
63	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / TAUJ3O0 / MODE0
64	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP /MODE1

Table 2C.3 Pin Assignment 80-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P10_15 / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RYI / CSIH2RYO / TAUB0I11 / TAUB0O11
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
14	P0_11 / RIIC0SDA / CSIH1CSS2 / TAUB0I8 / TAUB0O8
15	P0_12 / RIIC0SCL / TAUB0I10 / TAUB0O10 / CSIG0SI
16	EVSS
17	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
18	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
19	JP0_4 / $\overline{\text{DCUTRST}}$
20	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
21	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
22	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
23	JP0_0 / INTP0 / FPDR / FPDT / TAUJ2I0 / TAUJ2O0 / DCUTDI / LPDI/ LPDIO
24	$\overline{\text{RESET}}$
25	AWOVSS
26	AWOVCL
27	REGVCC
28	X2
29	X1
30	FLMD0
31	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / TAUB0I6 / TAUB0O6
32	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / TAUB0I4 / TAUB0O4
33	P0_8 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2
34	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0
35	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / SENT0RX / ADCA0I0S / RIIC1SDA
36	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / SENT0SPCO / ADCA0I1S / RIIC1SCL
37	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
38	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / ADCA0I6S
39	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / ADCA0I7S
40	P8_6 / NMI / CSIH0CSS4 / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
41	A0VSS
42	A0VREF
43	AP0_10 / ADCA0I10
44	AP0_9 / ADCA0I9

Table 2C.3 Pin Assignment 80-Pin LQFP

Pin No.	Pin Name
45	AP0_8 / ADCA0I8
46	AP0_7 / ADCA0I7
47	AP0_6 / ADCA0I6
48	AP0_5 / ADCA0I5
49	AP0_4 / ADCA0I4
50	AP0_3 / ADCA0I3
51	AP0_2 / ADCA0I2
52	AP0_1 / ADCA0I1
53	AP0_0 / ADCA0I0
54	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / ADCA0I2S / RIIC1SDA
55	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / ADCA0I3S / RIIC1SCL
56	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
57	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
58	P9_4 / CSIH0CSS5 / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
59	P9_5 / CSIH0CSS6 / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
60	P9_6 / CSIH0CSS7 / ADCA0I13S
61	ISOVCL
62	ISOVSS
63	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MODE2
64	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / TAUJ3I1 / TAUJ3O1
65	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / TAUJ3I2 / TAUJ3O2 / FLMD1
66	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
67	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
68	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
69	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
70	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
71	P10_14 / PWGA19O / RLIN32TX / TAUB0I7 / TAUB0O7
72	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX / CSIH0CSS7 / TAUB0I13 / TAUB0O13
73	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / TAUB0I15 / TAUB0O15
74	P11_3 / CSIH2SC / RLIN32TX
75	P11_4 / CSIH2SI
76	EVCC
77	EVSS
78	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / CSIH1SI / TAUJ1O3
79	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / CSIH1SC / TAUJ3O0 / MODE0
80	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Table 2C.4 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RYI / CSIH2RYO / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / TAUJ2I1 / TAUJ2O1
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO / TAUJ2I2 / TAUJ2O2
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO / TAUJ2I3 / TAUJ2O3
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / TAUJ1I0 / TAUJ1O0
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8 / TAUB0I12 / TAUB0O12
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO / TAUB0I14 / TAUB0O14
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
14	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / PWGA34O
15	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
16	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
17	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
18	EVSS
19	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
20	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
21	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
22	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
23	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
24	JP0_4 / $\overline{\text{DCUTRST}}$
25	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
26	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / FPCK / DCUTCK / LPDCLK
27	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / FPDT / DCUTDO / LPDO
28	JP0_0 / INTP0 / FPDR / FPDT / TAUJ2I0 / TAUJ2O0 / DCUTDI / LPDI/ LPDIO
29	$\overline{\text{RESET}}$
30	EVCC
31	AWOVSS
32	AWOVCL
33	REGVCC
34	X2
35	X1
36	FLMD0
37	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
38	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
39	P0_8 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
40	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
41	EVSS
42	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / SENT0RX / ADCA0I0S / RIIC1SDA
43	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / SENT0SPCO / ADCA0I1S / RIIC1SCL
44	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S

Table 2C.4 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
45	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
46	P8_5 / TAUJ0I3 / TAUJ0O3 / NMI / CSIH0CSS3 / PWGA37O / ADCA0I7S
47	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S / $\overline{\text{RESETOUT}}$
48	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0SEL0 / RTCA0OUT / ADCA0I14S
49	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0SEL1 / ADCA0I15S
50	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0SEL2 / ADCA0I16S
51	A0VSS
52	A0VREF
53	AP0_15 / ADCA0I15
54	AP0_14 / ADCA0I14
55	AP0_13 / ADCA0I13
56	AP0_12 / ADCA0I12
57	AP0_11 / ADCA0I11
58	AP0_10 / ADCA0I10
59	AP0_9 / ADCA0I9
60	AP0_8 / ADCA0I8
61	AP0_7 / ADCA0I7
62	AP0_6 / ADCA0I6
63	AP0_5 / ADCA0I5
64	AP0_4 / ADCA0I4
65	AP0_3 / ADCA0I3
66	AP0_2 / ADCA0I2
67	AP0_1 / ADCA0I1
68	AP0_0 / ADCA0I0
69	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / TAUJ1I1 / TAUJ1O1 / SENT1RX / ADCA0I2S / RIIC1SDA
70	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / TAUJ1I2 / TAUJ1O2 / SENT1SPCO / ADCA0I3S / RIIC1SCL
71	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
72	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
73	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
74	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
75	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
76	EVCC
77	ISOVCL
78	ISOVSS
79	EVSS
80	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MODE2
81	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / TAUJ3I1 / TAUJ3O1
82	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / TAUJ3I2 / TAUJ3O2 / FLMD1
83	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
84	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / TAUJ3I3 / TAUJ3O3
85	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
86	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
87	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
88	P10_14 / PWGA19O / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7

Table 2C.4 Pin Assignment 100-Pin LQFP

Pin No.	Pin Name
89	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX / CSIH0CSS7 / PWGA26O / TAUB0I13 / TAUB0O13
90	P11_2 / CSIH2SO / RLIN32RX / INTP12 / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15
91	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / RLIN32TX
92	P11_4 / CSIH2SI / CAN3TX / PWGA29O
93	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
94	P11_6 / RLIN33RX / INTP13 / CAN5TX / PWGA31O / CSIH3SO
95	P11_7 / INTP5 / PWGA32O / CSIH3SC
96	EVCC
97	EVSS
98	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAUJ1I3 / TAPA0UP / CSIH1SI / TAUJ1O3
99	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAUJ3I0 / TAPA0UN / CSIH1SC / TAUJ3O0 / MODE0
100	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

2C.2 Pin Description

Table 2C.5 Pin Functions

	No. of Pins						
Pin Name	48 Pins	64 Pins	80 Pins	100 Pins	IO	Pin Function	Unit
A0VREF	✓	✓	✓	✓	—	ADCAn voltage supply and reference voltage	ADCAn
A0VSS	✓	✓	✓	✓	—	ADCAn ground	
ADCA0Im	✓	✓	✓	✓	I	ADCA0 input channel m with 12-bit resolution	
	m = 0 to 7	m = 0 to 9	m = 0 to 10	m = 0 to 15			
ADCA0ImS	✓	✓	✓	✓	I	ADCA0 input channel m with 10-bit resolution	
	m = 0 to 3	m = 0 to 10	m = 0 to 13	m = 0 to 19			
ADCA0SELy	✓	✓	✓	✓	O	External MPX selection pin y for ADCA0 input	
	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2			
ADCA0TRGy	✓	✓	✓	✓	I	ADCA0 external trigger pin y	
	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2			
AP0_m	✓	✓	✓	✓	IO	Analog port 0_m	Port
	m = 0 to 7	m = 0 to 9	m = 0 to 10	m = 0 to 15			
APO	✓	✓	✓	✓	O	Port output signal for analog input	LPS0
AWOVCL	✓	✓	✓	✓	—	Voltage regulator for Always-On area (AWO area) capacitor connection	Power
AWOVSS	✓	✓	✓	✓	—	Internal logic for Always-On area (AWO area) ground	
CANzRX	✓	✓	✓	✓	I	CANz receive data input	RCFDCn
	z = 0	z = 0 to 2	z = 0 to 2	z = 0 to 5			
CANzTX	✓	✓	✓	✓	O	CANz transmit data output	
	z = 0	z = 0 to 2	z = 0 to 2	z = 0 to 5			
CSCXFOUT	✓	✓	✓	✓	O	Clock output	Clock
CSIG0RYI	✓	✓	✓	✓	I	CSIGn ready (1) / busy (0) input signal	CSIGn
CSIG0RYO	✓	✓	✓	✓	O	CSIGn ready (1) / busy (0) output signal	
CSIG0SC	✓	✓	✓	✓	IO	CSIGn serial clock signal	
CSIG0SI	✓	✓	✓	✓	I	CSIGn serial data input	
CSIG0SO	✓	✓	✓	✓	O	CSIGn serial data output	
$\overline{\text{CSIG0SSI}}$	✓	✓	✓	✓	I	CSIGn SS function control input signal	
CSIHnCSS0	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
CSIHnCSS1	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 1	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
CSIHnCSS2	—	✓	✓	✓	O	CSIHn serial peripheral chip select signal 2	
		n = 0	n = 0 to 2	n = 0 to 3			
CSIHnCSS3	—	✓	✓	✓	O	CSIHn serial peripheral chip select signal 3	
		n = 0	n = 0 to 2	n = 0 to 3			
CSIHnCSS4	—	—	✓	✓	O	CSIHn serial peripheral chip select signal 4	
			n = 0	n = 0, 1			
CSIHnCSS5	—	—	✓	✓	O	CSIHn serial peripheral chip select signal 5	
			n = 0	n = 0, 1			
CSIHnCSS6	—	—	✓	✓	O	CSIHn serial peripheral chip select signal 6	
			n = 0	n = 0			
CSIHnCSS7	—	—	✓	✓	O	CSIHn serial peripheral chip select signal 7	
			n = 0	n = 0			
CSIHnRYI	✓	✓	✓	✓	I	CSIHn ready (1) / busy (0) input signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
CSIHnRYO	✓	✓	✓	✓	O	CSIHn ready (1) / busy (0) output signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
CSIHnSC	✓	✓	✓	✓	IO	CSIHn serial clock signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			

Table 2C.5 Pin Functions

	No. of Pins						
Pin Name	48 Pins	64 Pins	80 Pins	100 Pins	IO	Pin Function	Unit
CSIHnSI	✓	✓	✓	✓	I	CSIHn serial data input	CSIHn
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
CSIHnSO	✓	✓	✓	✓	O	CSIHn serial data output	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
$\overline{\text{CSIHnSSI}}$	✓	✓	✓	✓	I	CSIHn slave select input signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3			
$\overline{\text{DCURDY}}$	✓	✓	✓	✓	O	Debug ready	OCD
DCUTCK	✓	✓	✓	✓	I	Debug clock	
DCUTDI	✓	✓	✓	✓	I	Debug data input	
DCUTDO	✓	✓	✓	✓	O	Debug data output	
DCUTMS	✓	✓	✓	✓	I	Debug mode select	
$\overline{\text{DCUTRST}}$	✓	✓	✓	✓	I	Debug reset	
DPINm	✓	✓	✓	✓	I	Digital port input m	LPS0
	m = 0 to 2	m = 0 to 4, 8 to 10	m = 0 to 11	m = 0 to 16			
DPO	✓	✓	✓	✓	O	Port output signal for digital input	
ENCA0TINm	✓	✓	✓	✓	I	ENCA0 capture trigger input m	ENCAn
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1			
ENCA0EC	✓	✓	✓	✓	I	ENCA0 encoder clear input	
ENCA0E0	✓	✓	✓	✓	I	ENCA0 encoder input 0	
ENCA0E1	✓	✓	✓	✓	I	ENCA0 encoder input 1	
EVCC	✓	✓	✓	✓	—	Port buffer voltage supply	Power
EVSS	✓	✓	✓	✓	—	Port buffer ground	
FLMD0	✓	✓	✓	✓	I	Operating mode select pin 0	Mode
FLMD1	✓	✓	✓	✓	I	Operating mode select pin 1	
FPDR	✓	✓	✓	✓	I	Serial Communication Interface RXD	FLASH
FPDT	✓	✓	✓	✓	O	Serial Communication Interface TXD	
FPCK	✓	✓	✓	✓	I	Serial Communication Interface clock	
INTPm	✓	✓	✓	✓	I	External interrupt input m	INTC
	m = 0 to 5, 10, 11	m = 0 to 5, 10, 11	m = 0 to 8, 10 to 12	m = 0 to 8, 10 to 13			
ISOVCL	✓	✓	✓	✓	—	Voltage regulator for Isolated area (ISO area) capacitor connection	Power
ISOVSS	✓	✓	✓	✓	—	Internal logic for Isolated area (ISO area) ground	
JP0_m	✓	✓	✓	✓	IO	JTAG port 0_m	JTAG
	m = 0 to 5	m = 0 to 5	m = 0 to 5	m = 0 to 5			
KR0Im	✓	✓	✓	✓	I	KR0 key input signal	KRn
	m = 0 to 5	m = 0 to 7	m = 0 to 7	m = 0 to 7			
LPDCLK	✓	✓	✓	✓	I	LPD clock input (4-pin mode)	LPD
LPDCLKOUT	✓	✓	✓	✓	O	LPD clock output (4-pin mode)	
LPDI	✓	✓	✓	✓	I	LPD data input (4-pin mode)	
LPDIO	✓	✓	✓	✓	IO	LPD data input / output (1-pin mode)	
LPDO	✓	✓	✓	✓	O	LPD data output (4-pin mode)	
MODEm	✓	✓	✓	✓	I	Sub operating mode select (Boundary scan)	Mode
	m = 0 to 2	m = 0 to 2	m = 0 to 2	m = 0 to 2			
NMI	✓	✓	✓	✓	I	External non-maskable interrupt input	INTC
P0_m	✓	✓	✓	✓	IO	Port 0_m	Port
	m = 0 to 3	m = 0 to 6	m = 0 to 12	m = 0 to 14			
P8_m	✓	✓	✓	✓	IO	Port 8_m	
	m = 0 to 1	m = 0 to 6	m = 0 to 6	m = 0 to 12			
P9_m	✓	✓	✓	✓	IO	Port 9_m	
	m = 0 to 1	m = 0 to 3	m = 0 to 6	m = 0 to 6			

Table 2C.5 Pin Functions

	No. of Pins						
Pin Name	48 Pins	64 Pins	80 Pins	100 Pins	IO	Pin Function	Unit
P10_m	✓	✓	✓	✓	IO	Port 10_m	Port
	m = 0 to 10	m = 0 to 14	m = 0 to 15	m = 0 to 15			
P11_m	—	—	✓	✓	IO	Port 11_m	
			m = 0 to 4	m = 0 to 7			
PWGAnO	✓	✓	✓	✓	O	PWGAn output signal	PWGAn
	n = 0 to 12	n = 0 to 23	m =0 to 23	n = 0 to 47			
REGVCC	✓	✓	✓	✓	—	Voltage regulators voltage supply	Power
$\overline{\text{RESET}}$	✓	✓	✓	✓	I	External reset input	Reset
$\overline{\text{RESETOUT}}$	—	✓	✓	✓	O	Reset output	
RIICnSCL	✓	✓	✓	✓	IO	RIICn serial clock	RIICn
	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
RIICnSDA	✓	✓	✓	✓	IO	RIICn serial data	
	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
RLIN2mRX	✓	✓	✓	✓	I	RLIN2m receive data input	RLIN24n
	m = 0 to 1	m = 0 to 1	m = 0 to 1	m = 0 to 2			
RLIN2mTX	✓	✓	✓	✓	O	RLIN2m transmit data output	
	m = 0 to 1	m = 0 to 1	m = 0 to 1	m = 0 to 2			
RLIN3nRX	✓	✓	✓	✓	I	RLIN3n receive data input	RLIN3n
	n = 0	n = 0 to 1	n = 0 to 2	n = 0 to 3			
RLIN3nTX	✓	✓	✓	✓	O	RLIN3n transmit data output	
	n = 0	n = 0 to 1	n = 0 to 2	n = 0 to 3			
SENTnRX	✓	✓	✓	✓	I	SENT data input	RSENTn
	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
SENTnSPCO	✓	✓	✓	✓	O	SENT SPC extension output	
	n = 0 to 1	n = 0 to 1	n = 0 to 1	n = 0 to 1			
RTCA0OUT	✓	✓	✓	✓	O	RTCA0 1Hz output	RTCAn
SELDPk	—	✓	✓	✓	O	External multiplexer selection output signal k for digital port	LPS0
		k = 0 to 2	k = 0 to 2	k = 0 to 2			
TAPA0ESO	✓	✓	✓	✓	I	Hi-Z control	TAPAn
TAPA0UN	✓	✓	✓	✓	O	Motor control output U phase (negative)	
TAPA0UP	✓	✓	✓	✓	O	Motor control output U phase (positive)	
TAPA0VN	✓	✓	✓	✓	O	Motor control output V phase (negative)	
TAPA0VP	✓	✓	✓	✓	O	Motor control output V phase (positive)	
TAPA0WN	✓	✓	✓	✓	O	Motor control output W phase (negative)	
TAPA0WP	✓	✓	✓	✓	O	Motor control output W phase (positive)	
TAUD0Im	✓	✓	✓	✓	I	TAUD0 channel input m	TAUDn
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUD0Om	✓	✓	✓	✓	O	TAUD0 channel output m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUB0Im	—	—	✓	✓	I	TAUBn channel input m	TAUBn
			m = 0 to 15	m = 0 to 15			
TAUB0Om	—	—	✓	✓	O	TAUBn channel output m	
			m = 0 to 15	m = 0 to 15			

Table 2C.5 Pin Functions

	No. of Pins						
Pin Name	48 Pins	64 Pins	80 Pins	100 Pins	IO	Pin Function	Unit
TAUJnIm	✓	✓	✓	✓	I	TAUJn channel input m	TAUJn
	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3			
TAUJnOm	✓	✓	✓	✓	O	TAUJn channel output m	
	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3	n = 0 to 3 m = 0 to 3			
X1, X2	✓	✓	✓	✓	—	MainOSC connections	MOSC

CAUTION

When pin functions for a peripheral module are allocated to multiple pins, use the pins from the same port group or nearby pins as the pins for a given channel.

- **(e.g.)** When RS-CANFD channel 0 is used:

  CAN0TX P0_0 P10_1
  CAN0RX P0_1 P10_0

  Use one of the following pin combinations:

  - P0_0 and P0_1, or
  - P10_0 and P10_1.

  The combinations of P0_0 and P10_0, and P0_1 and P10_1 are not allowed.

2C.3 Pin Functions During and After Reset

Table 2C.6 Pin Functions During and After Reset

Pins	During Reset	After Reset
JP0_0	High impedance	JP0_0: Input Serial programming mode: FPDR, FPDT (1 wire UART) FPDR (2 wire UART) Nexus I/F: DCUTDI input LPD (4 pins): LPDI input LPD (1 pin): LPDIO input/output
JP0_1	High impedance	JP0_1: Input Serial programming mode: FPDT Nexus I/F: DCUTDO output LPD (4 pins): LPDO output LPD (1 pin): High impedance
JP0_2	High impedance	JP0_2: Input Serial programming mode: FPCK Nexus I/F: DCUTCK input LPD (4 pins): LPDCLK input LPD (1 pin): High impedance
JP0_3	High impedance	JP0_3: Input Serial programming mode: High impedance Nexus I/F: DCUTMS input LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_4	Input[*3,*5]	JP0_4: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCUTRST}}$ input[*1] LPD (4 pins): High impedance LPD (1 pin): High impedance
JP0_5	High impedance	JP0_5: Input Serial programming mode: High impedance Nexus I/F: $\overline{\text{DCURDY}}$ output LPD (4 pins): LPDCLKOUT output LPD (1 pin): High impedance
P8_6	Output[*2 *4]	Output (OPBT0.RESETOUTEN = 1)[*2] High impedance (OPBT0.RESETOUTEN = 0)[*2, *4]
P0, P8 to 11 (except P8_6, P10_1, P10_2, P10_6 and P10_8)	High impedance	High impedance
P10_1	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE0 input (FLMD0 = 1, FLMD1 = 1)
P10_2	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE1 input (FLMD0 = 1, FLMD1 = 1)
P10_6	High impedance	High impedance (FLMD0 = 0) High impedance (FLMD0 = 1, FLMD1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 0) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 0, MODE1 = 1) High impedance (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 0) MODE2 input (FLMD0 = 1, FLMD1 = 1, MODE0 = 1, MODE1 = 1)
P10_8	High impedance	High impedance (FLMD0 = 0) FLMD1 input (FLMD0 = 1)
FLMD0	Input	Input
$\overline{\text{RESET}}$	Input	Input
AP0	High impedance	High impedance

Note 1. When Nexus is enabled and no external device is connected, the level of the pin must always be fixed to low level.

Note 2. $\overline{\text{RESETOUT}}$ is output. For details, see **Section 2C.11, Port (Special I/O) Function Overview**.

Note 3. When the power is turned on or when $\overline{\text{RESET}}$ is low level, JP0_4 pin should be driven low level.

Note 4. If OPBT0.RESETOUTEN = 0, P8_6 pin status has a possibility to become unstable (less than 15 μs) at the transition moment to reset status by internal reset factors.

Note 5. When $\overline{\text{RESET}}$ is low level, on-chip pull-down resistor is connected to JP0_4.

2C.4 Port State in Standby Mode

For the port state in standby mode, see **Section 14.1.4, I/O Buffer Control**.

2C.5 Recommended Connection of Unused Pins

If the pins are not used, it is recommended to connect them as shown below.

Table 2C.7 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
A0VREF	Connected to EVCC
A0VSS	Connected to EVSS
$\overline{\text{RESET}}$	Connected to EVCC via a resistor
X1	Connected to AWOVSS via a resistor
X2	Open
JP0 (excluding JP0_4) P0 P8 (excluding P8_6) P9	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P8_6	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS via a resistor
P10 (excluding P10_1, P10_2, P10_6, P10_8) P11	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P10_1, P10_2, P10_6, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC via a resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Connected to EVCC via a resistor $\overline{\text{DCUTRST}}$ (JP0_4): Connected to EVSS via a resistor*1 $\overline{\text{DCURDY}}$ /LPDCLKOUT (JP0_5): Open

Note 1. For in case when a debugging interface is used, this pin should be connected to EVCC through resistor depending on the development tool made by a third party.

2C.6 Features of RH850/F1KM Port

2C.6.1 Port Group

The RH850/F1KM provides the following port groups, indicated by the numbers in the table below.

Table 2C.8 Port Groups in RH850/F1KM-S1

No. of Pins	Port Group	RH850/F1KM-S1
48 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
64 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
80 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
100 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0

2C.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n”. For example, the port mode control register of the Pn pin is PMCn (n = 0, and 8 to 11).

2C.6.3 Register Base Addresses

Port and JTAG port base addresses are listed in the following table.

Port and JTAG port register addresses are given as offsets from the base addresses.

Table 2C.9 Register Base Addresses

Base Address Name	Base Address
<PORTn_base>	FFC1 0000_H
<JPORT0_base>	FFC2 0000_H

2C.6.4 Clock Supply

The clock supply to ports is shown in the following table.

Table 2C.10 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Port	Register access clock	CPUCLK_UL

2C.7 Port Functions

This product has various pins for input/output ports. The ports are organized in port groups.

The RH850/F1KM also has several control registers to enable pins to be used as other than general-purpose input/output pins.

For a description of the terms pin, port, and port group, see **Section 2C.7.2, Terms**.

2C.7.1 Functional Overview

- All the port settings can be specified individually.
- The maximum number of bits (pins) in a port is 16.
- The output level of any pin can be set independently without affecting the other pins in the same port.
- Input buffers are enabled through registers settings.
- Pin level is read by dedicated port-pin-read register (PPR)
- All possible port functions are shown in the tables listed below.

  Table 2C.39, JTAG Port 0 (JP0), **Table 2C.41, Port 0 (P0)**, **Table 2C.43, Port 8 (P8)**, **Table 2C.45, Port 9 (P9)**, **Table 2C.47, Port 10 (P10)**, **Table 2C.49, Port 11 (P11)**, **Table 2C.51, Analog Port 0 (AP0)** and **Section 2C.9.2, Pin Function Configuration**.

CAUTION

Some input or output functions may be assigned to more than one port. Only activate a given function on a single pin. Do not activate a function on multiple pins at the same time. This also applies in cases where multiple peripheral functions are assigned to a single multiplexed function and only one of these functions is used.

[Example]

INTP0 is assigned to the following pins on this device. However, the INTP0 function should not be activated on more than one pin. After activating the function on one pin, do not activate it on another.

- JP0_0 (1st input alternative function)
- P0_1 (2nd, 3rd input alternative function)
- P10_0 (2nd input alternative function)

In the above case, when the 1st input alternative function (INTP0) of JP0_0 is selected, using the 2nd input alternative function (CAN0RX/INTP0) of P0_1 only for the CAN signal is also prohibited.

2C.7.2 Terms

The following terms are used in this section:

Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

Port mode and ports

A pin in port mode works as a general-purpose input/output pin. It is then called “port”.

The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions, such as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that two different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function of the pin at that time.

2C.7.2.1 JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging.

JTAG port group registers and bit names are prefixed by a “J”. For example, JP0 denotes JTAG port group 0, and JPM0.JPM0_m denotes the JPM0_m port mode bit of the JPM0 port mode register.

NOTE

In this section, the descriptions about all ports and their registers other than PFCAEn and PIPCn apply to the JTAG port unless otherwise specified.

2C.7.3 Overview of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCn_m bit = 0)

  A pin in port mode operates as a general-purpose input/output pin. The I/O mode is selected by setting the PMn.PMn_m bit.

- Software I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 0)

  In this mode, the pins operate as alternative functions. The I/O mode is selected by setting the PMn.PMn_m bit.

- Direct I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 1)

  In this mode, the pins operate as alternative functions. Unlike the software I/O control alternative mode, however, the I/O mode is directly controlled by the alternative function.

An overview of the register settings is given in the tables below.

Table 2C.11 Pin Function Configuration (Overview)

	Bit			
Mode	PMCn_m	PMn_m	PIPCn_m	I/O
Port mode	0	0	X	O
		1[*1]		I
Software I/O control alternative mode	1	0	0	O
		1	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

Note 1. The input buffer must be enabled (PIBCn_m bit = 1).

- Software I/O control alternative mode (PIPCn.PIPCn_m bit = 0)
  - Output (PMn_m bit = 0): Alternative output mode 1 to Alternative output mode 7
  - Input (PMn_m bit = 1): Alternative input mode 1 to Alternative input mode 7
- Direct I/O control alternative mode (PIPCn.PIPCn_m bit = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 7 and Alternative input mode 1 to Alternative input mode 7 is directly selected by the alternative function.

Table 2C.12 Alternative Mode Selection Overview (PMCn.PMCn_m Bit = 1)

	Register					
Mode	PIPC*[1]	PM*[1]	PFCAE	PFCE	PFC	I/O
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)		1				I
Alternative output mode 2 (ALT-OUT2)		0			1	O
Alternative input mode 2 (ALT-IN2)		1				I
Alternative output mode 3 (ALT-OUT3)		0		1	0	O
Alternative input mode 3 (ALT-IN3)		1				I
Alternative output mode 4 (ALT-OUT4)		0			1	O
Alternative input mode 4 (ALT-IN4)		1				I
Alternative output mode 5 (ALT-OUT5)		0	1	0	0	O
Alternative input mode 5 (ALT-IN5)		1				I
Alternative output mode 6 (ALT-OUT6)		0			1	O
Alternative input mode 6 (ALT-IN6)		1				I
Alternative output mode 7 (ALT-OUT7)		0		1	0	O
Alternative input mode 7 (ALT-IN7)		1				I
Other than the above	Setting prohibited					

Note 1. If PIPCn.PIPCn_m bit = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

If a pin is in alternative mode (PMCn.PMCn_m bit = 1), one of up to seven alternative functions can be selected for that pin by using the PFCn, PFCEn, and PFCAEn registers.

2C.7.4 Pin Data Input/Output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.

2C.7.4.1 Output Data

In the port mode (PMCn.PMCn_m bit = 0), the value of the Pn.Pn_m bit is output from the Pn_m pin.

2C.7.4.2 Input Data

When the PPRn register is read, either the value of the Pn_m pin, the value of the Pn.Pn_m bit, or the value output by the alternative function is returned.

Which value is returned depends on the pin mode and setting of several control bits. The different PPRn read modes are shown in the table below.

Table 2C.13 PPRn_m Read Values

PMC n_m	PM n_m	PIBC n_m	PIPC n_m	PODC n_m	Mode	PPRn_m Read Value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m bit
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m bit*1
				1	Port open-drain output	
1	1	X	0	X	Software I/O control alternative input	Pn_m pin
	0			0	Software I/O control alternative push-pull output	Output signal from the alternative function*1
				1	Software I/O control alternative open-drain output	
	X		1	0	Direct I/O control alternative input or push-pull output	I/O port in alternative mode: • Input: Pn_m pin • Output: Output signal from the alternative function*1
				1	Direct I/O control alternative input or open-drain output	

Note 1. When PBDCn_m = 1, the level of the Pn_m pin is returned by the PPRn_m bit.

The control registers in the above table have the following effects:

- PMCn.PMCn_m bit

  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).

- PMn.PMn_m bit

  This bit selects input (PMn_m = 1) or output (PMn_m = 0) when the port mode (PMCn_m = 0) and software I/O control alternative mode (PMCn_m = 1, PIPCn_m = 0) have been selected.

- PIBCn.PIBCn_m bit

  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). If the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit; otherwise the Pn_m pin level is returned.

- PIPCn.PIPCn_m bit

  This bit selects software I/O control alternative mode or direct I/O control alternative mode.

- PODCn.PODCn_m bit

  This bit selects push-pull output (PODCn_m = 0) or open-drain output (PODCn_m = 1).

- PBDCn.PBDCn_m bit

  In output mode, when this bit is set to 1, the pin enters the bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

CAUTION

When using Pn_m as an alternative output function (PMCn.PMCn_m bit = 1, PMn.PMn_m bit = 0), the level of the Pn_m pin can be read at the PPRn.PPRn_m bit by enabling bidirectional mode (PBDCn.PBDCn_m bit = 1).
Note, however, that the level of the Pn_m pin will be input to the alternative input function that the Pn_m pin is being used as.

2C.7.4.3 Writing to the Pn Register

The data to be output via port Pn_m in port mode (PMCn.PMCn_m bit = 0) is held in port register Pn. Pn data can be overwritten in two ways:

- By writing data directly to the Pn register.

  In this case, new data can be written directly to the Pn register.

- By performing an indirect bitwise operation (a “set”, “reset”, or “not” operation) on the Pn register.

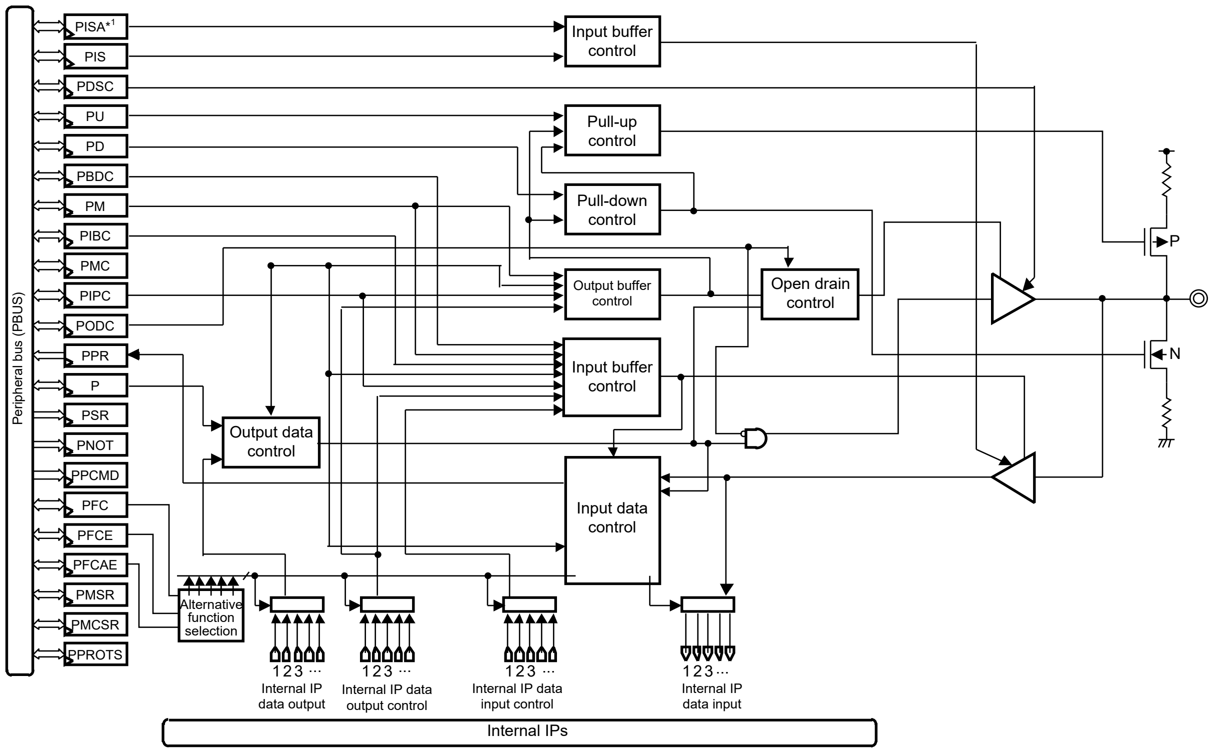
  An indirect bitwise operation (“set”, “reset”, or “not”) can be performed on the Pn register by using the following two registers:

  - Port Set/Reset register PSRn
    If the PSRn.PSRn (m + 16) bit = 1, the value of the Pn.Pn_m bit is determined by the value of the PSRn.PSRn_m bit.
    In other words, the Pn_m bit can be set or reset without writing directly to the Pn register.
  - Port NOT register PNOTn
    By setting PNOTn.PNOTn_m bit to 1, the Pn.Pn_m bit can be inverted without writing directly to the Pn register.

An indirect bitwise operation on the Pn register (“set”, “reset”, or “not”) has no effect on the bits that do not need to be updated, allowing you to overwrite only the bit or bits that need to be overwritten.

2C.8 Schematic View of Port Control

The following figure is a schematic view of the port control functions.



Note 1. There is no PISAn register in RH850/F1KM-S1.

Figure 2C.5 Schematic View of Port Control

CAUTION

Use documented alternative functions only. The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2C.9 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then describes all registers in detail. The configuration registers are grouped as follows:

- **Section 2C.9.2, Pin Function Configuration**
- **Section 2C.9.3, Pin Data Input/Output**
- **Section 2C.9.4, Configuration of Electrical Characteristics**

2C.9.1 Overview

The following registers are used for setting the individual pins of the port groups.

For details on <PORTn_base> and <JPORT0_base>, see **Section 2C.6.3, Register Base Addresses**.

Table 2C.14 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin function configuration			
PORT	Port mode control register	PMCn	<PORTn_base> + 0400$_H$ + n × 4
JTAG		JPMC0	<JPORT0_base> + 0040$_H$
PORT	Port mode control set/reset register	PMCSRn	<PORTn_base> + 0900$_H$ + n × 4
JTAG		JPMCSR0	<JPORT0_base> + 0090$_H$
PORT	Port IP control register	PIPCn	<PORTn_base> + 4200$_H$ + n × 4
PORT	Port mode register	PMn	<PORTn_base> + 0300$_H$ + n × 4
		APMn	<PORTn_base> + 03C8$_H$ + n × 4
JTAG		JPM0	<JPORT0_base> + 0030$_H$
PORT	Port mode set/reset register	PMSRn	<PORTn_base> + 0800$_H$ + n × 4
		APMSRn	<PORTn_base> + 08C8$_H$ + n × 4
JTAG		JPMSR0	<JPORT0_base> + 0080$_H$
PORT	Port input buffer control register	PIBCn	<PORTn_base> + 4000$_H$ + n × 4
		APIBCn	<PORTn_base> + 40C8$_H$ + n × 4
JTAG		JPIBC0	<JPORT0_base> + 0400$_H$
PORT	Port function control register	PFCn	<PORTn_base> + 0500$_H$ + n × 4
JTAG		JPFC0	<JPORT0_base> + 0050$_H$
PORT	Port function control expansion register	PFCEn	<PORTn_base> + 0600$_H$ + n × 4
JTAG		JPFCE0	<JPORT0_base> + 0060$_H$
PORT	Port function control additional expansion register	PFCAEn	<PORTn_base> + 0A00$_H$ + n × 4
Pin data input/output			
PORT	Port bidirection control register	PBDCn	<PORTn_base> + 4100$_H$ + n × 4
		APBDCn	<PORTn_base> + 41C8$_H$ + n × 4
JTAG		JPBDC0	<JPORT0_base> + 0410$_H$
PORT	Port pin read register	PPRn	<PORTn_base> + 0200$_H$ + n × 4
		APPRn	<PORTn_base> + 02C8$_H$ + n × 4
JTAG		JPPR0	<JPORT0_base> + 0020$_H$
PORT	Port register	Pn	<PORTn_base> + 0000$_H$ + n × 4
		APn	<PORTn_base> + 00C8$_H$ + n × 4
JTAG		JP0	<JPORT0_base> + 0000$_H$

Table 2C.14 Port Group Configuration Registers

Module Name	Register Name	Symbol	Address
Pin data input/output			
PORT	Port NOT register	PNOTn	<PORTn_base> + 0700_H + n × 4
		APNOTn	<PORTn_base> + $07C8_H$ + n × 4
JTAG		JPNOT0	<JPORT0_base> + 0070_H
PORT	Port set/reset register	PSRn	<PORTn_base> + 0100_H + n × 4
		APSRn	<PORTn_base> + $01C8_H$ + n × 4
JTAG		JPSR0	<JPORT0_base> + 0010_H
Configuration of electrical characteristics			
PORT	Pull-up option register	PUn	<PORTn_base> + 4300_H + n × 4
JTAG		JPU0	<JPORT0_base> + 0430_H
PORT	Pull-down option register	PDn	<PORTn_base> + 4400_H + n × 4
JTAG		JPD0	<JPORT0_base> + 0440_H
PORT	Port drive strength control register	PDSCn	<PORTn_base> + 4600_H + n × 4
JTAG		JPDSC0	<JPORT0_base> + 0460_H
PORT	Port open drain control register	PODCn	<PORTn_base> + 4500_H + n × 4
JTAG		JPODC0	<JPORT0_base> + 0450_H
PORT	Port input buffer selection register	PISn	<PORTn_base> + 4700_H + n × 4
JTAG		JPIS0	<JPORT0_base> + 0470_H
JTAG	Port input buffer selection advanced register	JPISA0	<JPORT0_base> + $04A0_H$
Port register protection			
PORT	Port protection command register	PPCMDn	<PORTn_base> + $4C00_H$ + n × 4
JTAG		JPPCMD0	<JPORT0_base> + $04C0_H$
PORT	Port protection status register	PPROTSn	<PORTn_base> + $4B00_H$ + n × 4
JTAG		JPPROTS0	<JPORT0_base> + $04B0_H$

Index n

In **Table 2C.14, Port Group Configuration Registers**, the index “n” in register symbols denotes the actual indices of the individual port groups. For example, PMCn generically indicates a port mode control register for port group n (Pn). The values for n differ according to the number of pins on the device in the way shown in **Table 2C.15, Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols**.

Table 2C.15 Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols

Number of Pins on the Device	Port Groups	Values for “n”
48 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
64 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
80 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
100 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0

JTAG port registers

JTAG port registers are not explicitly described in the following register descriptions.

All descriptions (except for those of the PFCAEn register and PIPCn register) apply to JTAG port registers. Note, however, that the JTAG port register base address differs from that of regular ports.

Value after reset

The values after reset depend on the ports. For the values after reset, see the register descriptions in the following pages.

2C.9.2 Pin Function Configuration

2C.9.2.1 PMCn / JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access: PMCn: This register can be read or written in 16-bit units.

JPMC0: This register can be read or written in 8-bit units.

Address: PMCn: <PORTn_base> + 0400_H + n × 4 (n = 0, 8, 9, 10, 11)

JPMC0: <JPORT0_base> + 0040_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC n_15	PMC n_14	PMC n_13	PMC n_12	PMC n_11	PMC n_10	PMC n_9	PMC n_8	PMC n_7	PMC n_6	PMC n_5	PMC n_4	PMC n_3	PMC n_2	PMC n_1	PMC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.16 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

CAUTIONS

1. I/O is not controlled by only setting alternative mode (PMCn.PMCn_m bit = 1). If the alternative function requires direct I/O control, also set the PIPCn.PIPCn_m bit to 1.
2. If a port is to be used as an input pin in alternative mode, the signals from some pins will pass through a noise filter. These pins may require the setting of the FCLA0CTLm_<*name*>, DNFA<*name*>CTL and the DNFA<*name*>EN register. For details, see **Section 2C.12, Noise Filter & Edge/Level Detector**, and **Section 2C.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].

2C.9.2.2 PMCSRn / JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value of PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access:	PMCSRn: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_H$. Reading bits 15 to 0 returns the value of register PMCn. JPMCSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000$_H$. Reading bits 7 to 0 returns the value of register JPMC0.
Address:	PMCSRn: <PORTn_base> + 0900$_H$ + n × 4 (n = 0, 8, 9, 10, 11) JPMCSR0: <JPORT0_base> + 0090$_H$^{*1}
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMC SRn_31	PMC SRn_30	PMC SRn_29	PMC SRn_28	PMC SRn_27	PMC SRn_26	PMC SRn_25	PMC SRn_24	PMC SRn_23	PMC SRn_22	PMC SRn_21	PMC SRn_20	PMC SRn_19	PMC SRn_18	PMC SRn_17	PMC SRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC SRn_15	PMC SRn_14	PMC SRn_13	PMC SRn_12	PMC SRn_11	PMC SRn_10	PMC SRn_9	PMC SRn_8	PMC SRn_7	PMC SRn_6	PMC SRn_5	PMC SRn_4	PMC SRn_3	PMC SRn_2	PMC SRn_1	PMC SRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.17 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMCSRn_m (PMCSRn_[15:0]) is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m. 1: PMCn_m is PMCSRn_m. Example: If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_[15:0]	Data bits that specify the value of PMCn_m if PMCSRn_m of the corresponding upper bit (PMCSRn_[31:16]) is 1. 0: PMCn_m is 0. 1: PMCn_m is 1.

NOTE

The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].

2C.9.2.3 PIPCn — Port IP Control Register

This register specifies whether the I/O direction of the Pn_m pin is controlled by the port mode register PMn.PMn_m or by an alternative function.

If the Pn_m pin is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn_m setting.

Regarding the alternative functions for which the PIPC register must be set, see **Section 2C.11, Port (Special I/O) Function Overview**.

Access: This register can be read or written in 16-bit units.

Address: PIPCn: <PORTn_base> + 4200_H + n × 4 (n = 0, 10, 11) *1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.42, Control Registers (P0)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.18 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

2C.9.2.4 PMn / APMn / JPM0 — Port Mode Register

This register specifies whether the individual pins of the port group n are in input mode or in output mode.

Access:	PMn, APMn: These registers can be read or written in 16-bit units. JPM0: This register can be read or written in 8-bit units.
Address:	PMn: <PORTn_base> + 0300_H + n × 4 (n = 0, 8, 9, 10, 11) APMn: <PORTn_base> + $03C8_H$ (n = 0) JPM0: <JPORT0_base> + 0030_H*1
Value after reset:	$FFFF_H$*2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	1	1	1	1	1	1	1	1	1	1*3	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Note 2. The PM8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8 register is $FFBF_H$.
When the OPBT0.RESETOUTEN = 0, the PM8 register is $FFFF_H$.

Note 3. The PM8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PM8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PM8_6 bit is 1.

Table 2C.19 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

NOTES

1. To use a port in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).
2. By default, PMn.PMn_m specifies the I/O direction in port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m=1), since PIPCn.PIPCn_m = 0 (I/O mode is controlled by PMn.PMn_m) after reset.
3. The control bits of the analog port register (APMn) are APMn_[15:0].
4. The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].

2C.9.2.5 PMSRn / APMSRn / JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write data to the PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn_m is set by the corresponding bit in the lower 16 bits of PMSRn.

Access: PMSRn, APMSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_{\text{H}}$.

Reading bits 15 to 0 returns the value of registers PMn and APMn.

JPMSR0: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000$_{\text{H}}$. Bits 15 to 8 are read as FF$_{\text{H}}$. Reading bits 7 to 0 returns the value of register JPM0.

Address: PMSRn: <PORTn_base> + 0800$_{\text{H}}$ + n × 4 (n = 0, 8, 9, 10, 11)

APMSRn: <PORTn_base> + 08C8$_{\text{H}}$ (n = 0)

JPMSR0: <JPORT0_base> + 0080$_{\text{H}}$^{*1}

Value after reset: 0000 FFFF$_{\text{H}}$^{*2}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
Value after reset	1	1	1	1	1	1	1	1	1	1^{*3}	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Note 2. The PMSR8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8 register is 0000 FFBF$_{\text{H}}$.
When the OPBT0.RESETOUTEN = 0, the PMSR8 register is 0000 FFFF$_{\text{H}}$.

Note 3. The PMSR8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PMSR8_6 bit is 0.
When the OPBT0.RESETOUTEN = 0, the PMSR8_6 bit is 1.

Table 2C.20 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMSRn_m (PMSRn_[15:0]) is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m. Example: If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Data bits that specify the value of PMn_m if PMSRn_m of the corresponding upper bit (PMSRn_[31:16]) is 1. 0: PMn_m is 0. 1: PMn_m is 1.

NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[31:0].
2. The control bits of the analog port mode set/reset register (APMSRn) are APMSRn_[31:0].

2C.9.2.6 PIBCn / APIBCn / JPIBC0 — Port Input Buffer Control Register

In input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables the port pin's input buffer.

Access:	PIBCn, APIBCn: These registers can be read or written in 16-bit units.
	JPIBC0: This register can be read or written in 8-bit units.
Address:	PIBCn: <PORTn_base> + 4000_H + n × 4 (n = 0, 8, 9, 10, 11)
	APIBCn: <PORTn_base> + $40C8_H$ (n = 0)
	JPIBC0: <JPORT0_base> + 0400_H*1
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBCn_15	PIBCn_14	PIBCn_13	PIBCn_12	PIBCn_11	PIBCn_10	PIBCn_9	PIBCn_8	PIBCn_7	PIBCn_6	PIBCn_5	PIBCn_4	PIBCn_3	PIBCn_2	PIBCn_1	PIBCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.21 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled

NOTES

1. When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].

CAUTION

Settings in this register are overruled in bidirectional mode (PBDCn.PBDCn_m = 1).

2C.9.2.7 PFCn / JPFC0 — Port Function Control Register

This register, together with register PFCEn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is selected by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access:	PFCn: This register can be read or written in 16-bit units. JPFC0: This register can be read or written in 8-bit units.
Address:	PFCn: <PORTn_base> + 0500_H + n × 4 (n = 0, 8, 9, 10, 11) JPFC0: <JPORT0_base> + 0050_H[*1]
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFC n_15	PFC n_14	PFC n_13	PFC n_12	PFC n_11	PFC n_10	PFC n_9	PFC n_8	PFC n_7	PFC n_6	PFC n_5	PFC n_4	PFC n_3	PFC n_2	PFC n_1	PFC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.22 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2C.25, Setting Alternative Functions**.

NOTE

The control bits of the JTAG port function control register (JPFC0) are JPFC0_[7:0].

2C.9.2.8 PFCEn / JPFCE0 — Port Function Control Expansion Register

This register, together with register PFCn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCEn: This register can be read or written in 16-bit units.

JPFCE0: This register can be read or written in 8-bit units.

Address: PFCEn: <PORTn_base> + 0600_H + n × 4 (n = 0, 8, 9, 10, 11)

JPFCE0: <JPORT0_base> + 0060_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCE n_15	PFCE n_14	PFCE n_13	PFCE n_12	PFCE n_11	PFCE n_10	PFCE n_9	PFCE n_8	PFCE n_7	PFCE n_6	PFCE n_5	PFCE n_4	PFCE n_3	PFCE n_2	PFCE n_1	PFCE n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.23 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2C.25, Setting Alternative Functions**.

NOTE

The control bits of the JTAG port function control register (JPFCE0) are JPFCE0_[7:0].

2C.9.2.9 PFCAEn — Port Function Control Additional Expansion Register

This register selects the alternative peripheral functions together with PFCEn, PFCn registers.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCAEn: This register can be read or written in 16-bit units.

Address: PFCAEn: <PORTn_base> + $0A00_H$ + n × 4 (n = 0, 8, 9, 10, 11)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAE n_15	PFCAE n_14	PFCAE n_13	PFCAE n_12	PFCAE n_11	PFCAE n_10	PFCAE n_9	PFCAE n_8	PFCAE n_7	PFCAE n_6	PFCAE n_5	PFCAE n_4	PFCAE n_3	PFCAE n_2	PFCAE n_1	PFCAE n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.24 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2C.25, Setting Alternative Functions**.

Table 2C.25 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1
			0	Alternative output mode 1
		1	1	Alternative input mode 2
			0	Alternative output mode 2
	1	0	1	Alternative input mode 3
			0	Alternative output mode 3
		1	1	Alternative input mode 4
			0	Alternative output mode 4
1	0	0	1	Alternative input mode 5
			0	Alternative output mode 5
		1	1	Alternative input mode 6
			0	Alternative output mode 6
	1	0	1	Alternative input mode 7
			0	Alternative output mode 7
		1	X	Setting prohibited

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m, or PFCAEn_m bit, set the PMCn_m bit to “1”.
- With this product, the I/O of some functions is assigned to two or more pins, but a specific pin function can only be set to one pin at a time. Setting the same pin function to two or more pins at the same time is prohibited.
  For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as a pin function other than b.

NOTE

For more details on the assignment of each function, see **Sections 2C.10.1, JTAG Port 0 (JP0)** to **2C.10.7, Analog Port 0 (AP0)**.

2C.9.3 Pin Data Input/Output

2C.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

This register enables the input buffer in output mode and sets the port to bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

Access:	PBDCn, APBDCn: These registers can be read or written in 16-bit units. JPBDC0: This register can be read or written in 8-bit units.
Address:	PBDCn: <PORTn_base> + 4100_H + n × 4 (n = 0, 8, 9, 10, 11) APBDCn: <PORTn_base> + $41C8_H$ (n = 0) JPBDC0: <JPORT0_base> + 0410_H[*1]
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.26 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin. 0: Bidirectional mode disabled 1: Bidirectional mode enabled

CAUTION

When the Pn_m port is used for the alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read from PPRn.PPRn_m by enabling the bidirectional mode (PBDCn.PBDCn_m = 1).

However, output of that alternative output function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn_m, PFCEn.PFCEn_m, and PFCAEn.PFCAEn_m). If the alternative input function in question is being used by another pin, the alternative input function is not guaranteed.

NOTE

The control bits of the JTAG port bidirection control register (JPBDC0) are JPBDC0_[7:0].

2C.9.3.2 PPRn / APPRn / JPPR0 — Port Pin Read Register

This register reflects the actual level of the Pn_m pin, whether it is the value of the Pn.Pn_m bit or the level of an alternative output function.

Access: PPRn, APPRn: These registers are read-only registers that can be read in 16-bit units.

JPPR0: This register is a read-only register that can be read in 8-bit units.

Address: PPRn: <PORTn_base> + 0200_H + n × 4 (n = 0, 8, 9, 10, 11)

APPRn: <PORTn_base> + $02C8_H$ (n = 0)

JPPR0: <JPORT0_base> + 0020_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.27 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	The Pn_m Pin, Pn.Pn_m value or alternative function output.

NOTES

1. For the read values of the PPRn register, see **Section 2C.7.4, Pin Data Input/Output**.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].

2C.9.3.3 Pn / APn / JP0 — Port Register

This register holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access:	Pn, APn: These registers can be read or written in 16-bit units. JP0: This register can be read or written in 8-bit units.
Address:	Pn: <PORTn_base> + 0000_H + n × 4 (n = 0, 8, 9, 10, 11) APn: <PORTn_base> + $00C8_H$ (n = 0) JP0: <JPORT0_base> + 0000_H^{*1}
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.28 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

NOTE

The control bits of the JTAG port register (JP0) are JP0_[7:0].

2C.9.3.4 PNOTn / APNOTn / JPNOT0 — Port NOT Register

This register allows the Pn_m bit of the port register Pn to be inverted without directly writing to Pn.

Access: PNOTn, APNOTn: These registers are write-only registers that can be written in 16-bit units. When read, 0000_H is returned.

JPNOT0: This register is a write-only register that can be written in 8-bit units. When read, 00_H is returned.

Address: PNOTn: <PORTn_base> + 0700_H + n × 4 (n = 0,8, 9, 10, 11)

APNOTn: <PORTn_base> + $07C8_H$ (n = 0)

JPNOT0: <JPORT0_base> + 0070_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOT n_15	PNOT n_14	PNOT n_13	PNOT n_12	PNOT n_11	PNOT n_10	PNOT n_9	PNOT n_8	PNOT n_7	PNOT n_6	PNOT n_5	PNOT n_4	PNOT n_3	PNOT n_2	PNOT n_1	PNOT n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.29 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted. 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted ($\overline{\text{Pn\_m}}$ → Pn_m)

NOTE

The control bits of the JTAG port NOT register are JPNOT0_[7:0].

2C.9.3.5 PSRn / APSRn / JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to the Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn_m is set by the corresponding bit in the lower 16 bits of PSRn.

Access: PSRn, APSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of registers Pn and APn.

JPSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JP0.

Address: PSRn: <PORTn_base> + 0100_H + n × 4 (n = 0, 8, 9, 10, 11)

APSRn: <PORTn_base> + $01C8_H$ (n = 0)

JPSR0: <JPORT0_base> + 0010_H*1

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, **Table 2C.50, Control Registers (P11)** and **Table 2C.52, Control Registers (AP0)**.

Table 2C.30 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m (PSRn_[15:0]) is written to Pn_m. 0: Pn_m is not affected by PSRn_m 1: Pn_m is PSRn_m Example: If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit (PSRn_[31:16]) PSRn_m is 1. 0: Pn_m = 0 1: Pn_m = 1

NOTE

The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].

2C.9.4 Configuration of Electrical Characteristics

2C.9.4.1 PUn / JPU0 — Pull-Up Option Register

This register specifies whether an internal pull-up resistor is connected to an input pin.

Access:	PUn: This register can be read or written in 16-bit units. JPU0: This register can be read or written in 8-bit units.
Address:	PUn: <PORTn_base> + 4300_H + n × 4 (n = 0, 8, 9, 10, 11) JPU0: <JPORT0_base> + $0430_H{}^{*1}$
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.31 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an internal pull-up resistor is connected to the corresponding pin. 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The pull-up resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].

2C.9.4.2 PDn / JPD0 — Pull-Down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin.

Access: PDn: This register can be read or written in 16-bit units.

JPD0: This register can be read or written in 8-bit units.

Address: PDn: <PORTn_base> + 4400_H + n × 4 (n = 0, 8, 9, 10, 11)

JPD0: <JPORT0_base> + $0440_H{}^{*1}$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.32 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin. 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].

2C.9.4.3 PDSCn / JPDSC0 — Port Drive Strength Control Register

This register specifies the output driver strength of the port pin. This function selects the fast mode (high drive strength) or slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.
Regarding the alternative functions for which the PDSC register needs to be set, see **Section 2C.11.3.3, Output Buffer Control (PDSC)**.

Access: PDSCn, JPDSC0: These registers can be read or written in 32-bit units.

Address: PDSCn: <PORTn_base> + 4600_H + n × 4 (n = 0, 10, 11)

JPDSC0: <JPORT0_base> + 0460_H ^{*1}

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSC n_15	PDSC n_14	PDSC n_13	PDSC n_12	PDSC n_11	PDSC n_10	PDSC n_9	PDSC n_8	PDSC n_7	PDSC n_6	PDSC n_5	PDSC n_4	PDSC n_3	PDSC n_2	PDSC n_1	PDSC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.33 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Lower drive strength (when the frequency output from the pin is 10 MHz or below) 1: High drive strength (when the frequency output from the pin is 40 MHz or less).

NOTE

The control bits of the JTAG port drive strength control register (JPDSC0) are JPDSC0_[31:0].

2C.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: PODCn, JPODC0: These registers can be read or written in 32-bit units.

Address: PODCn: <PORTn_base> + 4500_H + n × 4 (n = 0, 8, 9, 10, 11)

JPODC0: <JPORT0_base> + 0450_H[*1]

Value after reset: 0000 0000_H[*2]

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0[*3]	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Note 2. The PODC8 register is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8 register is 0000 0040_H.
When the OPBT0.RESETOUTEN = 0, the PODC8 register is 0000 0000_H.

Note 3. The PODC8_6 bit is as follows.
When the OPBT0.RESETOUTEN = 1, the PODC8_6 bit is 1.
When the OPBT0.RESETOUTEN = 0, the PODC8_6 bit is 0.

Table 2C.34 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

NOTE

The control bits of the JTAG port open drain control register (JPODC0) are JPODC0_[31:0].

2C.9.4.5 PISn/JPIS0 — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

Access: PISn: This register can be read or written in 16-bit units.
JPIS0: This register can be read or written in 8-bit units.

Address: PISn: <PORTn_base> + 4700_H + n × 4 (n = 0, 8, 9, 10, 11)
JPIS0: <JPORT0_base> + 0470_H*1

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**, **Table 2C.42, Control Registers (P0)**, **Table 2C.44, Control Registers (P8)**, **Table 2C.46, Control Registers (P9)**, **Table 2C.48, Control Registers (P10)**, and **Table 2C.50, Control Registers (P11)**.

Table 2C.35 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer Characteristics: 0: Type 1 (SHMT1) 1: Type 2 (SHMT4)

NOTES

1. Details of the definition of type 1 and type 2 are given in **Section 2C.11.3.2, Input Buffer Control (PISn/JPIS0, JPISA0)**. For details, also see **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for input buffer characteristics.
2. The control bits of the JTAG port input buffer selection register (JPIS0) are JPIS0_[7:0]

2C.9.4.6 JPISA0 — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics.

Access: JPISA0: This register can be read or written in 8-bit units.

Address: JPISA0: <JPORT0_base> + $04A0_H$*1

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	JPISA0_3	JPISA0_2	—	JPISA0_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2C.10, Port (General I/O) Function Overview**: **Table 2C.40, Control Registers (JP0)**.

Table 2C.36 JPISA0 Register Contents

Bit Position	Bit Name	Function
7 to 4, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2, 0	JPISA0_[3, 2, 0]	Specifies the input buffer characteristics: 0: Type 2 (SHMT4) 1: Type 5 (TTL)

Table 2C.37 JTAG Port Input Selection Advanced Register Contents

JPISA0	JPIS0	Function
0	0	Type 1 input buffer is selected (SHMT1)
	1	Type 2 input buffer is selected (SHMT4)
1	X	Type 5 input buffer is selected (TTL)

NOTE

Details of the definition of type 2 and type 5 are given in **Section 2C.11.3.2, Input Buffer Control (PISn/JPIS0, JPISA0)**. For details, also see **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for input buffer characteristics.

2C.9.5 Port Register Protection

RH850/F1KM has Port Protection Command Registers (PPCMDn) and Port Protection Status Registers (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 5, Write-Protected Registers**.

2C.9.6 Flowchart Examples for Port Settings

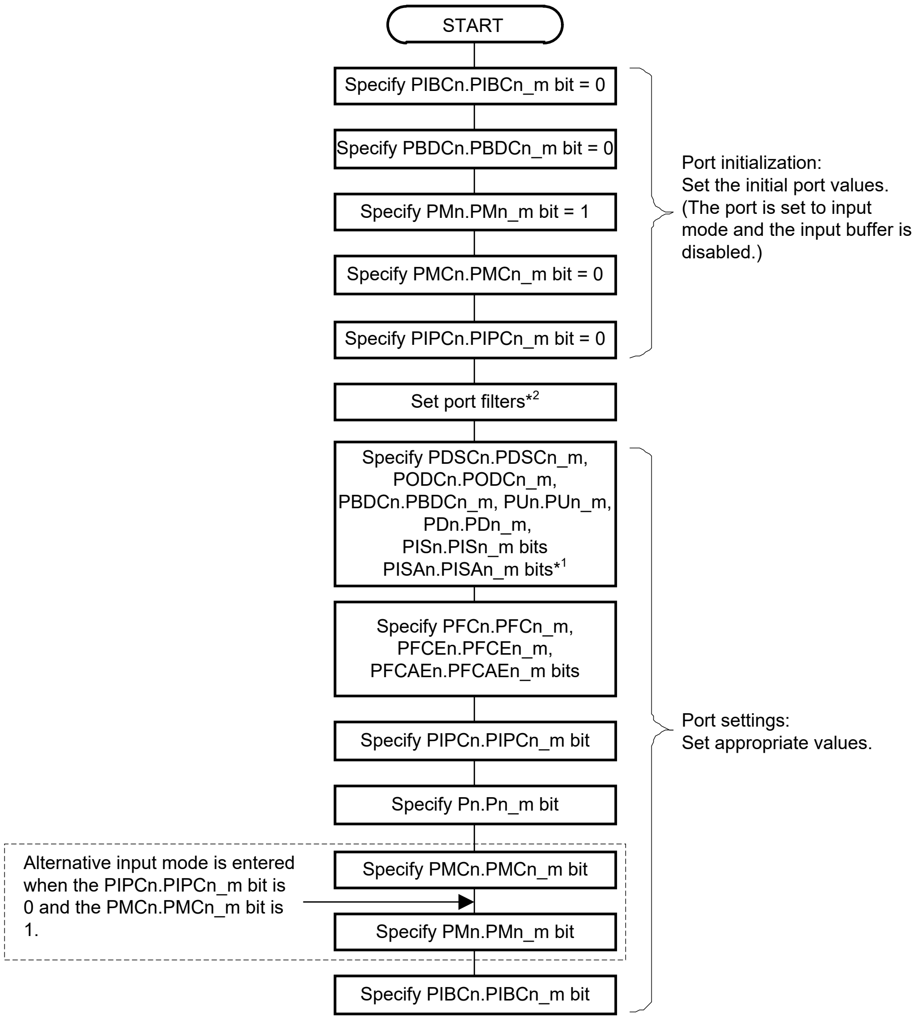
Examples of the port settings are shown in the flowchart below.

CAUTION

If the port is set to the PIPCn.PIPCn_m bit = 0 and alternative output mode, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCn_m bit is set to 1 and when the PMn.PMn_m bit is set to 0. If an interrupt- related signal is specified as an alternate function of the port, the mode will temporarily become the alternative input mode, so either disable the interrupt in question, or specify that the interrupt is ignored.

2C.9.6.1 Batch Setting

An example of specifying batch port group settings is shown in the flowchart below.



Note 1. There is no PISAn.PISAn_m bit in RH850/F1KM-S1.

Note 2. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-5 and INTP10-12 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

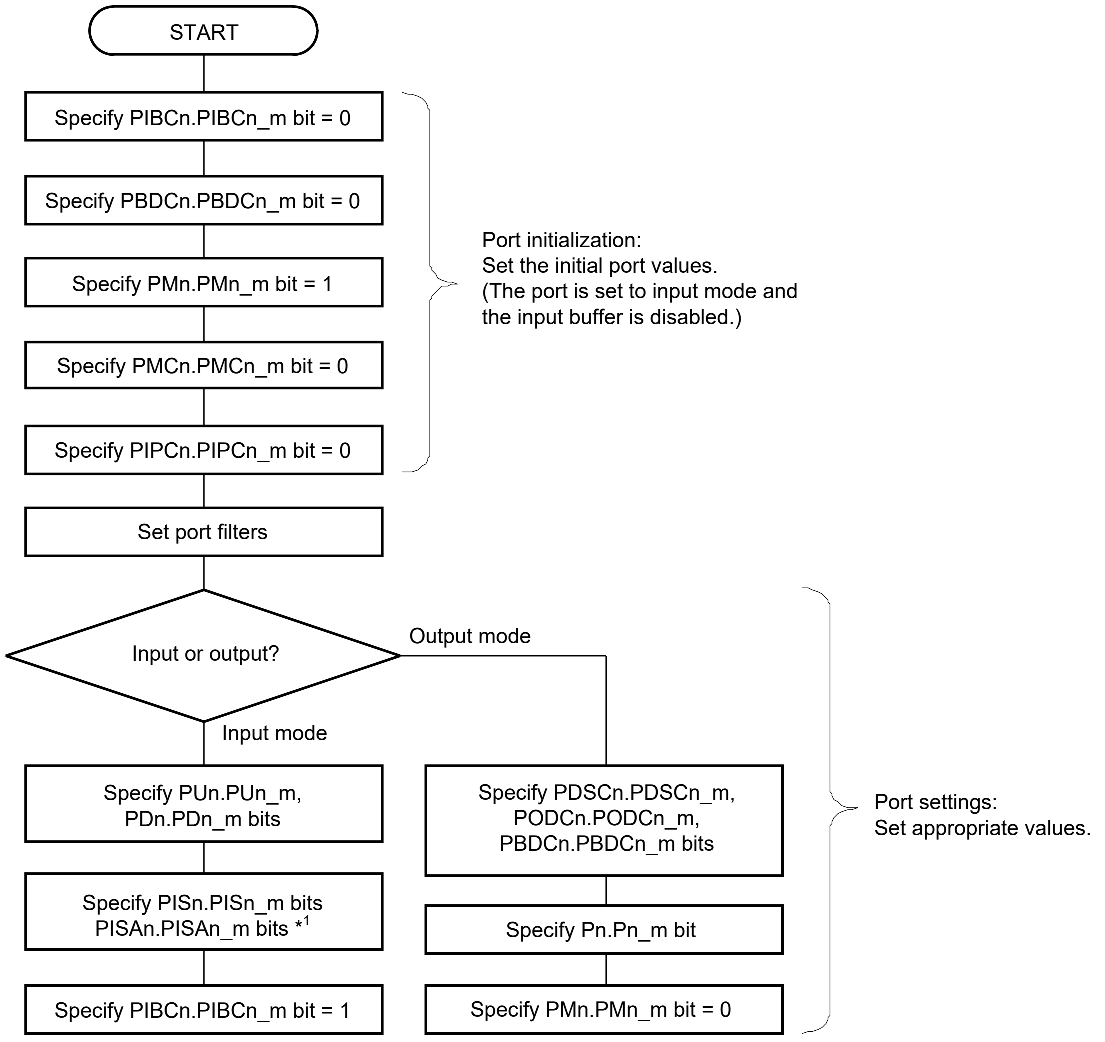
In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2C.6 Example of Port Settings (When Specified in Batch)

2C.9.6.2 Individual Settings

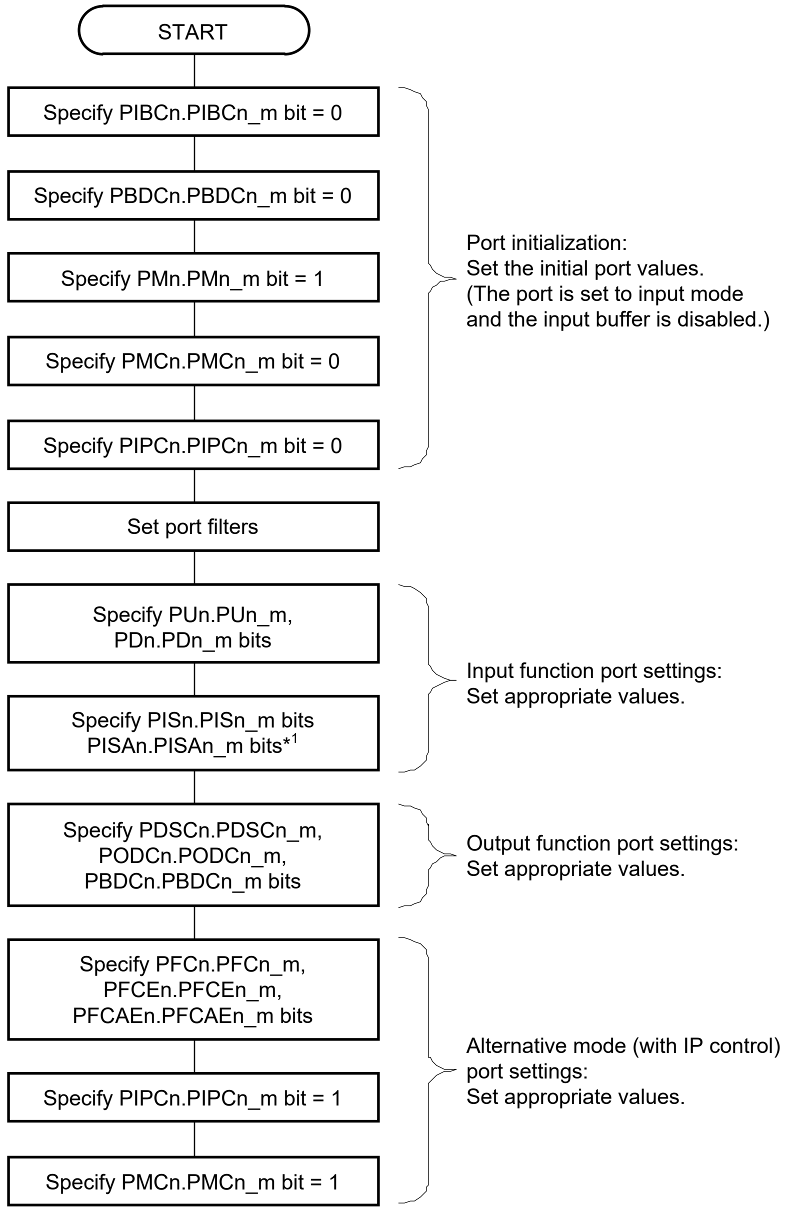
An example of specifying individual port settings is shown in the flowchart below.



Note 1. There is no PISAn.PISAn_m bit in RH850/F1KM-S1.

Figure 2C.7 Example of Port Settings (in Port Mode)

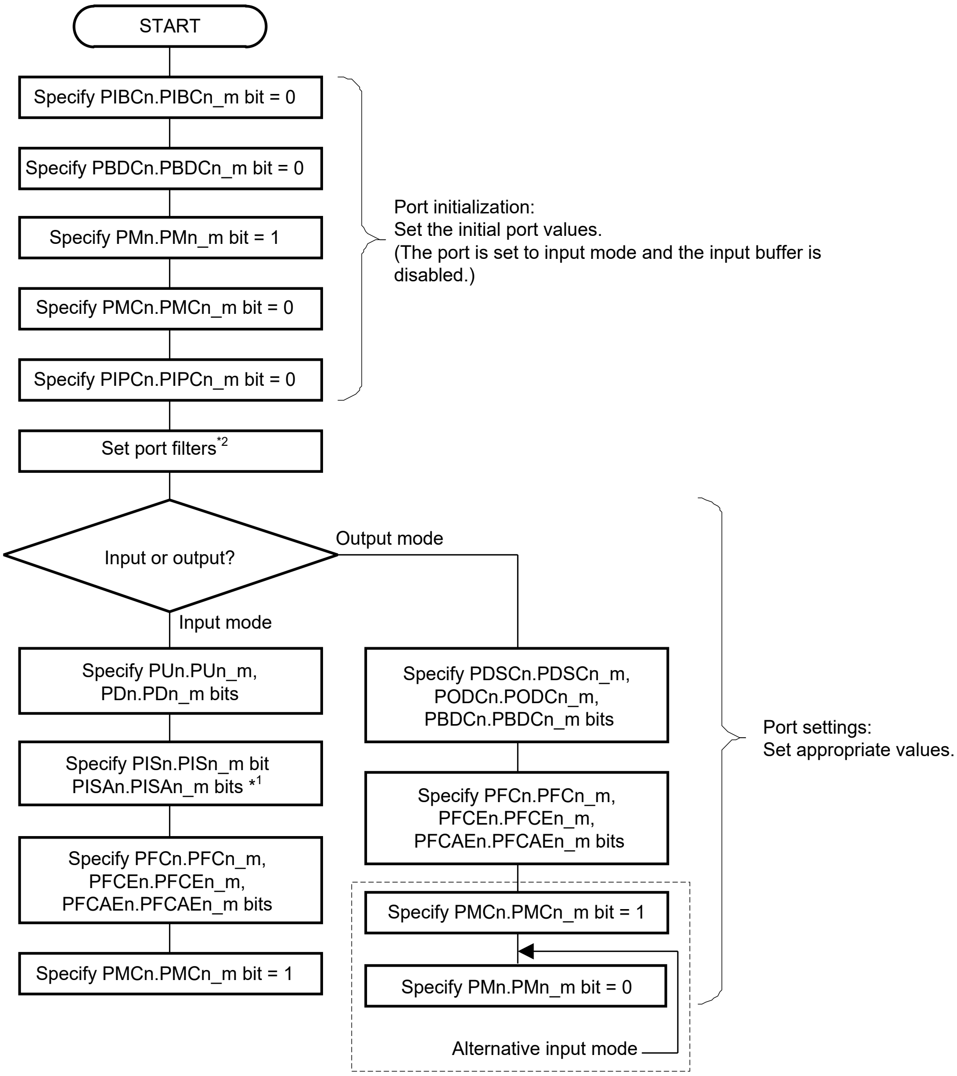
(1) With IP Control



Note 1. There is no PISAn.PISAn_m bit in RH850/F1KM-S1.

Figure 2C.8 Example of Port Settings (in Alternative Mode)

(2) Without IP Control



Note 1. There is no PISAn.PISAn_m bit in RH850/F1KM-S1.

Note 2. While PMC = 0, an interrupt may be triggered during the configuration of the port registers under the following conditions:

For NMI, INTP7 and INTP8 interrupt requests:

- The port filter is set to low level detection.
- The port filter is set to rising edge or both edge detection and the PMC register is set to 1 while the input terminal is at high level.

For INTP0-5 and INTP10-12 interrupt requests:

- The port filter is set to high level detection.
- The port filter is set to falling edge or both edges detection and the PMC register is set to 1 while the input terminal is at low level.

In order to avoid the unintended interrupt occurrence, use the following configuration sequence:

1. Configure the PMC register.
2. Wait for the period of pulse rejection.
3. Configure the edge/level detection register.

Figure 2C.9 Example of Port Settings (in Alternative Mode)

2C.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functions and all the functions assigned to the ports. See the following pages for details.

In addition, whether the port mode is alternative mode or not can be selected by PMCn register setting. When PMCn.PMCn_m = 1, alternative functions are selected by the PFCn, PFCEn, and PFCAEn registers.

Table 2C.38 Port Function

						Device			
Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	48 Pins	64 Pins	80 Pins	100 Pins
JTAG Port 0	JP0_0 - 5	6 bits	In/Out	AWO	JTAG, LPD	✓	✓	✓	✓
Port 0	P0_0 - 3	4 bits	In/Out	AWO		✓	—	—	—
	P0_0 - 6	7 bits				—	✓	—	—
	P0_0 - 12	13 bits				—	—	✓	—
	P0_0 - 14	15 bits				—	—	—	✓
Port 8	P8_0 - 1	2 bits	In/Out	AWO	ADCA0 (10-bit resolution)	✓	—	—	—
	P8_0 - 6	7 bits				—	✓	✓	—
	P8_0 - 12	13 bits				—	—	—	✓
	P8_6	1 bit	In/Out	AWO	$\overline{\text{RESETOUT}}$	—	✓	✓	✓
Port 9	P9_0 - 1	2 bits	In/Out	ISO	ADCA0 (10-bit resolution)	✓	—	—	—
	P9_0 - 3	4 bits				—	✓	—	—
	P9_0 - 6	7 bit				—	—	✓	✓
Port 10	P10_0 - 10	11 bits	In/Out	ISO		✓	—	—	—
	P10_0 - 14	15 bits				—	✓	—	—
	P10_0 - 15	16 bits				—	—	✓	✓
Port 11	P11_0 - 4	5 bits	In/Out	ISO		—	—	✓	—
	P11_0 - 7	8 bits				—	—	—	✓
Analog Port 0	AP0_0 - 7	8 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	✓	—	—	—
	AP0_0 – 9	10 bits				—	✓	—	—
	AP0_0 – 10	11 bits				—	—	✓	—
	AP0_0 - 15	16 bits				—	—	—	✓

2C.10.1 JTAG Port 0 (JP0)

2C.10.1.1 Alternative Function

The following alternative functions are available when JTAG port 0 is configured as a general-purpose I/O port by setting OPJTAG[1:0] on the corresponding option byte to 00_B.

Table 2C.39 JTAG Port 0 (JP0)

Port Mode (JPMC 0_m = 0)	Alternative Mode (JPMC0_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
JP0_0[*1]	INTP0		TAUJ2I0	TAUJ2O0	FPDR	FPDT										DCUTDI/ LPDI/ LPDIO	15	19	23	28
JP0_1	INTP1		TAUJ0I0	TAUJ0O0		FPDT										DCUTDO/ LPDO	14	18	22	27
JP0_2	INTP2		TAUJ0I1	TAUJ0O1	FPCK											DCUTCK/ LPDCLK	13	17	21	26
JP0_3	INTP3	CSCXFOUT	TAUJ0I2	TAUJ0O2												DCUTMS	12	16	20	25
JP0_4																$\overline{\text{DCUTRST}}$	11	15	19	24
JP0_5	NMI	RTCA0OUT	TAUJ0I3	TAUJ0O3												$\overline{\text{DCURDY}}$ / LPDCLKOUT	10	14	18	23

Note 1. In LPD (1 pin) mode, the JP0_0 output buffer state is Open-drain.

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2C.10.1.2 Control Registers

Table 2C.40 Control Registers (JP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
JP0	JTAG port register 0	8	5-0	R/W	0000_H	00_H	✓	✓	✓	✓
JPSR0	JTAG port set/reset register 0	32	21-16, 5-0	R/W	0010_H	$0000\ 0000_H$	✓	✓	✓	✓
JPPR0	JTAG port pin read register 0	8	5-0	R	0020_H	00_H	✓	✓	✓	✓
JPM0	JTAG port mode register 0	8	5-0	R/W	0030_H	FF_H	✓	✓	✓	✓
JPMC0	JTAG port mode control register 0	8	5, 3-0	R/W	0040_H	00_H	✓	✓	✓	✓
JPFC0	JTAG port function control register 0	8	5, 3-0	R/W	0050_H	00_H	✓	✓	✓	✓
JPFCE0	JTAG port function control expansion register 0	8	2-0	R/W	0060_H	00_H	✓	✓	✓	✓
JPNOT0	JTAG port NOT register 0	8	5-0	W	0070_H	00_H	✓	✓	✓	✓
JPMSR0	JTAG port mode set/reset register 0	32	21-16, 5-0	R/W	0080_H	$0000\ FFFF_H$	✓	✓	✓	✓
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	R/W	0090_H	$0000\ 0000_H$	✓	✓	✓	✓
JPIBC0	JTAG port input buffer control register 0	8	5-0	R/W	0400_H	00_H	✓	✓	✓	✓
JPBDC0	JTAG port bidirection control register 0	8	5-0	R/W	0410_H	00_H	✓	✓	✓	✓
JPU0	Pull-up option register 0	8	5-0	R/W	0430_H	00_H	✓	✓	✓	✓
JPD0	Pull-down option register 0	8	5-0	R/W	0440_H	00_H	✓	✓	✓	✓
JPODC0	JTAG port open drain control register 0	32	5-0	R/W	0450_H	$0000\ 0000_H$	✓	✓	✓	✓
JPDSC0	JTAG port drive strength control register 0	32	5, 3-1	R/W	0460_H	$0000\ 0000_H$	✓	✓	✓	✓
JPIS0	JTAG port input buffer selection register 0	8	5, 3-0	R/W	0470_H	FF_H	✓	✓	✓	✓
JPISA0	JTAG port input buffer selection advanced register 0	8	3, 2, 0	R/W	$04A0_H$	00_H	✓	✓	✓	✓
JPPROTS0	JTAG port protection status register 0	32	0	R	$04B0_H$	$0000\ 0000_H$	✓	✓	✓	✓
JPPCMD0	JTAG port protection command register 0	32	7-0	W	$04C0_H$	$xxxx\ xx00_H$	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.10.2 Port 0 (P0)

2C.10.2.1 Alternative Function

Table 2C.41 Port 0 (P0)

Port Mode (PMC0_ m = 0)	Alternative Mode (PMC0_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative							
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output			48 Pins	64 Pins	80 Pins	100 Pins
P0_0	TAUD0I2	TAUD0O2	RLIN20RX	CAN0TX		PWGA10O	$\overline{\text{CSIH0SSI}}$	DPO	TAUJ2I1	TAUJ2O1							4	4	6	6
P0_1	TAUD0I4	TAUD0O4	CAN0RX/ INTP0	RLIN20TX	INTP0	PWGA11O	CSIH0SI	APO	TAUJ2I2	TAUJ2O2			CAN0RX				5	5	7	7
P0_2	TAUD0I6	TAUD0O6		RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO	TAUJ2I3	TAUJ2O3					6	—	—	—
	TAUD0I6	TAUD0O6	CAN1RX/ INTP1	RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO	TAUJ2I3	TAUJ2O3	CAN1RX				—	6	8	8
P0_3	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10		DPIN1			CSIH0SO	INTP10		TAUJ1I0	TAUJ1O0	RLIN30RX				7	—	—	—
	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10	CAN1TX	DPIN1	PWGA13O		CSIH0SO	INTP10		TAUJ1I0	TAUJ1O0	RLIN30RX				—	7	9	9
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O		SELDP0	DPIN8						RLIN31RX				—	9	—	—
	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O	CSIH1SI	SELDP0	DPIN8		TAUB0I12	TAUB0O12			RLIN31RX				—	—	11	11
P0_5	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1									CAN2RX				—	10	—	—
	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO			TAUB0I14	TAUB0O14			CAN2RX				—	—	12	12
P0_6	INTP2		DPIN10	SELDP2													—	11	—	—
	INTP2		DPIN10	SELDP2	CSIH1SC												—	—	13	—
	INTP2		DPIN10	SELDP2	CSIH1SC					PWGA35O							—	—	—	13
P0_7	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0									—	—	34	—
	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	CAN3RX/ INTP3				CAN3RX				—	—	—	40
P0_8		RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2									—	—	33	—
		RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX							—	—	—	39

Table 2C.41 Port 0 (P0)

Port Mode (PMC 0_m = 0)	Alternative Mode (PMC0_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P0_9	INTP12	CSIH1CSS0	DPIN7				TAUB0I4	TAUB0O4									—	—	32	—
	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	CAN4RX/ INTP4				CAN4RX				—	—	—	38
P0_10	INTP3	CSIH1CSS1	DPIN11				TAUB0I6	TAUB0O6									—	—	31	—
	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6		CAN4TX							—	—	—	37
P0_11	RIIC0SDA			CSIH1CSS2	TAUB0I8	TAUB0O8											—	—	14	—
	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8		PWGA34O									—	—	—	14
P0_12	RIIC0SCL				TAUB0I10	TAUB0O10	CSIG0SI										—	—	15	—
	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI										—	—	—	15
P0_13	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	CAN5RX/ INTP5		RLIN32RX		CAN5RX				—	—	—	16
P0_14		RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX							—	—	—	17

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2C.10.2.2 Control Registers

Table 2C.42 Control Registers (P0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W^{*1}			48 Pins	64 Pins	80 Pins	100 Pins
P0	Port register 0	16	3-0	R/W	0000$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PSR0	Port set/reset register 0	32	19-16, 3-0	R/W	0100$_{H}$	0000 0000$_{H}$	✓	—	—	—
			22-16, 6-0				—	✓	—	—
			28-16, 12-0				—	—	✓	—
			30-16, 14-0				—	—	—	✓
PPR0	Port pin read register 0	16	3-0	R	0200$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PM0	Port mode register 0	16	3-0	R/W	0300$_{H}$	FFFF$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PMC0	Port mode control register 0	16	3-0	R/W	0400$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PFC0	Port function control register 0	16	3-0	R/W	0500$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PFCE0	Port function control expansion register 0	16	3-0	R/W	0600$_{H}$	0000$_{H}$	✓	—	—	—
			5-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PNOT0	Port NOT register 0	16	3-0	W	0700$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PMSR0	Port mode set/reset register 0	32	19-16, 3-0	R/W	0800$_{H}$	0000 FFFF$_{H}$	✓	—	—	—
			22-16, 6-0				—	✓	—	—
			28-16, 12-0				—	—	✓	—
			30-16, 14-0				—	—	—	✓
PMCSR0	Port mode control set/reset register 0	32	19-16, 3-0	R/W	0900$_{H}$	0000 0000$_{H}$	✓	—	—	—
			22-16, 6-0				—	✓	—	—
			28-16, 12-0				—	—	✓	—
			30-16, 14-0				—	—	—	✓
PFCAE0	Port function control additional expansion register 0	16	3-0	R/W	0A00$_{H}$	0000$_{H}$	✓	—	—	—
			5-0				—	✓	✓	—
			14, 13, 10-0				—	—	—	✓
PIBC0	Port input buffer control register 0	16	3-0	R/W	4000$_{H}$	0000$_{H}$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓

Table 2C.42 Control Registers (P0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
PBDC0	Port bidirection control register 0	16	3-0	R/W	4100_H	0000_H	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PIPC0	Port IP control register 0	16	3, 2	R/W	4200_H	0000_H	✓	—	—	—
			6, 5, 3, 2				—	✓	✓	—
			14, 13, 6, 5, 3, 2				—	—	—	✓
PU0	Pull-up option register 0	16	3-0	R/W	4300_H	0000_H	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PD0	Pull-down option register 0	16	3-0	R/W	4400_H	0000_H	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PODC0	Port open drain control register 0	32	3-0	R/W	4500_H	$0000\ 0000_H$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PDSC0	Port drive strength control register 0	32	3-0	R/W	4600_H	$0000\ 0000_H$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PIS0	Port input buffer selection register 0	16	3-0	R/W	4700_H	$FFFF_H$	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	—	✓
PPROTS0	Port protection status register 0	32	0	R	$4B00_H$	$0000\ 0000_H$	✓	✓	✓	✓
PPCMD0	Port protection command register 0	32	7-0	W	$4C00_H$	$XXXX\ XX00_H$	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.10.3 Port 8 (P8)

2C.10.3.1 Alternative Function

Table 2C.43 Port 8 (P8)

Port Mode (PMC8_m = 0)	Alternative Mode (PMC8_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P8_0	TAUJ0I0	TAUJ0O0	DPIN2		INTP4	CSIH0CSS0			RIIC1SDA		SENT0RX				ADCA0I0S		23	—	—	—
	TAUJ0I0	TAUJ0O0	DPIN2	PWGA14O	INTP4	CSIH0CSS0			RIIC1SDA		SENT0RX				ADCA0I0S		—	27	35	42
P8_1	TAPA0ESO	TAUJ0O1	DPIN0		INTP5				RIIC1SCL			SENT0SPCO			ADCA0I1S		24	—	—	—
	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5				RIIC1SCL			SENT0SPCO			ADCA0I1S		—	28	—	—
	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5	CSIH1CSS3			RIIC1SCL			SENT0SPCO			ADCA0I1S		—	—	36	43
P8_2	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0		PWGA22O									ADCA0I4S		—	13	—	—
	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O									ADCA0I4S		—	—	17	19
P8_3	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1		PWGA23O									ADCA0I5S		—	29	—	—
	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1	INTP7	PWGA23O									ADCA0I5S		—	—	37	44
P8_4	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2											ADCA0I6S		—	30	—	—
	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8										ADCA0I6S		—	—	38	—
	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8	PWGA36O									ADCA0I6S		—	—	—	45
P8_5	TAUJ0I3	TAUJ0O3	NMI	CSIH0CSS3											ADCA0I7S		—	31	39	—
	TAUJ0I3	TAUJ0O3	NMI	CSIH0CSS3		PWGA37O									ADCA0I7S		—	—	—	46
P8_6	NMI					RTCA0OUT									ADCA0I8S	$\overline{\text{RESETOUT}}$	—	32	—	—
	NMI	CSIH0CSS4				RTCA0OUT									ADCA0I8S	$\overline{\text{RESETOUT}}$	—	—	40	—
	NMI	CSIH0CSS4		PWGA38O		RTCA0OUT									ADCA0I8S	$\overline{\text{RESETOUT}}$	—	—	—	47
P8_7		CSIH3CSS0		PWGA39O		ADCA0SEL0		RTCA0OUT							ADCA0I14S		—	—	—	48
P8_8		CSIH3CSS1		PWGA40O		ADCA0SEL1									ADCA0I15S		—	—	—	49
P8_9		CSIH3CSS2		PWGA41O		ADCA0SEL2									ADCA0I16S		—	—	—	50
P8_10		CSIH3CSS3	DPIN14	PWGA42O											ADCA0I17S		—	—	—	20
P8_11	TAUJ1I2	TAUJ1O2	DPIN15	PWGA43O		CSIH1CSS4									ADCA0I18S		—	—	—	21
P8_12	TAUJ1I3	TAUJ1O3	DPIN16	PWGA44O		CSIH1CSS5									ADCA0I19S		—	—	—	22

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. Use ADC functions with their initial settings. For details, see **Table 2C.44, Control Registers (P8)**.
3. When the $\overline{\text{RESETOUT}}$ function is selected for the P8_6 pin, the P8_6 pin outputs a low-level as the $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is released. For details, see **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

2C.10.3.2 Control Registers

Table 2C.44 Control Registers (P8)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
P8	Port register 8	16	1,0	R/W	0020H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PSR8	Port set/reset register 8	32	17,16,1,0	R/W	0120H	0000 0000H	✓	—	—	—
			22-16, 6-0				—	✓	✓	—
			28-16,12-0				—	—	—	✓
PPR8	Port pin read register 8	16	1,0	R	0220H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PM8	Port mode register 8	16	1,0	R/W	0320H	FFBFH	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PMC8	Port mode control register 8	16	1,0	R/W	0420H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PFC8	Port function control register 8	16	1,0	R/W	0520H	0000H	✓	—	—	—
			5-0				—	✓	✓	—
			12-0				—	—	—	✓
PFCE8	Port function control expansion register 8	16	1,0	R/W	0620H	0000H	✓	—	—	—
			6, 3-0				—	✓	—	—
			6, 4-0				—	—	✓	—
			12, 11, 9-0				—	—	—	✓
PNOT8	Port NOT register 8	16	1,0	W	0720H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PMSR8	Port mode set/reset register 8	32	17,16,1,0	R/W	0820H	0000 FFBFH	✓	—	—	—
			22-16, 6-0				—	✓	✓	—
			28-16,12-0				—	—	—	✓
PMCSR8	Port mode control set/reset register 8	32	17,16,1,0	R/W	0920H	0000 0000H	✓	—	—	—
			22-16, 6-0				—	✓	✓	—
			28-16,12-0				—	—	—	✓
PFCAE8	Port function control additional expansion register 8	16	1,0	R/W	0A20H	0000H	✓	✓	✓	✓
PIBC8	Port input buffer control register 8	16	1,0	R/W	4020H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PBDC8	Port bidirection control register 8	16	1,0	R/W	4120H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PU8	Pull-up option register 8	16	1,0	R/W	4320H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PD8	Pull-down option register 8	16	1,0	R/W	4420H	0000H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PODC8	Port open drain control register 8	32	1,0	R/W	4520H	0000 0040H	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓

Table 2C.44 Control Registers (P8)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
PIS8	Port input buffer selection register 8	16	1,0	R/W	4720_H	$FFFF_H$	✓	—	—	—
			6-0				—	✓	✓	—
			12-0				—	—	—	✓
PPROTS8	Port protection status register 8	32	0	R	$4B20_H$	$0000\ 0000_H$	✓	✓	✓	✓
PPCMD8	Port protection command register 8	32	7-0	W	$4C20_H$	$xxxx\ xx00_H$	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

CAUTION

P8_6 drives a low level after any kind of reset release, until it is later configured differently by register settings. For details, see **Section 2C.11.1.1, P8_6: $\overline{\textbf{RESETOUT}}$** .

2C.10.4 Port 9 (P9)

2C.10.4.1 Alternative Function

Table 2C.45 Port 9 (P9)

Port Mode (PMC9_m = 0)	Alternative Mode (PMC9_m = 1)																PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative							
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	ADC	Special function	48 Pins	64 Pins	80 Pins	100 Pins
P9_0	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0		KR0I4		TAUJ1I1	TAUJ1O1	SENT1RX		RIIC1SDA		ADCA0I2S		35	45	—	—
	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0	CSIH2CSS0	KR0I4		TAUJ1I1	TAUJ1O1	SENT1RX		RIIC1SDA		ADCA0I2S		—	—	54	69
P9_1	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5				TAUJ1I2	TAUJ1O2		SENT1SPCO	RIIC1SCL		ADCA0I3S		36	46	—	—
	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5	CSIH2CSS1			TAUJ1I2	TAUJ1O2		SENT1SPCO	RIIC1SCL		ADCA0I3S		—	—	55	70
P9_2	KR0I6	PWGA20O	TAPA0ESO												ADCA0I9S		—	47	—	—
	KR0I6	PWGA20O	TAPA0ESO	CSIH2CSS2											ADCA0I9S		—	—	56	71
P9_3	KR0I7	PWGA21O			TAUJ1I1	TAUJ1O1									ADCA0I10S		—	48	—	—
	KR0I7	PWGA21O		CSIH2CSS3	TAUJ1I1	TAUJ1O1									ADCA0I10S		—	—	57	72
P9_4		CSIH0CSS5			TAUJ1I0	TAUJ1O0									ADCA0I11S		—	—	58	—
		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0									ADCA0I11S		—	—	—	73
P9_5		CSIH0CSS6			TAUJ1I1	TAUJ1O1									ADCA0I12S		—	—	59	—
		CSIH0CSS6		PWGA34O	TAUJ1I1	TAUJ1O1									ADCA0I12S		—	—	—	74
P9_6		CSIH0CSS7													ADCA0I13S		—	—	60	—
		CSIH0CSS7		PWGA35O											ADCA0I13S		—	—	—	75

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. Use ADC functions with their initial settings. For details, see **Table 2C.46, Control Registers (P9)**.

2C.10.4.2 Control Registers

Table 2C.46 Control Registers (P9)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
P9	Port register 9	16	1, 0	R/W	0024_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PSR9	Port set/reset register 9	32	17, 16, 1, 0	R/W	0124_H	0000 0000_H	✓	—	—	—
			19-16, 3-0				—	✓	—	—
			22-16, 6-0				—	—	✓	✓
PPR9	Port pin read register 9	16	1, 0	R	0224_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PM9	Port mode register 9	16	1, 0	R/W	0324_H	$FFFF_H$	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PMC9	Port mode control register 9	16	1, 0	R/W	0424_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PFC9	Port function control register 9	16	1, 0	R/W	0524_H	0000_H	✓	—	—	—
			2-0				—	✓	—	—
			3-0				—	—	✓	—
			6-0				—	—	—	✓
PFCE9	Port function control expansion register 9	16	1, 0	R/W	0624_H	0000_H	✓	—	—	—
			3, 1, 0				—	✓	—	—
			5-3, 1, 0				—	—	✓	✓
PNOT9	Port NOT register 9	16	1, 0	W	0724_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PMSR9	Port mode set/reset register 9	32	17, 16, 1, 0	R/W	0824_H	0000 $FFFF_H$	✓	—	—	—
			19-16, 3-0				—	✓	—	—
			22-16, 6-0				—	—	✓	✓
PMCSR9	Port mode control set/reset register 9	32	17, 16, 1, 0	R/W	0924_H	0000 0000_H	✓	—	—	—
			19-16, 3-0				—	✓	—	—
			22-16, 6-0				—	—	✓	✓
PFCAE9	Port function control additional expansion register 9	16	1, 0	R/W	$0A24_H$	0000_H	✓	✓	✓	✓
PIBC9	Port input buffer control register 9	16	1, 0	R/W	4024_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PBDC9	Port bidirection control register 9	16	1, 0	R/W	4124_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PU9	Pull-up option register 9	16	1, 0	R/W	4324_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PD9	Pull-down option register 9	16	1, 0	R/W	4424_H	0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PODC9	Port open drain control register 9	32	1, 0	R/W	4524_H	0000 0000_H	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓

Table 2C.46 Control Registers (P9)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
PIS9	Port input buffer selection register 9	16	1, 0	R/W	4724$_H$	FFFF$_H$	✓	—	—	—
			3-0				—	✓	—	—
			6-0				—	—	✓	✓
PPROTS9	Port protection status register 9	32	0	R	4B24$_H$	0000 0000$_H$	✓	✓	✓	✓
PPCMD9	Port protection command register 9	32	7-0	W	4C24$_H$	xxxx xx00$_H$	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.10.5 Port 10 (P10)

2C.10.5.1 Alternative Function

Table 2C.47 Port 10 (P10)

Port Mode (PMC10_ m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P10_0	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O	TAUJ1I3	TAPA0UP				TAUJ1O3	CAN0RX				46	62	—	—
	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O	TAUJ1I3	TAPA0UP	CSIH1SI			TAUJ1O3	CAN0RX				—	—	78	98
P10_1	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN						TAUJ3O0		MODE0	47	63	—	—
	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O	TAUJ3I0	TAPA0UN	CSIH1SC					TAUJ3O0		MODE0	—	—	79	99
P10_2	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP								MODE1	48	64	—	—
	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO						MODE1	—	—	80	100
P10_3	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN									1	1	—	—
	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$								—	—	1	1
P10_4	TAUD0I9	TAUD0O9	RLIN21RX		KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$								2	2	2	2
P10_5	TAUD0I11	TAUD0O11		RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO							3	3	3	3
P10_6	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2										MODE2	39	—	—	—
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1						CAN1RX			MODE2	—	51	63	80
P10_7	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O							TAUJ3I1	TAUJ3O1			40	—	—	—
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX					TAUJ3I1	TAUJ3O1			—	52	64	81
P10_8	TAUD0I10	TAUD0O10	CSIG0SI		ENCA0EC	PWGA5O							TAUJ3I2	TAUJ3O2		FLMD1	41	53	65	82
P10_9	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO					RLIN30RX				42	54	66	83
P10_10	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1					TAUJ3I3	TAUJ3O3			43	55	67	84
P10_11		PWGA16O	RLIN31RX/ INTP11										RLIN31RX				—	56	—	—
		PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0	TAUB0I1	TAUB0O1					RLIN31RX				—	—	68	85

Table 2C.47 Port 10 (P10)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P10_12		PWGA17O		RLIN31TX													—	57	—	—
		PWGA17O		RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3									—	—	69	86
P10_13	$\overline{\text{CSIH0SSI}}$	PWGA18O															—	58	—	—
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12				TAUB0I5	TAUB0O5					RLIN32RX				—	—	70	87
P10_14		PWGA19O															—	59	—	—
		PWGA19O		RLIN32TX			TAUB0I7	TAUB0O7									—	—	71	—
		PWGA19O		RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7									—	—	—	88
P10_15							TAUB0I9	TAUB0O9									—	—	4	—
	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9									—	—	—	4

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2C.10.5.2 Control Registers

Table 2C.48 Control Registers (P10)

		Register Size	Effective Bit		Offset Address	Value after Reset	Device			
Register	Function		Position	R/W*1			48 Pins	64 Pins	80 Pins	176 Pins
P10	Port register 10	16	10-0	R/W	0028$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PSR10	Port set/reset register 10	32	26-16, 10-0	R/W	0128$_H$	0000 0000$_H$	✓	—	—	—
			30-16, 14-0				—	✓	—	—
			31-16, 15-0				—	—	✓	✓
PPR10	Port pin read register 10	16	10-0	R	0228$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PM10	Port mode register 10	16	10-0	R/W	0328$_H$	FFFF$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PMC10	Port mode control register 10	16	10-0	R/W	0428$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PFC10	Port function control register 10	16	10-0	R/W	0528$_H$	0000$_H$	✓	—	—	—
			12-0				—	✓	—	—
			15-0				—	—	✓	✓
PFCE10	Port function control expansion register 10	16	10-0	R/W	0628$_H$	0000$_H$	✓	—	—	—
			11-0				—	✓	—	—
			15-0				—	—	✓	✓
PNOT10	Port NOT register 10	16	10-0	W	0728$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PMSR10	Port mode set/reset register 10	32	26-16, 10-0	R/W	0828$_H$	0000 FFFF$_H$	✓	—	—	—
			30-16, 14-0				—	✓	—	—
			31-16, 15-0				—	—	✓	✓
PMCSR10	Port mode control set/reset register 10	32	26-16, 10-0	R/W	0928$_H$	0000 0000$_H$	✓	—	—	—
			30-16, 14-0				—	✓	—	—
			31-16, 15-0				—	—	✓	✓
PFCAE10	Port function control additional expansion register 10	16	10-7, 5, 4, 1, 0	R/W	0A28$_H$	0000$_H$	✓	—	—	—
			11-4, 1, 0				—	✓	—	—
			13, 11-0				—	—	✓	✓
PIBC10	Port input buffer control register 10	16	10-0	R/W	4028$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PBDC10	Port bidirection control register 10	16	10-0	R/W	4128$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PIPC10	Port IP control register 10	16	7-0	R/W	4228$_H$	0000$_H$	✓	✓	✓	✓
PU10	Pull-up option register 10	16	10-0	R/W	4328$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PD10	Pull-down option register 10	16	10-0	R/W	4428$_H$	0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PODC10	Port open drain control register 10	32	10-0	R/W	4528$_H$	0000 0000$_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓

Table 2C.48 Control Registers (P10)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W[*1]			48 Pins	64 Pins	80 Pins	176 Pins
PDSC10	Port drive strength control register 10	32	10-0	R/W	4628_H	$0000\ 0000_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PIS10	Port input buffer selection register 10	16	10-0	R/W	4728_H	$FFFF_H$	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PPROTS10	Port protection status register 10	32	0	R	$4B28_H$	$0000\ 0000_H$	✓	✓	✓	✓
PPCMD10	Port protection command register 10	32	7-0	W	$4C28_H$	$xxxx\ xx00_H$	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.10.6 Port 11 (P11)

2C.10.6.1 Alternative Function

Table 2C.49 Port 11 (P11)

Port Mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
P11_0	CSIH2RYI	CSIH2RYO					TAUB0I11	TAUB0O11									—	—	5	—
	CSIH2RYI	CSIH2RYO		PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11									—	—	—	5
P11_1	$\overline{\text{CSIH2SSI}}$		RLIN20RX	CSIH0CSS7			TAUB0I13	TAUB0O13									—	—	72	—
	$\overline{\text{CSIH2SSI}}$		RLIN20RX	CSIH0CSS7		PWGA26O	TAUB0I13	TAUB0O13									—	—	—	89
P11_2		CSIH2SO	RLIN32RX/ INTP12	RLIN20TX			TAUB0I15	TAUB0O15					RLIN32RX				—	—	73	—
		CSIH2SO	RLIN32RX/ INTP12	RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15					RLIN32RX				—	—	—	90
P11_3	CSIH2SC									RLIN32TX							—	—	74	—
	CSIH2SC		CAN3RX/ INTP3	PWGA28O						RLIN32TX			CAN3RX				—	—	—	91
P11_4	CSIH2SI																—	—	75	—
	CSIH2SI	CAN3TX		PWGA29O													—	—	—	92
P11_5	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI								CAN5RX				—	—	—	93
P11_6	RLIN33RX/ INTP13	CAN5TX		PWGA31O		CSIH3SO							RLIN33RX				—	—	—	94
P11_7	INTP5			PWGA32O	CSIH3SC												—	—	—	95

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2C.10.6.2 Control Registers

Table 2C.50 Control Registers (P11)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W^{*1}			48 Pins	64 Pins	80 Pins	100 Pins
P11	Port register 11	16	4-0	R/W	$002C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PSR11	Port set/reset register 11	32	20-16, 4-0	R/W	$012C_H$	$0000\ 0000_H$	—	—	✓	—
			23-16, 7-0				—	—	—	✓
PPR11	Port pin read register 11	16	4-0	R	$022C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PM11	Port mode register 11	16	4-0	R/W	$032C_H$	$FFFF_H$	—	—	✓	—
			7-0				—	—	—	✓
PMC11	Port mode control register 11	16	4-0	R/W	$042C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PFC11	Port function control register 11	16	2-0	R/W	$052C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PFCE11	Port function control expansion register 11	16	3-0	R/W	$062C_H$	0000_H	—	—	✓	—
			7-5, 3-0				—	—	—	✓
PNOT11	Port NOT register 11	16	4-0	W	$072C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PMSR11	Port mode set/reset register 11	32	20-16, 4-0	R/W	$082C_H$	$0000\ FFFF_H$	—	—	✓	—
			23-16, 7-0				—	—	—	✓
PMCSR11	Port mode control set/reset register 11	32	20-16, 4-0	R/W	$092C_H$	$0000\ 0000_H$	—	—	✓	—
			23-16, 7-0				—	—	—	✓
PFCAE11	Port function control additional expansion register 11	16	3, 2	R/W	$0A2C_H$	0000_H	—	—	✓	—
			6, 5, 3, 2				—	—	—	✓
PIBC11	Port input buffer control register 11	16	4-0	R/W	$402C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PBDC11	Port bidirection control register 11	16	4-0	R/W	$412C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PIPC11	Port IP control register 11	16	3, 2	R/W	$422C_H$	0000_H	—	—	✓	—
			7, 6, 3, 2				—	—	—	✓
PU11	Pull-up option register 11	16	4-0	R/W	$432C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PD11	Pull-down option register 11	16	4-0	R/W	$442C_H$	0000_H	—	—	✓	—
			7-0				—	—	—	✓
PODC11	Port open drain control register 11	32	4-0	R/W	$452C_H$	$0000\ 0000_H$	—	—	✓	—
			7-0				—	—	—	✓
PDSC11	Port drive strength control register 11	32	4-0	R/W	$462C_H$	$0000\ 0000_H$	—	—	✓	—
			7-0				—	—	—	✓
PIS11	Port input buffer selection register 11	16	4-0	R/W	$472C_H$	$FFFF_H$	—	—	✓	—
			7-0				—	—	—	✓
PPROTS11	Port protection status register 11	32	0	R	$4B2C_H$	$0000\ 0000_H$	—	—	✓	✓
PPCMD11	Port protection command register 11	32	7-0	W	$4C2C_H$	$xxxx\ xx00_H$	—	—	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.10.7 Analog Port 0 (AP0)

2C.10.7.1 Alternative Function

Table 2C.51 Analog Port 0 (AP0)

Port Mode	Alternative Mode														ADC	Special function	PKG No.			
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative				48 Pins	64 Pins	80 Pins	100 Pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output						
AP0_0															ADCA0I0		34	44	53	68
AP0_1															ADCA0I1		33	43	52	67
AP0_2															ADCA0I2		32	42	51	66
AP0_3															ADCA0I3		31	41	50	65
AP0_4															ADCA0I4		30	40	49	64
AP0_5															ADCA0I5		29	39	48	63
AP0_6															ADCA0I6		28	38	47	62
AP0_7															ADCA0I7		27	37	46	61
AP0_8															ADCA0I8		—	36	45	60
AP0_9															ADCA0I9		—	35	44	59
AP0_10															ADCA0I10		—	—	43	58
AP0_11															ADCA0I11		—	—	—	57
AP0_12															ADCA0I12		—	—	—	56
AP0_13															ADCA0I13		—	—	—	55
AP0_14															ADCA0I14		—	—	—	54
AP0_15															ADCA0I15		—	—	—	53

CAUTION

Use ADC functions with their initial settings. For details, see **Table 2C.52, Control Registers (AP0)**.

2C.10.7.2 Control Registers

Table 2C.52 Control Registers (AP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device			
			Position	R/W*1			48 Pins	64 Pins	80 Pins	100 Pins
AP0	Analog port register 0	16	7-0	R/W	$00C8_H$	0000_H	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓
APSR0	Analog port set/reset register 0	32	23-16, 7-0	R/W	$01C8_H$	$0000\ 0000_H$	✓	—	—	—
			25-16, 9-0				—	✓	—	—
			26-16, 10-0				—	—	✓	—
			31-16, 15-0				—	—	—	✓
APPR0	Analog port pin read register 0	16	7-0	R	$02C8_H$	0000_H	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓
APM0	Analog port mode register 0	16	7-0	R/W	$03C8_H$	$FFFF_H$	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓
APNOT0	Analog port NOT register 0	16	7-0	W	$07C8_H$	0000_H	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓
APMSR0	Analog port mode set/reset register 0	32	23-16, 7-0	R/W	$08C8_H$	$0000\ FFFF_H$	✓	—	—	—
			25-16, 9-0				—	✓	—	—
			26-16, 10-0				—	—	✓	—
			31-16, 15-0				—	—	—	✓
APIBC0	Analog port input buffer control register 0	16	7-0	R/W	$40C8_H$	0000_H	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓
APBDC0	Analog port bidirection control register 0	16	7-0	R/W	$41C8_H$	0000_H	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2C.11 Port (Special I/O) Function Overview

This section describes the port (special I/O) functions.

2C.11.1 Special I/O after Reset

The special port function after reset is deasserted is shown below.

2C.11.1.1 P8_6: $\overline{\text{RESETOUT}}$

The P8_6 pin ($\overline{\text{RESETOUT}}$ signal) changes PM8.PM8_6 and PODC8.PODC8_6 registers value after reset by OPBT0.RESETOUTEN setting.

The P8_6 pin outputs a low level while a reset is asserted, and pin status of after the reset is different.

(Case 1): OPBT0.RESETOUTEN = 1

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 0: Output mode
- PODC8.PODC8_6 = 1: Open-drain

(Case 2): OPBT0.RESETOUTEN = 0

- P8.P8_6 = 0: Outputs low level
- PM8.PM8_6 = 1: Input mode
- PODC8.PODC8_6 = 0: Push-pull

For detail of OPBT0.RESETOUTEN register, see **Section 44.9.2, OPBT0 — Option Byte 0**, also see **Section 9BC.1.3, Reset Output ($\overline{\text{RESETOUT}}$)**.

When the P8_6 pin setting is updated with another value, the pin operates by new setting.

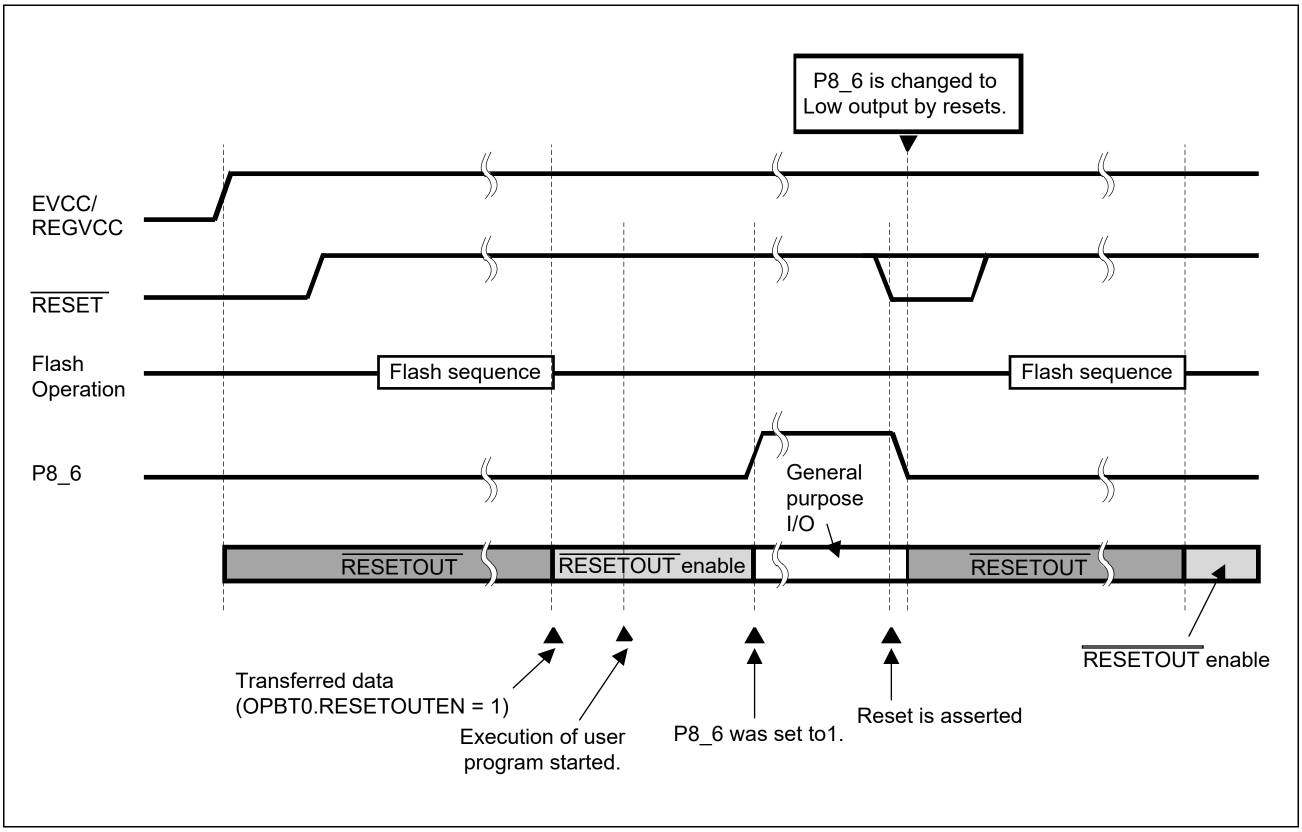
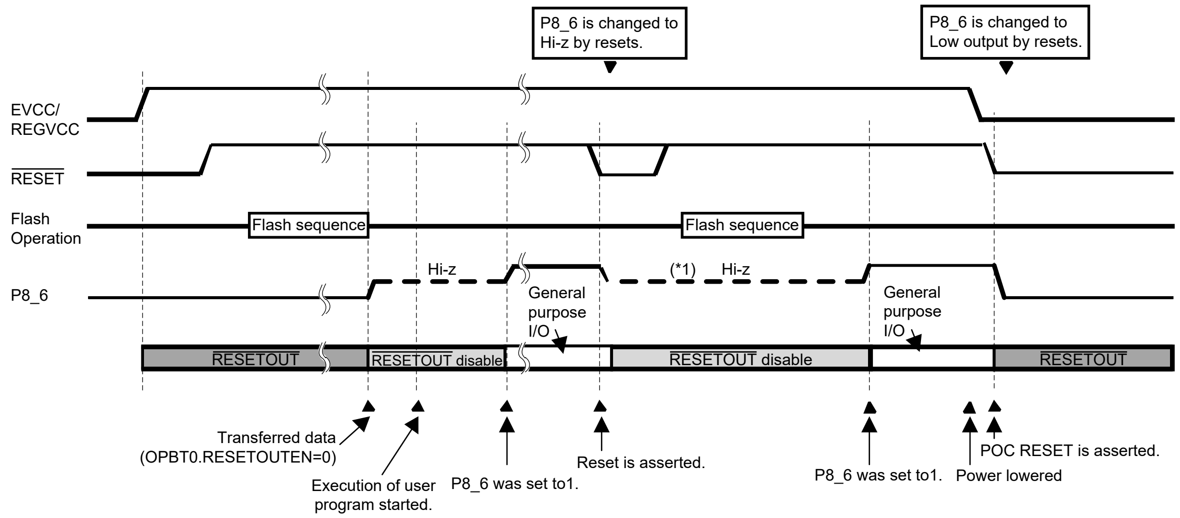


Figure 2C.10 P8_6 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation While a Reset is asserted and released: (Case 1) OPBT0.RESETOUTEN setting is 1



Note 1. When a reset except POC reset occurs with RESETOUT disable (OPBT0.RESETOUTEN = 0), P8_6 pin (RESETOUT signal) will be changed to Hi-z.

Figure 2C.11 P8_6 Pin (RESETOUT Signal) Operation While a Reset is asserted and released: (Case 2) OPBT0.RESETOUTEN setting is 0

2C.11.1.2 JP0_0 to JP0_5: Debug Interface

If the OPJTAG[1:0] setting is the combination below, the pins of the JTAG port group can be used as a debug interface after reset release.

Table 2C.53 Debug Interface

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	$\overline{\text{DCUTRST}}$ input	$\overline{\text{DCURDY}}$ output
0	1	LPD (4 pins)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output
1	0	LPD (1 pin)	LPDIO input/output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

NOTE

For the OPJTAG[1:0] settings, see **Section 44.9.2, OPBT0 — Option Byte 0**.

2C.11.1.3 FPDR(JP0_0), FPDT(JP0_1), FPCK(JP0_2): Flash Programmer

These pins are used for connecting a flash programmer. See *Flash Programmer's Manual* for details.

2C.11.1.4 Mode Pins

The FLMD0 pin in combination with the P10_8: FLMD1 pin can select serial programming mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_2: MODE1 and the P10_1: MODE0 pins can select boundary scan mode.

The FLMD0 pin in combination with the P10_8: FLMD1, the P10_6: MODE2, the P10_2: MODE1 and the P10_1: MODE0 pins can select user boot mode.

For details on the mode selection, see **Section 6, Operating Mode**.

2C.11.2 A/D Input Alternative I/O

The following ports are permanently connected to A/D input functions. (However, an analog input to the A/D is controlled by the A/D module.)

Table 2C.54 A/D Input Alternative Pins

		Device			
Port	A/D Input	48 Pins	64 Pins	80 Pins	100 Pins
P8_0	ADCA0I0S	✓	✓	✓	✓
P8_1	ADCA0I1S	✓	✓	✓	✓
P8_2	ADCA0I4S	—	✓	✓	✓
P8_3	ADCA0I5S	—	✓	✓	✓
P8_4	ADCA0I6S	—	✓	✓	✓
P8_5	ADCA0I7S	—	✓	✓	✓
P8_6	ADCA0I8S	—	✓	✓	✓
P8_7	ADCA0I14S	—	—	—	✓
P8_8	ADCA0I15S	—	—	—	✓
P8_9	ADCA0I16S	—	—	—	✓
P8_10	ADCA0I17S	—	—	—	✓
P8_11	ADCA0I18S	—	—	—	✓
P8_12	ADCA0I19S	—	—	—	✓
P9_0	ADCA0I2S	✓	✓	✓	✓
P9_1	ADCA0I3S	✓	✓	✓	✓
P9_2	ADCA0I9S	—	✓	✓	✓
P9_3	ADCA0I10S	—	✓	✓	✓
P9_4	ADCA0I11S	—	—	✓	✓
P9_5	ADCA0I12S	—	—	✓	✓
P9_6	ADCA0I13S	—	—	✓	✓
AP0_0	ADCA0I0	✓	✓	✓	✓
AP0_1	ADCA0I1	✓	✓	✓	✓
AP0_2	ADCA0I2	✓	✓	✓	✓
AP0_3	ADCA0I3	✓	✓	✓	✓
AP0_4	ADCA0I4	✓	✓	✓	✓
AP0_5	ADCA0I5	✓	✓	✓	✓
AP0_6	ADCA0I6	✓	✓	✓	✓
AP0_7	ADCA0I7	✓	✓	✓	✓
AP0_8	ADCA0I8	—	✓	✓	✓
AP0_9	ADCA0I9	—	✓	✓	✓
AP0_10	ADCA0I10	—	—	✓	✓
AP0_11	ADCA0I11	—	—	—	✓
AP0_12	ADCA0I12	—	—	—	✓
AP0_13	ADCA0I13	—	—	—	✓
AP0_14	ADCA0I14	—	—	—	✓
AP0_15	ADCA0I15	—	—	—	✓

2C.11.3 Special I/O Control

2C.11.3.1 Direct I/O Control (PIPC)

Some alternative functions take over the input and output control of the ports.

The following table lists all alternative functions where PIPCn.PIPCn_m must be set to 1. For details, see **Section 2C.9.2.3, PIPCn — Port IP Control Register**.

Table 2C.55 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1

Function	Alternative Functions Name	Port Name	Power Supply Area	Control	Reference Section
TAPA	TAPA0UP	P10_0	ISO	U phase Hi-Z control	**Section 36**
	TAPA0UN	P10_1	ISO		
	TAPA0VP	P10_2	ISO	V phase Hi-Z control	
	TAPA0VN	P10_3	ISO		
	TAPA0WP	P10_4	ISO	W phase Hi-Z control	
	TAPA0WN	P10_5	ISO		
CSIG	CSIG0SO	P0_13	AWO	Serial data output control signal	**Section 19**
		P10_6	ISO		
	CSIG0SC	P0_14	AWO	Master (1) / slave (0) mode signal	
		P10_7	ISO		
CSIH	CSIH0SO	P0_3	AWO	Serial data output control signal	**Section 20**
	CSIH0SC	P0_2	AWO	Master (1) / slave (0) mode signal	
	CSIH1SO	P0_5	AWO	Serial data output control signal	
		P10_2	ISO		
	CSIH1SC	P0_6	AWO	Master (1) / slave (0) mode signal	
		P10_1	ISO		
	CSIH2SO	P11_2	ISO	Serial data output control signal	
	CSIH2SC	P11_3	ISO	Master (1) / slave (0) mode signal	
	CSIH3SO	P11_6	ISO	Serial data output control signal	
	CSIH3SC	P11_7	ISO	Master (1) / slave (0) mode signal	

2C.11.3.2 Input Buffer Control (PISn/JPIS0, JPISA0)

The port input buffer characteristics (Type 1 or Type 2) of this device can be selected using the PISn/ JPIS0 register. The applicable pins are shown in the following table.

The JTAG port input buffer characteristics (Type 1/2 or Type 5) of this device can be selected using the JPISA0 register. The applicable pins are shown in **Table 2C.57, JTAG Port Input Buffer Characteristics Selection**.

Table 2C.56 Port Input Buffer Characteristics Selection

	Input Buffer Selection		Device			
Port Name	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	48 Pins	64 Pins	80 Pins	100 Pins
P0_0	SHMT1	SHMT4	✓	✓	✓	✓
P0_1	SHMT1	SHMT4	✓	✓	✓	✓
P0_2	SHMT1	SHMT4	✓	✓	✓	✓
P0_3	SHMT1	SHMT4	✓	✓	✓	✓
P0_4	SHMT1	SHMT4	—	✓	✓	✓
P0_5	SHMT1	SHMT4	—	✓	✓	✓
P0_6	SHMT1	SHMT4	—	✓	✓	✓
P0_7	SHMT1	SHMT4	—	—	✓	✓
P0_8	SHMT1	SHMT4	—	—	✓	✓
P0_9	SHMT1	SHMT4	—	—	✓	✓
P0_10	SHMT1	SHMT4	—	—	✓	✓
P0_11	SHMT1	SHMT4	—	—	✓	✓
P0_12	SHMT1	SHMT4	—	—	✓	✓
P0_13	SHMT1	SHMT4	—	—	—	✓
P0_14	SHMT1	SHMT4	—	—	—	✓
P8_0	SHMT1	SHMT4	✓	✓	✓	✓
P8_1	SHMT1	SHMT4	✓	✓	✓	✓
P8_2	SHMT1	SHMT4	—	✓	✓	✓
P8_3	SHMT1	SHMT4	—	✓	✓	✓
P8_4	SHMT1	SHMT4	—	✓	✓	✓
P8_5	SHMT1	SHMT4	—	✓	✓	✓
P8_6	SHMT1	SHMT4	—	✓	✓	✓
P8_7	SHMT1	SHMT4	—	—	—	✓
P8_8	SHMT1	SHMT4	—	—	—	✓
P8_9	SHMT1	SHMT4	—	—	—	✓
P8_10	SHMT1	SHMT4	—	—	—	✓
P8_11	SHMT1	SHMT4	—	—	—	✓
P8_12	SHMT1	SHMT4	—	—	—	✓
P9_0	SHMT1	SHMT4	✓	✓	✓	✓
P9_1	SHMT1	SHMT4	✓	✓	✓	✓
P9_2	SHMT1	SHMT4	—	✓	✓	✓
P9_3	SHMT1	SHMT4	—	✓	✓	✓
P9_4	SHMT1	SHMT4	—	—	✓	✓
P9_5	SHMT1	SHMT4	—	—	✓	✓
P9_6	SHMT1	SHMT4	—	—	✓	✓
P10_0	SHMT1	SHMT4	✓	✓	✓	✓
P10_1	SHMT1	SHMT4	✓	✓	✓	✓

Table 2C.56 Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection		Device			
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	48 Pins	64 Pins	80 Pins	100 Pins
P10_2	SHMT1	SHMT4	✓	✓	✓	✓
P10_3	SHMT1	SHMT4	✓	✓	✓	✓
P10_4	SHMT1	SHMT4	✓	✓	✓	✓
P10_5	SHMT1	SHMT4	✓	✓	✓	✓
P10_6	SHMT1	SHMT4	✓	✓	✓	✓
P10_7	SHMT1	SHMT4	✓	✓	✓	✓
P10_8	SHMT1	SHMT4	✓	✓	✓	✓
P10_9	SHMT1	SHMT4	✓	✓	✓	✓
P10_10	SHMT1	SHMT4	✓	✓	✓	✓
P10_11	SHMT1	SHMT4	—	✓	✓	✓
P10_12	SHMT1	SHMT4	—	✓	✓	✓
P10_13	SHMT1	SHMT4	—	✓	✓	✓
P10_14	SHMT1	SHMT4	—	✓	✓	✓
P10_15	SHMT1	SHMT4	—	—	✓	✓
P11_0	SHMT1	SHMT4	—	—	✓	✓
P11_1	SHMT1	SHMT4	—	—	✓	✓
P11_2	SHMT1	SHMT4	—	—	✓	✓
P11_3	SHMT1	SHMT4	—	—	✓	✓
P11_4	SHMT1	SHMT4	—	—	✓	✓
P11_5	SHMT1	SHMT4	—	—	—	✓
P11_6	SHMT1	SHMT4	—	—	—	✓
P11_7	SHMT1	SHMT4	—	—	—	✓

Table 2C.57 JTAG Port Input Buffer Characteristics Selection

Port Name	Input Buffer Selection			Devices			
	Type 1 (JPIS0_m = 0 & JPISA0_m = 0)	Type 2 (JPIS0_m = 1 & JPISA0_m = 0)	Type 5 (JPISA0_m = 1)	48 Pins	64 Pins	80 Pins	100 Pins
JP0_0	SHMT1	SHMT4	TTL*1,*2,*3,*4	✓	✓	✓	✓
JP0_1	SHMT1	SHMT4	—	✓	✓	✓	✓
JP0_2	SHMT1	SHMT4	TTL*1,*2,*3	✓	✓	✓	✓
JP0_3	SHMT1	SHMT4	TTL*1,*2	✓	✓	✓	✓
JP0_4	—	SHMT4	—*1,*2	✓	✓	✓	✓
JP0_5	SHMT1	SHMT4	—	✓	✓	✓	✓

Note 1. TTL is selected for Boundary scan mode without JPISA0 register setting.

Note 2. TTL is selected for Nexus in normal operating mode without JPISA0 register setting.

Note 3. TTL is selected for LPD (4 pins) in normal operating mode without JPISA0 register setting.

Note 4. TTL is selected for LPD (1 pin) in normal operating mode without JPISA0 register setting.

NOTES

1. For the SHMT1, SHMT4, and TTL pin characteristics, see **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.
2. For the input buffer after reset, Type 2 (SHMT4) is selected.

2C.11.3.3 Output Buffer Control (PDSC)

The port output driver strength (slow mode/fast mode) can be selected using the PDSCn register. The applicable pins are shown in the following table. Only slow mode is supported for ports other than those listed below.

Table 2C.58 Output Buffer Characteristics Selection

	Output Drive Strength Selection		Device			
Port Name	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	48 Pins	64 Pins	80 Pins	100 Pins
JP0_1	10 MHz	40 MHz	✓	✓	✓	✓
JP0_2	10 MHz	40 MHz	✓	✓	✓	✓
JP0_3	10 MHz	40 MHz	✓	✓	✓	✓
JP0_5	10 MHz	40 MHz	✓	✓	✓	✓
P0_0	10 MHz	40 MHz	✓	✓	✓	✓
P0_1	10 MHz	40 MHz	✓	✓	✓	✓
P0_2	10 MHz	40 MHz[*1]	✓	✓	✓	✓
P0_3	10 MHz	40 MHz[*1]	✓	✓	✓	✓
P0_4	10 MHz	40 MHz	—	✓	✓	✓
P0_5	10 MHz	40 MHz[*2]	—	✓	✓	✓
P0_6	10 MHz	40 MHz[*2]	—	✓	✓	✓
P0_7	10 MHz	40 MHz	—	—	✓	✓
P0_8	10 MHz	40 MHz	—	—	✓	✓
P0_9	10 MHz	40 MHz	—	—	✓	✓
P0_10	10 MHz	40 MHz	—	—	✓	✓
P0_11	10 MHz	40 MHz	—	—	✓	✓
P0_12	10 MHz	40 MHz	—	—	✓	✓
P0_13	10 MHz	40 MHz	—	—	—	✓
P0_14	10 MHz	40 MHz	—	—	—	✓
P10_0	10 MHz	40 MHz	✓	✓	✓	✓
P10_1	10 MHz	40 MHz[*2]	✓	✓	✓	✓
P10_2	10 MHz	40 MHz[*2]	✓	✓	✓	✓
P10_3	10 MHz	40 MHz	✓	✓	✓	✓
P10_4	10 MHz	40 MHz	✓	✓	✓	✓
P10_5	10 MHz	40 MHz	✓	✓	✓	✓
P10_6	10 MHz	40 MHz	✓	✓	✓	✓
P10_7	10 MHz	40 MHz	✓	✓	✓	✓
P10_8	10 MHz	40 MHz	✓	✓	✓	✓
P10_9	10 MHz	40 MHz	✓	✓	✓	✓
P10_10	10 MHz	40 MHz	✓	✓	✓	✓
P10_11	10 MHz	40 MHz	—	✓	✓	✓
P10_12	10 MHz	40 MHz	—	✓	✓	✓
P10_13	10 MHz	40 MHz	—	✓	✓	✓
P10_14	10 MHz	40 MHz	—	✓	✓	✓
P10_15	10 MHz	40 MHz	—	—	✓	✓
P11_0	10 MHz	40 MHz	—	—	✓	✓
P11_1	10 MHz	40 MHz	—	—	✓	✓
P11_2	10 MHz	40 MHz[*2]	—	—	✓	✓

Table 2C.58 Output Buffer Characteristics Selection

Port Name	Output Drive Strength Selection		Device			
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	48 Pins	64 Pins	80 Pins	100 Pins
P11_3	10 MHz	40 MHz*[2]	—	—	✓	✓
P11_4	10 MHz	40 MHz	—	—	✓	✓
P11_5	10 MHz	40 MHz	—	—	—	✓
P11_6	10 MHz	40 MHz*[2]	—	—	—	✓
P11_7	10 MHz	40 MHz*[2]	—	—	—	✓

Note 1. Supports Cload: 100 pF (The load capacitance of CSIH0 is 100 pF.)

Note 2. Supports Cload: 50 pF (The load capacitance of CSIH1 to CSIH3 are 50 pF.)

Note 3. In some of the functions, Fast mode or Slow mode is specified. For details, see **Section 47C.5, AC Characteristics**.

2C.12 Noise Filter & Edge/Level Detector

The input signals at some pins are passed through a filter to remove noise and glitches. The RH850/F1KM supports both analog and digital filters.

It also supports the function for edge and level detection after the signals have passed through a filter.

The first part of this section provides an overview of port input pins that are equipped with a filter and the filter type, noise filter & edge/level detection control registers and control bits, and register addresses.

For details on the digital/analog filter function and noise filter & edge/level detection control registers, see **Section 2C.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2C.12.1 Port Filter Assignment

A list of the input pins that incorporate an analog or digital filter is provided below.

2C.12.1.1 Input Pins that Incorporate Analog Filter Type A

The input pins of analog filter type A incorporate an analog filter and edge/level detection function. Edge/level detection is controlled by the following registers.

Filter control register FCLA0CTLm_<*name*> (m = 0 to 7)

A dedicated FCLA0CTLm_<*name*> register is provided for each pin in a port that incorporates an analog filter.

Table 2C.59 Input Pins that Incorporate Analog Filter Type A

		FCLA0CTL Register Configuration		Device			
Module Name	Input Pin	Register	Address	48 Pins	64 Pins	80 Pins	100 Pins
FCLA0	NMI	FCLA0CTL0_NMI	FFC3 4000_H	✓	✓	✓	✓
	INTP0	FCLA0CTL0_INTPL	FFC3 4020_H	✓	✓	✓	✓
	INTP1	FCLA0CTL1_INTPL	FFC3 4024_H	✓	✓	✓	✓
	INTP2	FCLA0CTL2_INTPL	FFC3 4028_H	✓	✓	✓	✓
	INTP3	FCLA0CTL3_INTPL	FFC3 $402C_H$	✓	✓	✓	✓
	INTP4	FCLA0CTL4_INTPL	FFC3 4030_H	✓	✓	✓	✓
	INTP5	FCLA0CTL5_INTPL	FFC3 4034_H	✓	✓	✓	✓
	INTP6	FCLA0CTL6_INTPL	FFC3 4038_H	—	—	✓	✓
	INTP7	FCLA0CTL7_INTPL	FFC3 $403C_H$	—	—	✓	✓
	INTP8	FCLA0CTL0_INTPH	FFC3 4040_H	—	—	✓	✓
	INTP10	FCLA0CTL2_INTPH	FFC3 4048_H	✓	✓	✓	✓
	INTP11	FCLA0CTL3_INTPH	FFC3 $404C_H$	✓	✓	✓	✓
	INTP12	FCLA0CTL4_INTPH	FFC3 4050_H	—	—	✓	✓
	INTP13	FCLA0CTL5_INTPH	FFC3 4054_H	—	—	—	✓

2C.12.1.2 Input Pins that Incorporate Analog Filter Type B

The input pins of analog filter type B incorporate an analog filter. Edge/level detection is controlled by the registers for individual peripheral functions.

Table 2C.60 Input Pins that Incorporate Analog Filter Type B

		Device			
Input Pin	Edge/Level Detection	48 Pins	64 Pins	80 Pins	100 Pins
TAUJ0I0	Edge detection*[1]	✓	✓	✓	✓
TAUJ0I1	Edge detection*[1]	✓	✓	✓	✓
TAUJ0I2	Edge detection*[1]	✓	✓	✓	✓
TAUJ0I3	Edge detection*[1]	✓	✓	✓	✓
TAUJ1I0	Edge detection*[1]	✓	✓	✓	✓
TAUJ1I1	Edge detection*[1]	✓	✓	✓	✓
TAUJ1I2	Edge detection*[1]	✓	✓	✓	✓
TAUJ1I3	Edge detection*[1]	✓	✓	✓	✓
TAUJ2I0	Edge detection*[1]	✓	✓	✓	✓
TAUJ2I1	Edge detection*[1]	✓	✓	✓	✓
TAUJ2I2	Edge detection*[1]	✓	✓	✓	✓
TAUJ2I3	Edge detection*[1]	✓	✓	✓	✓
TAUJ3I0	Edge detection*[1]	✓	✓	✓	✓
TAUJ3I1	Edge detection*[1]	✓	✓	✓	✓
TAUJ3I2	Edge detection*[1]	✓	✓	✓	✓
TAUJ3I3	Edge detection*[1]	✓	✓	✓	✓
TAPA0ESO	Edge detection*[2]	✓	✓	✓	✓
KR0I0	Low level detection	✓	✓	✓	✓
KR0I1	Low level detection	✓	✓	✓	✓
KR0I2	Low level detection	✓	✓	✓	✓
KR0I3	Low level detection	✓	✓	✓	✓
KR0I4	Low level detection	✓	✓	✓	✓
KR0I5	Low level detection	✓	✓	✓	✓
KR0I6	Low level detection	—	✓	✓	✓
KR0I7	Low level detection	—	✓	✓	✓

Note 1. For details on edge detection for TAUJ, see **Section 33.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register**.

Note 2. For details on edge detection for TAPA, see **Section 36.3.2, TAPAnCTL0 — TAPA Control Register 0**.

2C.12.1.3 Input Pins that Incorporate Analog Filter Type C

The input pins of analog filter type C only incorporate an analog filter function.

Table 2C.61 Input Pins that Incorporate Analog Filter Type C

Input Pin
FLMD0
FLMD1
MODE0
MODE1
MODE2
$\overline{\text{RESET}}$
$\overline{\text{DCUTRST}}$

2C.12.1.4 Input Pins that Incorporate Digital Filter Type D

The input pins of digital filter type D incorporate a digital filter and edge detection function. The digital filter and edge detection are controlled by the following registers.

- Filter control register FCLA0CTLm_<*name*> (m = 0 to 2)

  Each port with a digital filter has a special FCLA0CTLm_<name> register.

- Digital noise elimination control register DNFA<*name*>CTL

  Each DNFA<*name*>CTL control register controls digital filter processing for three input signals per group.

- Digital noise elimination enable register DNFA<*name*>EN

  The setting of the DNFA<*name*>ENL[2:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for three input signals per group.

Table 2C.62 Input Pins that Incorporate Digital Filter Type D

	Device				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Filter Control Register	
Input Pin	48 Pins	64 Pins	80 Pins	100 Pins	Control Register	Address	Control Register	Control Bit	Address	Control Register	Address
ADCA0TRG0	✓	✓	✓	✓	DNFAADCTL0CTL	FFC3 00A0$_{H}$	DNFAADCTL0EN (DNFAADCTL0ENL)	DNFAADCTL0 ENL0	FFC3 00A4$_{H}$ (FFC3 00AC$_{H}$)	FCLA0CTL0 _ADC0	FFC3 4060$_{H}$
ADCA0TRG1	✓	✓	✓	✓				DNFAADCTL0 ENL1		FCLA0CTL1 _ADC0	FFC3 4064$_{H}$
ADCA0TRG2	✓	✓	✓	✓				DNFAADCTL0 ENL2		FCLA0CTL2 _ADC0	FFC3 4068$_{H}$

2C.12.1.5 Input Pins that Incorporate Digital Filter Type E

The input pins of digital filter type E incorporate a digital filter. The digital filter is controlled by the following registers. Edge detection is controlled by the registers for individual peripheral functions.

- Digital noise elimination control register DNFA<*name*>CTL

  Each DNFA<*name*>CTL control register controls digital filter processing for up to 16 input signals per group.

- Digital noise elimination enable register DNFA<*name*>EN

  The setting of the DNFA<*name*>ENL[7:0] and DNFA<*name*>ENH[7:0] bits in DNFA<*name*>EN enables or disables digital noise elimination for up to 16 input signals per group.

Table 2C.63 Input Pins that Incorporate Digital Filter Type E

	Devises				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
Input Pin	48 Pins	64 Pins	80 Pins	100 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUD0I0	✓	✓	✓	✓	DNFA TAUD0ICTL	FFC3 0000$_H$	DNFATAUD0IEN (DNFA TAUD0IENH / DNFA TAUD0IENL)	DNFATAUD0IENL0	FFC3 0004$_H$ (FFC3 0008$_H$/ FFC3 000C$_H$)	*1
TAUD0I1	✓	✓	✓	✓				DNFATAUD0IENL1		
TAUD0I2	✓	✓	✓	✓				DNFATAUD0IENL2		
TAUD0I3	✓	✓	✓	✓				DNFATAUD0IENL3		
TAUD0I4	✓	✓	✓	✓				DNFATAUD0IENL4		
TAUD0I5	✓	✓	✓	✓				DNFATAUD0IENL5		
TAUD0I6	✓	✓	✓	✓				DNFATAUD0IENL6		
TAUD0I7	✓	✓	✓	✓				DNFATAUD0IENL7		
TAUD0I8	✓	✓	✓	✓				DNFATAUD0IENH0		
TAUD0I9	✓	✓	✓	✓				DNFATAUD0IENH1		
TAUD0I10	✓	✓	✓	✓				DNFATAUD0IENH2		
TAUD0I11	✓	✓	✓	✓				DNFATAUD0IENH3		
TAUD0I12	✓	✓	✓	✓				DNFATAUD0IENH4		
TAUD0I13	✓	✓	✓	✓				DNFATAUD0IENH5		
TAUD0I14	✓	✓	✓	✓				DNFATAUD0IENH6		
TAUD0I15	✓	✓	✓	✓				DNFATAUD0IENH7		
TAUB0I0	—	—	✓	✓	DNFA TAUB0ICTL	FFC3 0020$_H$	DNFATAUB0IEN (DNFA TAUB0IENH/ DNFA TAUB0IENL)	DNFATAUB0IENL0	FFC3 0024$_H$ (FFC3 0028$_H$/ FFC3 002C$_H$)	*2
TAUB0I1	—	—	✓	✓				DNFATAUB0IENL1		
TAUB0I2	—	—	✓	✓				DNFATAUB0IENL2		
TAUB0I3	—	—	✓	✓				DNFATAUB0IENL3		
TAUB0I4	—	—	✓	✓				DNFATAUB0IENL4		
TAUB0I5	—	—	✓	✓				DNFATAUB0IENL5		
TAUB0I6	—	—	✓	✓				DNFATAUB0IENL6		
TAUB0I7	—	—	✓	✓				DNFATAUB0IENL7		
TAUB0I8	—	—	✓	✓				DNFATAUB0IENH0		
TAUB0I9	—	—	✓	✓				DNFATAUB0IENH1		
TAUB0I10	—	—	✓	✓				DNFATAUB0IENH2		
TAUB0I11	—	—	✓	✓				DNFATAUB0IENH3		
TAUB0I12	—	—	✓	✓				DNFATAUB0IENH4		
TAUB0I13	—	—	✓	✓				DNFATAUB0IENH5		
TAUB0I14	—	—	✓	✓				DNFATAUB0IENH6		
TAUB0I15	—	—	✓	✓				DNFATAUB0IENH7		
ENCA0TIN0	✓	✓	✓	✓	DNFA ENCA0ICTL	FFC3 0060$_H$	DNFAENCA0IEN (DNFA ENCA0IENL)	DNFAENCA0IENL0	FFC3 0064$_H$ (FFC3 006C$_H$)	*3
ENCA0TIN1	✓	✓	✓	✓				DNFAENCA0IENL1		
ENCA0E0	✓	✓	✓	✓				DNFAENCA0IENL2		
ENCA0E1	✓	✓	✓	✓				DNFAENCA0IENL3		
ENCA0EC	✓	✓	✓	✓				DNFAENCA0IENL4		

Table 2C.63 Input Pins that Incorporate Digital Filter Type E

Input Pin	Devises				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	48 Pins	64 Pins	80 Pins	100 Pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
SENT0RX	✓	✓	✓	✓	DNFA SENTICTL	FFC3 00E0$_H$	DNFASENTIEN (DNFA SENTIENL)	DNFASENTIENL0	FFC3 00E4$_H$ (FFC3 00EC$_H$)	—*4
SENT1RX	✓	✓	✓	✓				DNFASENTIENL1		

Note 1. For the setting for TAUD edge detection, see **Section 32.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register**.

Note 2. For the setting for TAUB edge detection, see **Section 31.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register**.

Note 3. For the setting for ENCA edge detection, see **Section 35.3.3, ENCAnIOC0 — ENCAn I/O Control Register 0**.

Note 4. RSENT does not have the edge detection.

2C.12.2 Clock Supply for Port Filters

The following table shows the clock supply for each filter type in each port domain.

Table 2C.64 Clock Supply for Port Filters

Peripheral Function	Port Domain*[1]	Filter Type	Filter Clock	Setting Register	
				Source Clock Selection	Clock Selection
ADCA0	Always-On area (AWO area)	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
TAUD0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
TAUB0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
ENCA0	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
RSENTn	Isolated area (ISO area)	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—

Note 1. Power Domain

NOTE

For the Setting Register, see **Section 12C.4.3, Clock Selector Control Register**.

2C.13 Description of Port Noise Filter & Edge/Level Detection

External signals pass through different types of filters according to the use of each external input signal.

NOTE

In this section, <*name*> in the noise filter control register represents the peripheral function connected to a filter.

2C.13.1 Overview

2C.13.1.1 Analog Filter Types

Analog filters have fixed characteristics.

- Type A: An analog filter with edge detection or level detection.

  Used for external interrupt signals.

- Type B: An analog filter

  Edge detection is performed by each peripheral function. Used for the timer input signals, asynchronous Hi-Z control input signals, and key return input signals.

- Type C: An analog filter only

  Used for the external $\overline{\text{RESET}}$ input and mode signals.

2C.13.1.2 Digital Filter Types

The digital filter characteristics can be adjusted to suit the application.

- Type D: A digital filter with edge detection. Used for the A/D converter external trigger pin.
- Type E: A digital filter. Edge detection is performed by each peripheral function. Used for the timer input signals and encoder input signals.

2C.13.2 Analog Filters

2C.13.2.1 Analog Filter Characteristic

See **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for the input conditions for signals input to pins that incorporate an analog filter.

2C.13.2.2 Analog Filter Control Registers

A dedicated FCLA0CTLm_<*name*> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Table 2C.59, Input Pins that Incorporate Analog Filter Type A**.

2C.13.2.3 Analog Filter in Standby Mode

Analog filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Analog filters in the Always-On area (AWO area) always operate.

The analog filter in standby mode and its wake-up capability depend on the filter types. See the description of the analog filter types below.

(1) Analog Filter Type A

A block diagram of analog filter type A is shown below.

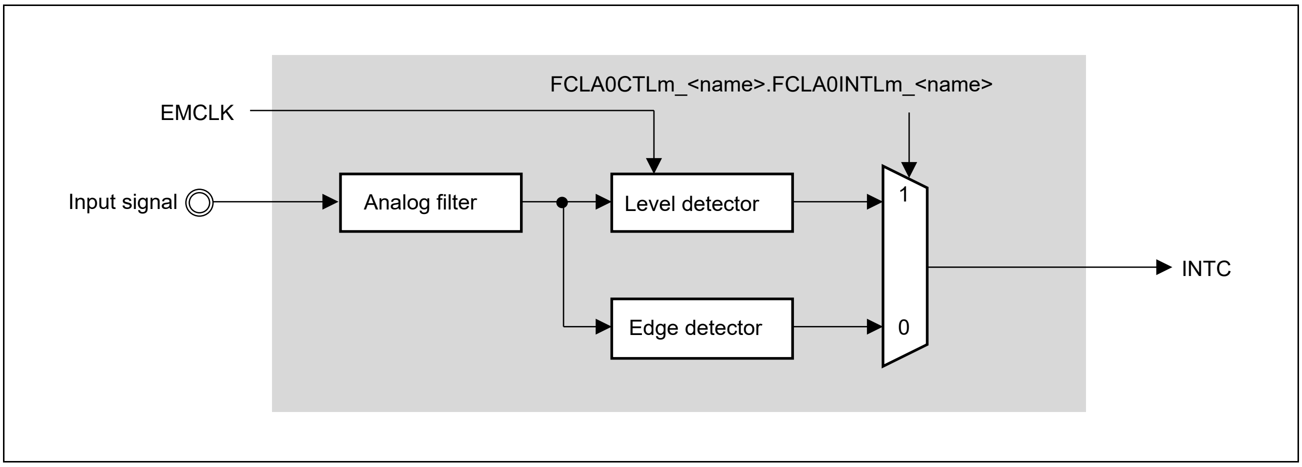


Figure 2C.12 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and glitches, an output signal is generated according to whether an event is detected; that is whether a specified level is detected or whether a change in the level (an edge) occurs.

Whether a level or an edge is detected is selected by the control bit FCLA0CTLm_<*name*>.FCLA0INTLm_<*name*>.

- FCLA0INTLm_<*name*> bit = 0: Edge detection

  Whether a rising or falling edge is detected can be specified by setting the FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> and FCLA0CTLm_<*name*>.FCLA0INTFm_<*name*> bits.

- FCLA0INTLm_<*name*> bit = 1: Level detection

  The detection of a high level or low level can be specified by setting FCLA0CTLm_<*name*>.FCLA0INTRm_<*name*> bit.

The table below summarizes the detection conditions of the analog filter.

Table 2C.65 Analog Filter Event Detection Conditions

FCLA0INTLm_<*name*>	FCLA0INTFm_<*name*>	FCLA0INTRm_<*name*>	Edge Detection	Level Detection
0	0	0	No edge detected	Disabled
	0	1	Rising edge	
	1	0	Falling edge	
	1	1	Rising and falling edges	
1	X	0	Disabled	Low level
	X	1		High level

Analog filter type A in Standby mode

The output signal of an analog filter type A can always be used as a standby mode wake-up signal.

(2) Analog filter type B

A block diagram of analog filter type B is shown below.

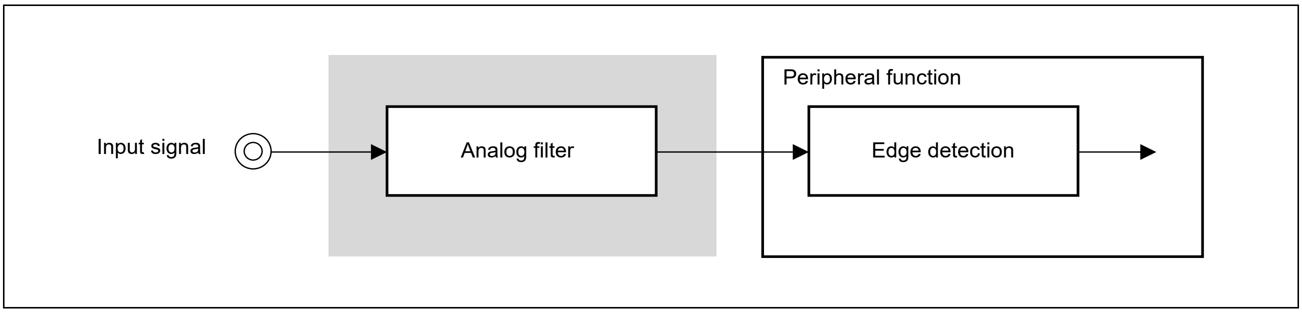


Figure 2C.13 Block Diagram of Analog Filter Type B

Analog filter type B in Standby mode

The output signal of an analog filter type B can always be used as a standby mode wake-up signal.

(3) Analog filter type C

A block diagram of analog filter type C is shown below.

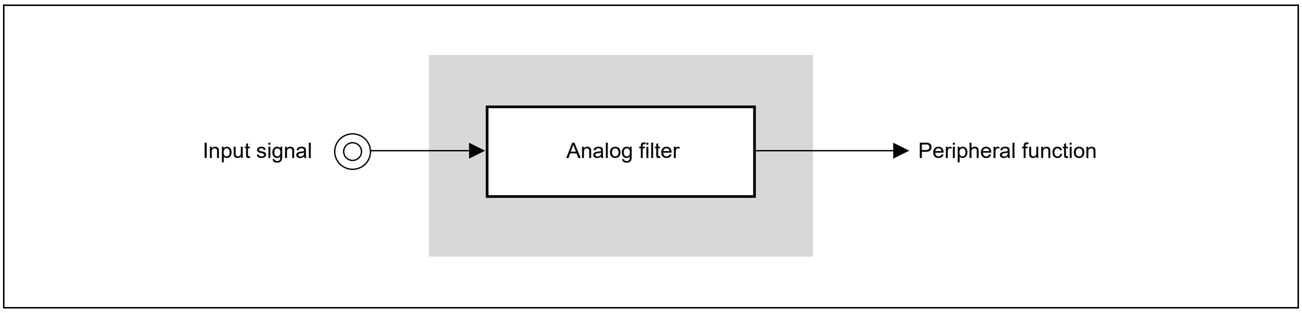


Figure 2C.14 Block Diagram of Analog Filter Type C

The generated signals are always input signals that have passed through an analog filter.

Analog filter type C in Standby mode

Pins equipped with type C analog filters in this product do not support the input of event signals to trigger wake-up from standby.

2C.13.3 Digital Filters

2C.13.3.1 Digital Filter Characteristic

The digital filters allow the filter characteristics to be adjusted accordingly to the needs of the application.

The input signal is sampled with the sampling frequency fs.

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] select the sampling frequency based on

  $f_s = f_{DNFATCKI} / 2^{DNFA<name>PRS[2:0]}$

  where $f_{DNFATCKI}$ is the frequency of the DNFATCKI clock.

- DNFA<*name*>CTL.DNFA<*name*>NFSTS[1:0] determines the number of same level samples, “s”, (2 to 5):

  s = DNFA<*name*>NFSTS[1:0] + 2

External signal pulses shorter than the following are suppressed at all times.

$s \times 1/f_s$

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

$(s + 1) \times 1/f_s$

External signal pulses in the following range may be suppressed or judged as valid.

$s \times 1/f_s$ to $(s + 1) \times 1/f_s$

The filter operation is illustrated in the figure below with DNFA<*name*>NFSTS[1:0] = 01_B, i.e. s = 3 same level samples.

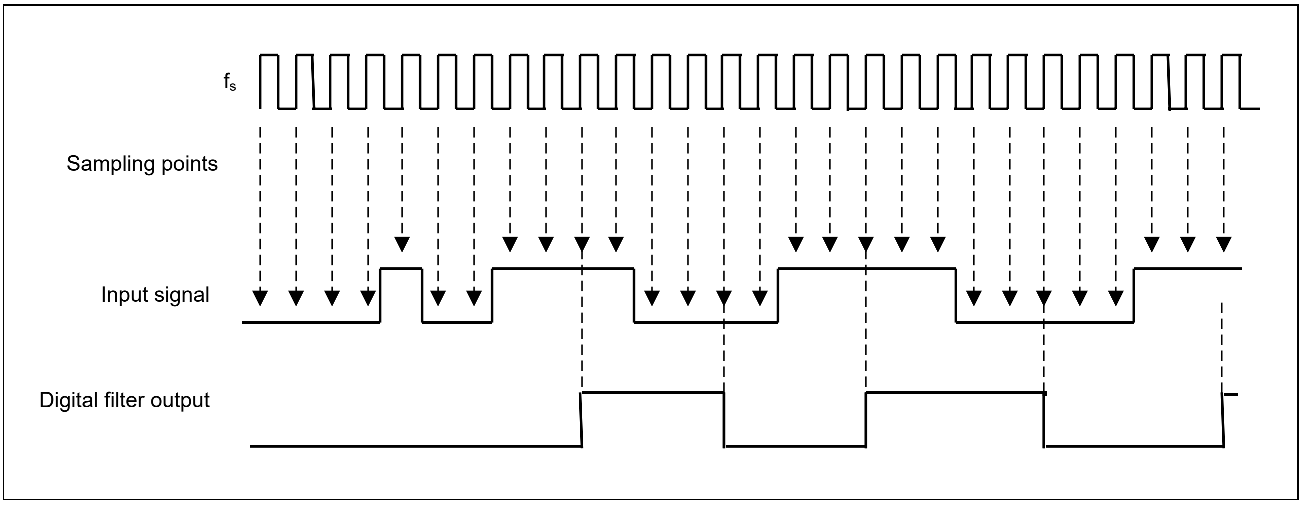


Figure 2C.15 Digital Filter Function

2C.13.3.2 Digital Filter Groups

The input signals processed through digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by DNFA<*name*>CTL.DNFA<*name*>PRS[2:0] and DNFA<*name*>NFSTS[1:0] apply to the signals.

However, the digital filter for each signal can be enabled or disabled separately by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

CAUTIONS

1. When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) for the port pin to switch to the alternative function.
   s = DNFA<*name*>NFSTS[1:0] + 2
   $s \times 1/f_s + 2 \times 1/f_{DNFATCKI}$
2. When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.
   $s \times 1/f_s + 3 \times 1/f_{DNFATCKI}$

2C.13.3.3 Digital Filters in Standby Mode

Digital filters for the function of waking-up from the DeepSTOP mode are located in the Always-On area (AWO area). Digital filters on the Always-On area (AWO area) are always operating.

Digital noise elimination requires the clock supply DNFATCKI to operate.

Pins equipped with digital filters in this product do not support the input of event signals to trigger wake-up from standby.

2C.13.3.4 Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFA<*name*>CTL and digital noise elimination enable register DNFA<*name*>EN are used to set all the filters in the same group (<*name*> = peripheral function group).

The DNFA<*name*>CTL register specifies the characteristics of the digital noise elimination filter for the digital filter of <*name*>.

The DNFA<*name*>EN register enables/disables each filter by setting the corresponding bit in DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The FCLA0CTLm_ADCn registers are ordered in groups of 3 registers with the same index n. The register index n is in 0 or 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2C.62, Input Pins that Incorporate Digital Filter Type D** and **Table 2C.63, Input Pins that Incorporate Digital Filter Type E** in **Section 2C.12.1, Port Filter Assignment**.

CAUTION

Do not change any control register settings while the corresponding digital filter is enabled by DNFA<*name*>EN.DNFA<*name*>ENLm (m = 0 to 7) = 1 and DNFA<*name*>EN.DNFA<*name*>ENHm (m = 0 to 7) = 1. Otherwise an unintended filter output may be generated.

(1) Digital filter type D

A block diagram of digital filter type D is shown below.

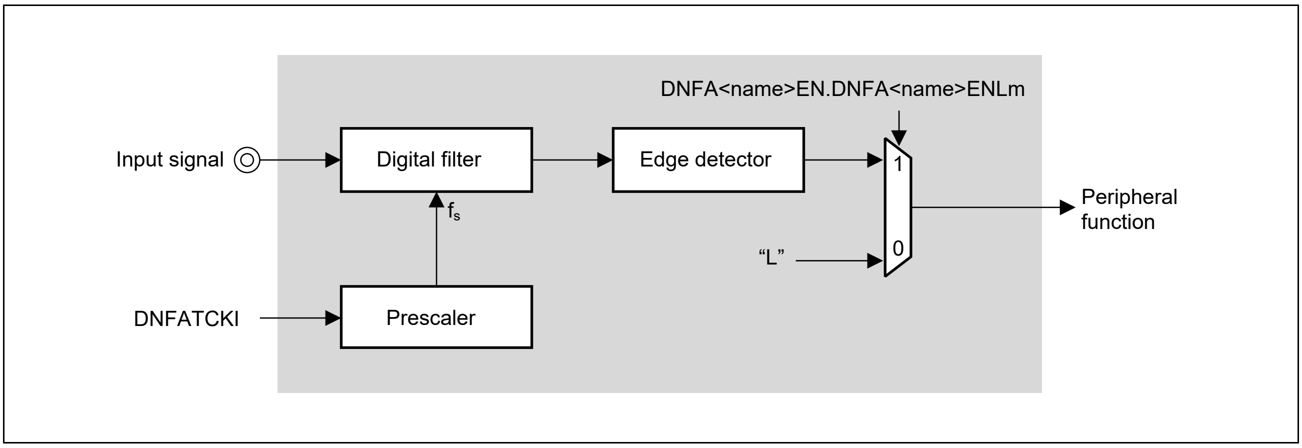


Figure 2C.16 Block Diagram of Digital Filter Type D

The generated signal depends on the register setting, as shown in the following table.

Table 2C.66 Output Options for Digital Filter Type D

DNFA<*name*>EN.DNFA<*name*>ENLm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

(2) Digital filter type E

A block diagram of digital filter type E is shown below.

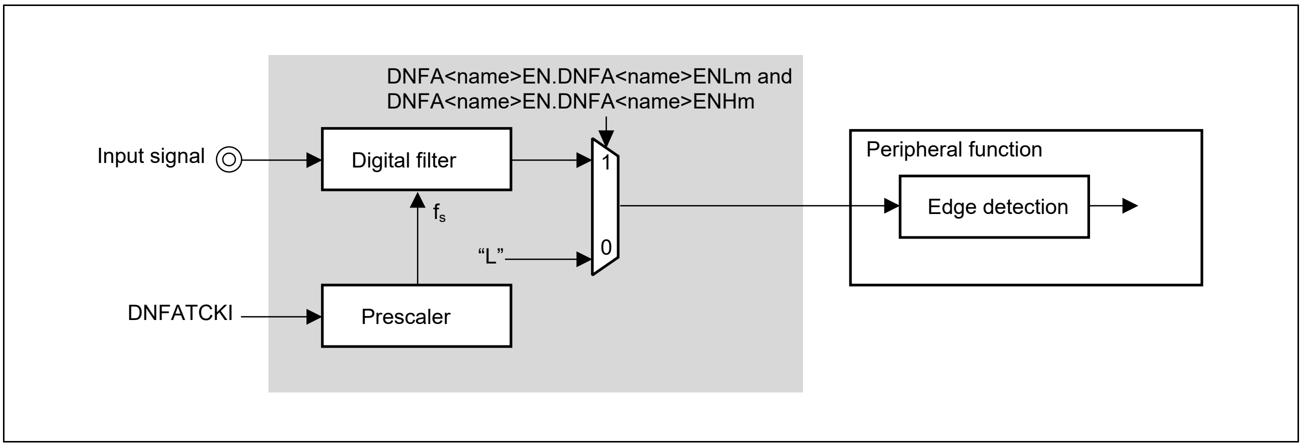


Figure 2C.17 Block Diagram of Digital Filter Type E

The generated signal depends on the register setting, as shown in the following table.

Table 2C.67 Output Options for Digital Filter Type E

DNFA<*name*>EN.DNFA<*name*>ENLm and DNFA<*name*>EN.DNFA<*name*>ENHm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

2C.13.4 Filter Control Registers

The analog and digital filters are controlled and operated by the following registers:

Table 2C.68 List of Filter Registers

Module Name	Register Name	Symbol	Address
FCLA0	Filter control register m	FCLA0CTLm_<name>	The addresses are shown in the tables in **Section 2C.12.1, Port Filter Assignment**.
DNF	Digital noise elimination control register	DNFA<name>CTL	
	Digital noise elimination enable register	DNFA<name>EN	
	Digital noise elimination enable H register	DNFA<name>ENH	
	Digital noise elimination enable L register	DNFA<name>ENL	

2C.13.4.1 FCLA0CTLm_<name> — Filter Control Register

This register controls the analog and digital filter operation.

Access: This register can be read or written in 8-bit units.

Address: The allocation of input signals to FCLA0CTLm_<*name*> registers and the address of each register are shown in the tables in **Section 2C.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLA0INTLm_<name>	FCLA0INTFm_<name>	FCLA0INTRm_<name>
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 2C.69 FCLA0CTLm_<name> Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	FCLA0INTLm_<name>	Detection Mode Selection 0: Edge detection 1: Level detection **NOTE:** This bit is only valid for analog filter type A.
1	FCLA0INTFm_<name>	• In level detection mode (FCLA0INTLm_<name> = 1): This bit has no effect. • In edge detection mode (FCLA0INTLm_<name> = 0): Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.
0	FCLA0INTRm_<name>	• In level detection mode (FCLA0INTLm_<name> = 1): Detected level selection 0: Low level detection 1: High level detection • In edge detection mode (FCLA0INTLm_<name> = 0): Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled **NOTE:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.

CAUTION

Digital filter type D: Always set bit 2 to “0”.

2C.13.4.2 DNFA<name>CTL — Digital Noise Elimination Control Register

This register is used to specify the filter characteristics of the digital noise elimination filter.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit units.

Address: For the correspondence between the DNFA<name>CTL register and input signals, and the addresses of individual registers, see **Table 2C.62, Input Pins that Incorporate Digital Filter Type D** and **Table 2C.63, Input Pins that Incorporate Digital Filter Type E** in **Section 2C.12.1, Port Filter Assignment**.

Value after reset: 00_{H}

Bit	7	6	5	4	3	2	1	0
	—	DNFA<name>NFSTS[1:0]		—	—	DNFA<name>PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2C.70 DNFA<name>CTL Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 5	DNFA<name> NFSTS[1:0]	The DNFA<name>NFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid.

DNFA<name>NFSTS[1:0]	Number of Samples
00_{B}	2
01_{B}	3
10_{B}	4
11_{B}	5

Bit Position	Bit Name	Function
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DNFA<name> PRS[2:0]	Digital filter sampling clock selection

DNFA<name>PRS[2:0]	Sampling Clock Frequency
000_{B}	DNFATCKI/1
001_{B}	DNFATCKI/2
010_{B}	DNFATCKI/4
011_{B}	DNFATCKI/8
100_{B}	DNFATCKI/16
101_{B}	DNFATCKI/32
110_{B}	DNFATCKI/64
111_{B}	DNFATCKI/128

2C.13.4.3 DNFA<name>EN — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 16-bit units.

The upper- and lower-order bytes (DNFA<name>ENH[7:0] and DNFA<name>ENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH. and DNFA<name>ENL.

Address: For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Table 2C.62, Input Pins that Incorporate Digital Filter Type D** and **Table 2C.63, Input Pins that Incorporate Digital Filter Type E** in **Section 2C.12.1, Port Filter Assignment**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFA<name>ENH7	DNFA<name>ENH6	DNFA<name>ENH5	DNFA<name>ENH4	DNFA<name>ENH3	DNFA<name>ENH2	DNFA<name>ENH1	DNFA<name>ENH0	DNFA<name>ENL7	DNFA<name>ENL6	DNFA<name>ENL5	DNFA<name>ENL4	DNFA<name>ENL3	DNFA<name>ENL2	DNFA<name>ENL1	DNFA<name>ENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2C.71 DNFA<name>EN Register Contents

Bit Position	Bit Name	Function
15 to 0	DNFA<name>ENH[7:0] DNFA<name>ENL[7:0]	Digital Noise Elimination Enable/Disable Control 0: Fixed to low level 1: Input signal passed through filter

2C.13.4.4 DNFA<name>ENH — Digital Noise Elimination Enable H Register

Setting in this register correspond to those of the 8 upper-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENH register and input signals, and the addresses of individual registers, see **Table 2C.63, Input Pins that Incorporate Digital Filter Type E** in **Section 2C.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA<name> ENH7	DNFA<name> ENH6	DNFA<name> ENH5	DNFA<name> ENH4	DNFA<name> ENH3	DNFA<name> ENH2	DNFA<name> ENH1	DNFA<name> ENH0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2C.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

2C.13.4.5 DNFA<name>ENL — Digital Noise Elimination Enable L Register

Setting in this register correspond to those of the 8 lower-order bits of the DNFA<name>EN register.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: For the correspondence between the DNFA<name>ENL register and input signals, and the addresses of individual registers, see **Table 2C.62, Input Pins that Incorporate Digital Filter Type D** and **Table 2C.63, Input Pins that Incorporate Digital Filter Type E** in **Section 2C.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DNFA<name> ENL7	DNFA<name> ENL6	DNFA<name> ENL5	DNFA<name> ENL4	DNFA<name> ENL3	DNFA<name> ENL2	DNFA<name> ENL1	DNFA<name> ENL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2C.13.4.3, DNFA<name>EN — Digital Noise Elimination Enable Register**.

Section 3A CPU System of RH850/F1KH-D8

3A.1 Overview

3A.1.1 Block Configuration

Figure 3A.1, Block Configuration Diagram of the RH850/F1KH shows the block configuration diagram of RH850/F1KH.

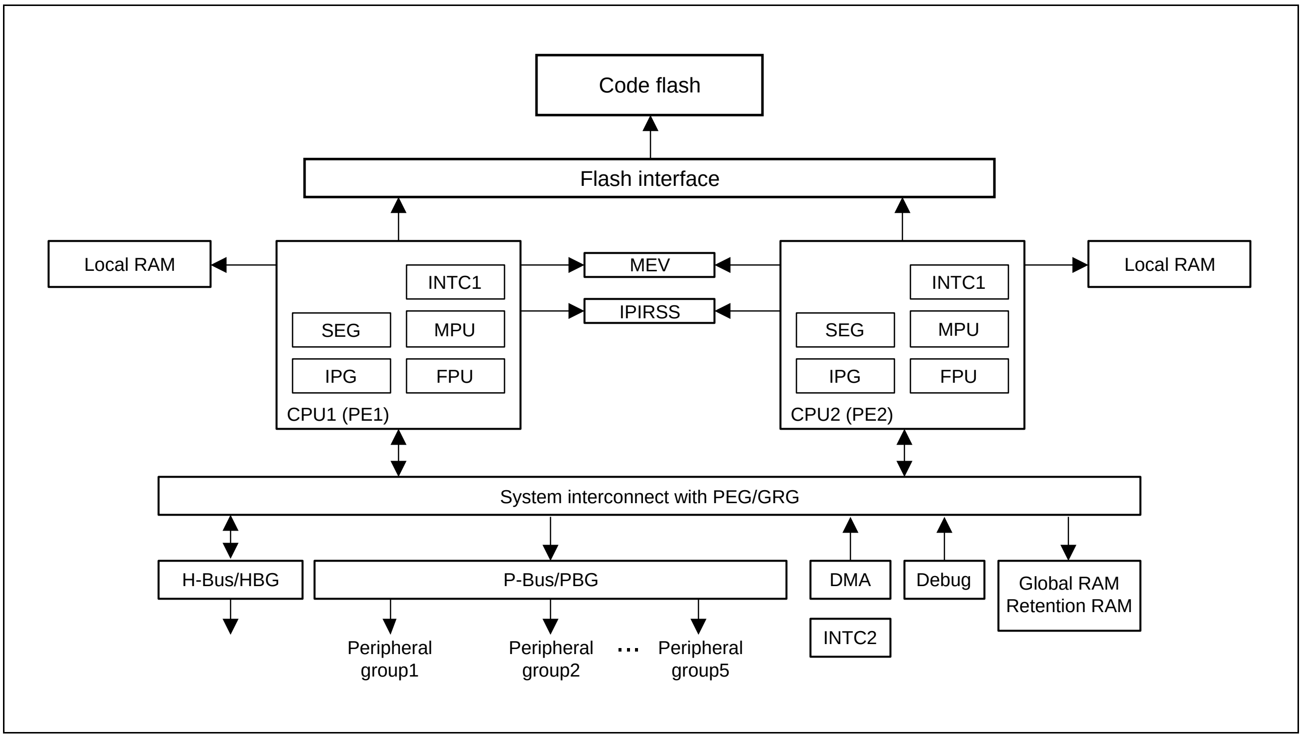


Figure 3A.1 Block Configuration Diagram of the RH850/F1KH

CPU1 (PE1)

The RH850G3KH2.0 Core is used as the main CPU.

CPU2 (PE2)

The RH850G3KH2.0 Core is used as an enhanced performance CPU.

Local RAM

Each CPU has a high-speed accessible RAM

Global RAM

The global RAM is a large-capacity RAM for data sharing among CPUs and with DMA.

Retention RAM

The retention RAM is used to retain values in DeepSTOP mode. Since the continuous global RAM area is assigned for the retention RAM, the retention RAM can also serve as a global RAM for sharing data with the DMA.

Code flash

The code flash memory is included for program storage. It is connected with CPU1 and CPU2 via the flash interface.

Data flash

The data flash memory can be rewritten by the CPUs. It has a greater write endurance than the code flash memory.

P-Bus and H-Bus

The P-Bus connects the peripheral IPs. The P-Bus is divided into five peripheral groups, 1 to 5.

INTC1, INTC2

There are two interrupt controllers, INTC1 and INTC2. INTC1 is an interrupt controller exclusive to each CPU. INTC2 is a common interrupt controller that CPU1 and CPU2 share. The PE to which an interrupt request is bound can be specified by a register setting.

DMA

The DMA transfer module (PDMA) is included.

Slave guard

The slave guard is a function to prevent unauthorized access from the specific bus master, and consists of the following guard structures:

(1) PE guard (PEG)
The PE guard is a function to prevent unauthorized access to the resources (local RAM) in the PE from an external master. After reset is released, access from other than the own PE is prohibited.

(2) Internal Peripheral Guard (IPG)
The PE with system interconnects supports “Internal Peripheral Guard” (IPG) that protects the registers of peripherals against invalid accesses.

(3) Global RAM guard (GRG)
The global RAM guard is a function to prevent unauthorized access to the global RAM and retention RAM from

an external master. The global RAM is in the unprotected state (accessible from all bus master) after reset is released. For details, see **Section 40A, Functional Safety of RH850/F1KH-D8**.

(4) Peripheral guard (PBG / HBG)
The peripheral guard is a function to prevent unauthorized access to peripherals. The control registers in the peripheral circuits are protected against illegal accesses.
For details, see **Section 40A, Functional Safety of RH850/F1KH-D8**.

3A.2 CPU

3A.2.1 Core Functions

3A.2.1.1 Features

Table 3A.1, Features of the RH850G3KH2.0 Core lists features of the RH850G3KH2.0 core.

Table 3A.1 Features of the RH850G3KH2.0 Core

Item	Feature
CPU	● Advanced 32-bit architecture for embedded control ● 32-bit internal data bus ● Thirty-two 32-bit general-purpose registers – RISC-type instruction set – Long-/short-format load/store instructions – Three-operand instructions – Instruction set based on C language ● CPU operating modes – User mode and supervisor mode ● Address space: 4-Gbyte linear address space for both data and instructions
Coprocessor	● Floating-point operation coprocessor (FPU) – Supports single precision (32 bits) – Supports data types and exceptions conforming to IEEE754. – Rounding mode: Neighborhood, 0 direction, +∞ direction, and −∞ direction – Handling of denormalized numbers: Rounding down to 0 or exception notification to conform to IEEE754
Exception/Interrupt	● 16 interrupt priority levels settable for each channel ● Vector selection method selectable according to performance request or memory usage – Direct branching exception vectors – Indirect branching exception vectors referring to the address table ● Supports the high-speed save/return processing of the context by the dedicated instructions (PUSHSP and POPSP) at the generation of an interrupt
Memory management	● Memory protection function (MPU): 16 areas settable
Cache	● No cache memory is equipped.

3A.2.1.2 Register Set

This subsection explains the program registers and system registers incorporated in this CPU.

(1) Program Registers

Program registers include the general-purpose registers (r0 to r31) and program counter (PC).

Table 3A.2 Program Registers

Program Register	Name	Function	Description
General-purpose registers	r0	Zero register	Always retains “0”
	r1	Assembler reserved register	Used as working register for generating addresses
	r2	Register for address and data variables (used when the real-time OS used does not use this register)	
	r3	Stack pointer (SP)	Used for generating a stack frame when a function is called
	r4	Global pointer (GP)	Used for accessing a global variable in the data area
	r5	Text pointer (TP)	Used as a register that indicates the start of the text area (area where program code is placed)
	r6 to r29	Register for address and data variables	
	r30	Element pointer (EP)	Used as a base pointer for generating addresses when accessing memory
	r31	Link pointer (LP)	Used when the compiler calls a function
Program counter	PC	Retains instruction addresses during execution of programs	

NOTE

For further descriptions of r1, r3 to r5, and r31 used by the assembler and/or C compiler, see the specification of each software development environment.

(a) General-Purpose Registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables. Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

1. r0, r3, r30

These registers are implicitly used by instructions.
r0 is a register that always retains “0”. It is used for operations that use 0 and addressing with base address being 0.
r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.
r30 is used as a base pointer when the SLD or SST instruction accesses memory.

2. r1, r4, r5, r31

These registers are implicitly used by the assembler and C compiler.
When using these registers, register contents must first be saved so they are not lost and can be restored after the registers are used.

3. r2

This register might be used by a real-time OS in some cases. If the real-time OS that is being used does not use r2, r2 can be used as a register for address variables or data variables.

(b) PC — Program Counter

The PC retains the address of the instruction being executed. Bit 0 is fixed to 0, and branching to an odd number address is disabled.

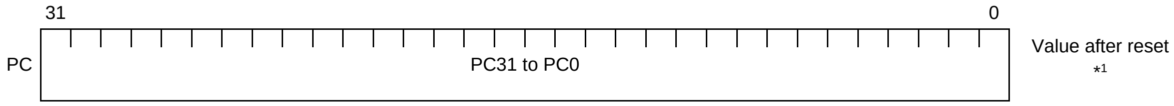


Table 3A.3 PC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	PC31 to PC1	These bits indicate the address of the instruction being executed.	R/W	*1
0	PC0	This bit is fixed to 0. Branching to an odd number address is disabled.	R/W	0

Note 1. The value after reset differs depending on the setting value of the reset vector. For details, see **(q) RBASE — Reset Vector Base Address Register**.

(2) Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

System registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and a selection ID.

Table 3A.4 Basic System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 0	EIPC	Status save registers when acknowledging EI level exception	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	SV
SR2, 0	FEPC	Status save registers when acknowledging FE level exception	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	SV
SR5, 0	PSW	Program status word	*1
SR6, 0	FPSR	(Refer to FPU function registers.)	CU and SV
SR7, 0	FPEPC	(Refer to FPU function registers.)	CU and SV
SR8, 0	FPST	(Refer to FPU function registers.)	CU
SR9, 0	FPCC	(Refer to FPU function registers.)	CU
SR10, 0	FPCFG	(Refer to FPU function registers.)	CU
SR11, 0	FPEC	(Refer to FPU function registers.)	CU and SV
SR13, 0	EIIC	EI level exception source register	SV
SR14, 0	FEIC	FE level exception source register	SV
SR16, 0	CTPC	CALLT execution status save register	UM
SR17, 0	CTPSW	CALLT execution status save register	UM
SR20, 0	CTBP	CALLT base pointer register	UM
SR28, 0	EIWR	EI level exception working register	SV
SR29, 0	FEWR	FE level exception working register	SV
SR0, 1	MCFG0	Machine configuration register	SV
SR2, 1	RBASE	Reset vector base address register	SV
SR3, 1	EBASE	Exception handler vector address register	SV
SR4, 1	INTBP	Base address register of the interrupt handler address table	SV
SR5, 1	MCTL	CPU control	SV
SR6, 1	PID	Processor ID register	SV
SR11, 1	SCCFG	SYSCALL operation setting register	SV
SR12, 1	SCBP	SYSCALL base pointer register	SV
SR0, 2	HTCFG0	Thread configuration register	SV
SR6, 2	MEA	Memory error address register	SV
SR7, 2	ASID	Address space ID register	SV
SR8, 2	MEI	Memory error information register	SV

Note 1. The access permission differs depending on the bit.

(a) EIPC — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see "*Types of Exceptions*" in *Software Manual*).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address cannot be specified.

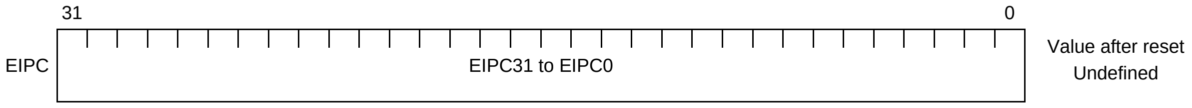


Table 3A.5 EIPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	EIPC31 to EIPC1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC saved when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

(b) EIPSW — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

	31	30	29											17	16	15	14						8	7	6	5	4	3	2	1	0
EIPSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

Value after reset
$0000\ 0020_H$

Table 3A.6 EIPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit stores the PSW.CU field setting when an EI level exception is acknowledged.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

(c) FEPC — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see “*Types of Exceptions*” in *Software Manual*).

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address cannot be specified.

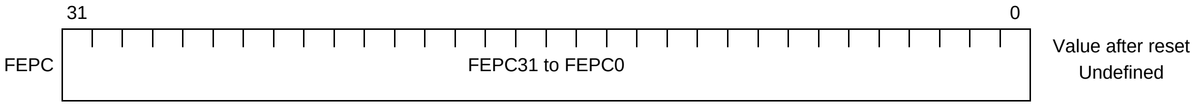


Table 3A.7 FEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	FEPC31 to FEPC1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	Undefined

(d) FEPSW — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

	31	30	29												17	16	15	14						8	7	6	5	4	3	2	1	0
FEPSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

Value after reset
$0000\ 0020_H$

Table 3A.8 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit stores the PSW.CU field setting when an FE level exception is acknowledged.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

(e) PSW — Program Status Word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by conditional instructions (Bcond, CMOV, etc.)).

CAUTIONS

1. When the LDSR instruction is used to change the contents of bit7 to 0 in this register, the changed contents become valid immediately after completion of the LDSR instruction execution. See “*APPENDIX A. Hazard Resolution Procedure for System Registers*” in *Software Manual* when the content of the other bits in this register is changed.
2. The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See **Table 3A.9, Access Permission for PSW Register** for the access permission for each bit.

Table 3A.9 Access Permission for PSW Register

Bit		Access Permission when Reading	Access Permission when Writing
30	UM	UM	SV*1
16	CU	UM	SV*1
15	EBV	UM	SV*1
7	NP	UM	SV*1
6	EP	UM	SV*1
5	ID	UM	SV*1
4	SAT	UM	UM
3	CY	UM	UM
2	OV	UM	UM
1	S	UM	UM
0	Z	UM	UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

	31	30	29												17	16	15	14						8	7	6	5	4	3	2	1	0	
PSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z	Value after reset 0000 0020H

Table 3A.10 PSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit indicates that the CPU is in user mode (in UM mode) 0: Supervisor mode 1: User mode	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit indicates the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor unusable exception occurs if an instruction for the coprocessor is executed or a coprocessor resource (system register) is accessed. CU bit 16: FPU	R/W	0

Table 3A.10 PSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
15	EBV	This bit indicates the reset vector and exception vector operation. See the description on RBASE (**(q) RBASE — Reset Vector Base Address Register**) and EBASE (**(r) EBASE — Exception Handler Vector Address Register**) in this section.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit disables the acknowledgement of FE level exception. When an FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level and FE level exceptions. As for the exceptions which the NP bit disables the acknowledgment, see **Table 7A.1, List of Exception Sources**. 0: The acknowledgement of FE level exception is enabled. 1: The acknowledgement of FE level exception is disabled.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt controlled by the interrupt controller is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0
5	ID	This bit disables the acknowledgement of EI level exception. When an EI level or FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level exception. As for the exceptions which the ID bit disables the acknowledgment, see **Table 7A.1, List of Exception Sources**. This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. The change of the ID bit by the EI or ID instruction will be enabled from the next instruction. 0: EI level exception is not being processed or the section is not a critical section (after execution of EI instruction). 1: EI level exception is being processed or the section is a critical section (after execution of DI instruction).	R/W	1
4	SAT*[1]	This bit indicates that the operation result is saturated because the operation result of a saturated operation instruction has overflowed. This is a cumulative flag, so when the operation result of the saturated operation instruction becomes saturated, this bit is set to 1, but it is not cleared to 0 when the operation result for a subsequent instruction is not saturated. This bit is cleared to 0 by the LDSR instruction. This bit is neither set to 1 nor cleared to 0 when an arithmetic operation instruction is executed. 0: Not saturated 1: Saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV*[1]	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S*[1]	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative.	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. The operation result of the saturation processing is determined in accordance with the contents of the OV flag and S flag during a saturated operation. The SAT flag is set to 1 only when the OV flag is set to 1 in a saturated operation.

Operation Result Status	Flag Status			Operation Result after Saturation Processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF$_H$
Exceeded negative maximum value	1	1	1	8000 0000$_H$
Positive (maximum value not exceeded)	Value prior to operation is retained.	0	0	Operation result itself
Negative (maximum value not exceeded)			1	

(f) EIIC — EI Level Exception Source Register

The EIIC register retains the source of any EI level exception that occurs. The value retained in this register is an exception source code corresponding to a specific exception source.

Table 3A.11 EIIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIIC31 to EIIC0	These bits store the exception source code when an EI level exception is acknowledged. The EIIC15 to EIIC0 field stores the lower 16 bits of the exception source code. The EIIC31 to EIIC16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception, these bits are set to 0.	R/W	0

(g) FEIC — FE Level Exception Source Register

The FEIC register retains the source of any FE level exception that occurs. The value retained in this register is an exception source code corresponding to a specific exception source.

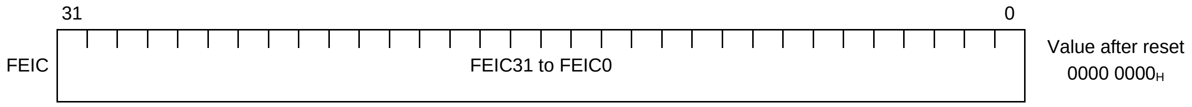


Table 3A.12 FEIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEIC31 to FEIC0	These bits store the exception source code when an FE level exception is acknowledged. The FEIC15 to FEIC0 field stores the lower 16 bits of the exception source code. The FEIC31 to FEIC16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception, these bits are set to 0.	R/W	0

(h) CTPC — Status Save Register when Executing CALLT Instruction

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC. Be sure to set an even-numbered address to the CTPC register. An odd-numbered address cannot be specified.

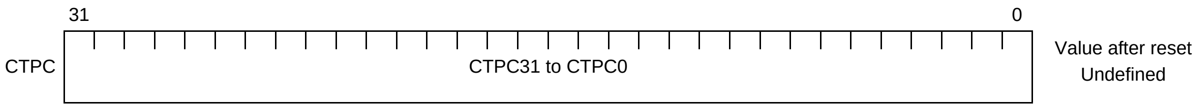


Table 3A.13 CTPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTPC31 to CTPC1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

(i) CTPSW — Status Save Register when Executing CALLT Instruction

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

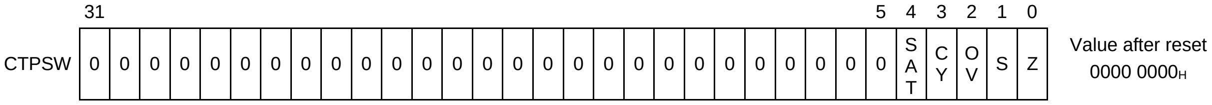


Table 3A.14 CTPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4	SAT	This bit stores the PSW.SAT bit setting when the CALLT instruction is executed.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when the CALLT instruction is executed.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when the CALLT instruction is executed.	R/W	0
1	S	This bit stores the PSW.S bit setting when the CALLT instruction is executed.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when the CALLT instruction is executed.	R/W	0

(j) CTBP — CALLT Base Pointer Register

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

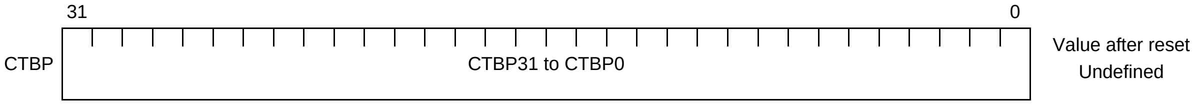


Table 3A.15 CTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTBP31 to CTBP1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the start address of the table used by the CALLT instruction.	R/W	Undefined
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the start address of the table used by the CALLT instruction. Always set this bit to 0.	R	0

(k) ASID — Address Space ID Register

This register indicates the address space ID. This is used to identify the address space provided by the memory management function.

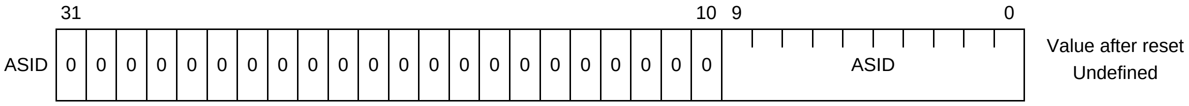


Table 3A.16 ASID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
9 to 0	ASID	These bits indicate the address space ID.	R/W	Undefined

(l) EIWR — EI Level Exception Working Register

The EIWR register is used as a working register when an EI level exception has occurred.

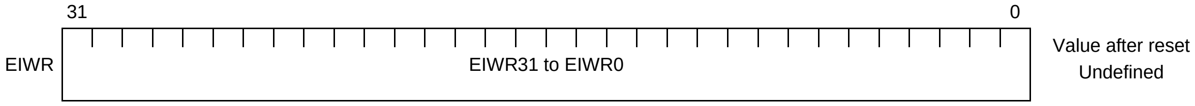


Table 3A.17 EIWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIWR31 to EIWR0	These bits constitute a working register that can be used for any purpose during the servicing of an EI level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(m) FEWR — FE Level Exception Working Register

The FEWR register is used as a working register when an FE level exception has occurred.

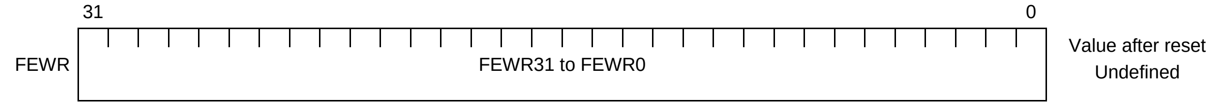


Table 3A.18 FEWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEWR31 to FEWR0	These bits constitute a working register that can be used for any purpose during the servicing of an FE level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(n) HTCFG0 — Thread Configuration Register

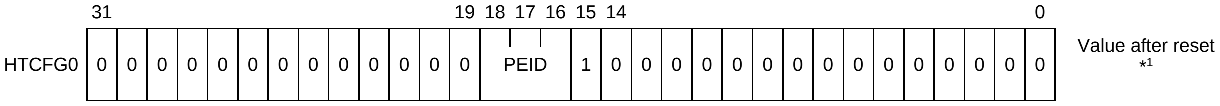


Table 3A.19 HTCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	PEID	These bits indicate the processor element number.	R	*2
15	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
14 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

Note 1. The value after reset is 0001 8000$_H$ for CPU1 (PE1) and 0002 8000$_H$ for CPU2 (PE2).

Note 2. The value after reset is 001$_B$ for CPU1 (PE1) and 010$_B$ for CPU2 (PE2).

(o) MEA — Memory Error Address Register

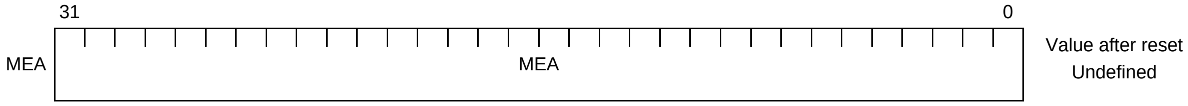


Table 3A.20 MEA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MEA	These bits store the violation address when an MAE (misaligned) or MPU occurs.	R/W	Undefined

(p) MEI — Memory Error Information Register

This register is used to store information about the instruction that caused a misaligned (MAE) or memory protection (MDP) exception when such an exception occurred. This information is used during emulation.

	31										21	20 16	15				11	10 9	8	7	6	5 1	0	
MEI	0	0	0	0	0	0	0	0	0	0	0	REG	0	0	0	0	0	DS		0	0	ITYPE	RW	Value after reset Undefined

Table 3A.21 MEI Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	REG	These bits indicate the number of the source or destination register accessed by the instruction that caused the exception. For details, see **Table 3A.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
15 to 11	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
10, 9	DS	These bits indicate the data type of the instruction that caused the exception.*1 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) For details, see **Table 3A.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
8	U	This bit indicates the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see **Table 3A.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 1	ITYPE	These bits indicate the instruction that caused the exception. For details, see **Table 3A.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
0	RW	This bit indicates whether the operation of the instruction that caused the exception was read (Load-memory) or write (Store-memory). 0: Read (Load-memory) 1: Write (Store-memory) For details, see **Table 3A.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3A.22 Instructions Causing Exceptions and Values of MEI Register

Instruction	REG	DS	U	RW	ITYPE
SLD.B	dst	0 (byte)	0 (signed)	0 (read)	00000_B
SLD.BU	dst	0 (byte)	1 (unsigned)	0 (read)	00000_B
SLD.H	dst	1 (halfword)	0 (signed)	0 (read)	00000_B
SLD.HU	dst	1 (halfword)	1 (unsigned)	0 (read)	00000_B
SLD.W	dst	2 (word)	0 (signed)	0 (read)	00000_B
SST.B	src	0 (byte)	0 (signed)	1 (write)	00000_B
SST.H	src	1 (halfword)	0 (signed)	1 (write)	00000_B

Table 3A.22 Instructions Causing Exceptions and Values of MEI Register

Instruction	REG	DS	U	RW	ITYPE
SST.W	src	2 (word)	0 (signed)	1 (write)	00000_B
LD.B (disp16)	dst	0 (byte)	0 (signed)	0 (read)	00001_B
LD.BU (disp16)	dst	0 (byte)	1 (unsigned)	0 (read)	00001_B
LD.H (disp16)	dst	1 (halfword)	0 (signed)	0 (read)	00001_B
LD.HU (disp16)	dst	1 (halfword)	1 (unsigned)	0 (read)	00001_B
LD.W (disp16)	dst	2 (word)	0 (signed)	0 (read)	00001_B
ST.B (disp16)	src	0 (byte)	0 (signed)	1 (write)	00001_B
ST.H (disp16)	src	1 (halfword)	0 (signed)	1 (write)	00001_B
ST.W (disp16)	src	2 (word)	0 (signed)	1 (write)	00001_B
LD.B (disp23)	dst	0 (byte)	0 (signed)	0 (read)	00010_B
LD.BU (disp23)	dst	0 (byte)	1 (unsigned)	0 (read)	00010_B
LD.H (disp23)	dst	1 (halfword)	0 (signed)	0 (read)	00010_B
LD.HU (disp23)	dst	1 (halfword)	1 (unsigned)	0 (read)	00010_B
LD.W (disp23)	dst	2 (word)	0 (signed)	0 (read)	00010_B
ST.B (disp23)	src	0 (byte)	0 (signed)	1 (write)	00010_B
ST.H (disp23)	src	1 (halfword)	0 (signed)	1 (write)	00010_B
ST.W (disp23)	src	2 (word)	0 (signed)	1 (write)	00010_B
LD.DW (disp23)	dst	3 (double-word)	0 (signed)	0 (read)	00010_B
ST.DW (disp23)	src	3 (double-word)	0 (signed)	1 (write)	00010_B
LDL.W	dst	2 (word)	0 (signed)	0 (read)	00111_B
STC.W	src	2 (word)	0 (signed)	1 (write)	00111_B
CAXI	dst	2 (word)	0 (signed)	0 (read)/1 (write)	01000_B
SET1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
CLR1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
NOT1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
TST1	—	0 (byte)	0 (signed)	0 (read)	01001_B
PREPARE	—	2 (word)	0 (signed)	1 (write)	01100_B
DISPOSE	—	2 (word)	0 (signed)	0 (read)	01100_B
PUSHSP	—	2 (word)	0 (signed)	1 (write)	01101_B
POPSP	—	2 (word)	0 (signed)	0 (read)	01101_B
SWITCH	—	1 (halfword)	0 (signed)	0 (read)	10000_B
CALLT	—	1 (halfword)	1 (unsigned)	0 (read)	10001_B
SYSCALL	—	2 (word)	0 (signed)	0 (read)	10010_B
CACHE	—	—	—	—	—
Interrupt (table reference)*1	—	2 (word)	0 (signed)	0 (read)	10101_B

Note 1. When reading the interrupt vector by using the table reference method.

NOTE

dst: destination register number, src: source register number

(q) RBASE — Reset Vector Base Address Register

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this vector address is also used as the exception vector address.

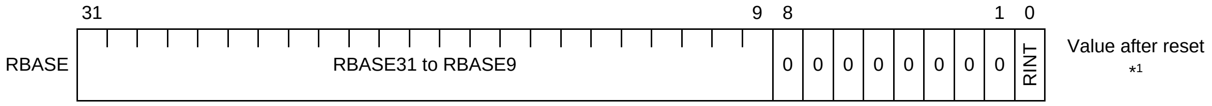


Table 3A.23 RBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	RBASE31 to RBASE9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. For RBASE8 to RBASE0, 0 is used implicitly.	R	CPU1: 0000 0000 0000 0000 0000 000$_B$ CPU2: 0000 0000 1000 0000 0000 000$_B$*1
8 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	RINT	When the RINT bit is set, the exception handler address for interrupt servicing is reduced. See **Section 7A.10.1, Direct Vector Method**. This bit is valid when PSW.EBV = 0.	R	0

Note 1. The value depends on the reset vector. The values set at shipment are shown in the table. When the reset vector is modified, the address will be changed.

(r) EBASE — Exception Handler Vector Address Register

This register indicates the exception handler vector address. This register is valid when the PSW.EBV bit is 1.

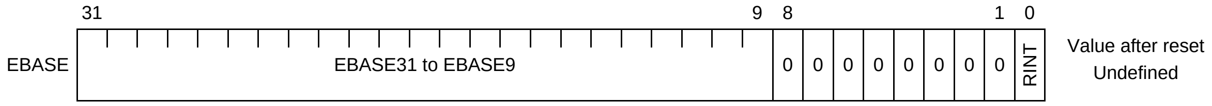


Table 3A.24 EBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	EBASE31 to EBASE9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. For EBASE8 to EBASE0, 0 is used implicitly.	R/W	Undefined
8 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	RINT	When the RINT bit is set, the exception handler address for interrupt servicing is reduced. See **Section 7A.10.1, Direct Vector Method**.	R/W	Undefined

(s) INTBP — Base Address Register of the Interrupt Handler Address Table

This register indicates the base address of the address table when the table reference method is selected as the interrupt handler address selection method.

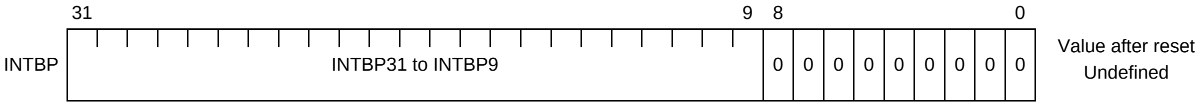


Table 3A.25 INTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	INTBP31 to INTBP9	These bits indicate the base pointer address for an interrupt when the table reference method is used. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt specified by the table reference method (EIINT0 to EIINT511) is acknowledged. For INTBP8 to INTBP0, 0 is used implicitly.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(t) PID — Processor ID Register

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

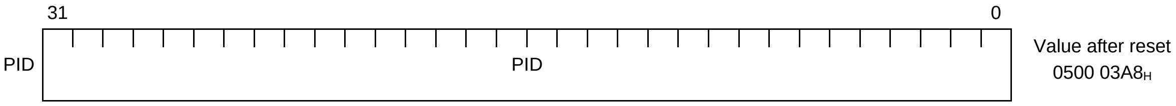


Table 3A.26 PID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.	R	05_H
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bits 23 to 11: Reserved Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operation function Bit 8: Memory protection function (MPU)	R	0003_H
7 to 0		Version Identifier This identifier indicates the version of the processor.	R	$A8_H$

(u) SCCFG — SYSCALL Operation Setting Register

This register is used to specify operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

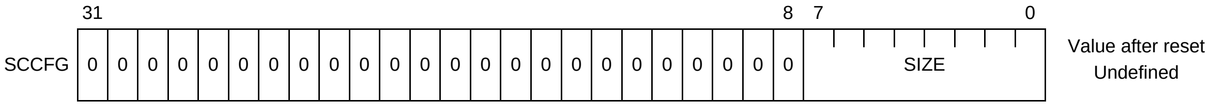


Table 3A.27 SCCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SIZE	These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If vectors exceeding the maximum number of entries are specified for the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry.	R/W	Undefined

(v) SCBP — SYSCALL Base Pointer Register

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

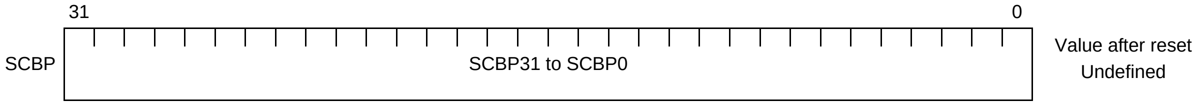


Table 3A.28 SCBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	SCBP31 to SCBP2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction.	R/W	Undefined
1, 0	SCBP1, SCBP0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction. Always set these bits to 0.	R	0

(w) MCFG0 — Machine Configuration Register

This register indicates the CPU configuration.

	31													18	17	16	15												3	2	1	0	
MCFG0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPID		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Value after reset *1

Table 3A.29 MCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 18	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
17, 16	SPID	These bits indicate the system protection number.	R/W	*2
15 to 3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

Note 1. The value after reset is $0001\ 0004_H$ for CPU1 (PE1) and $0002\ 0004_H$ for CPU2 (PE2).

Note 2. The value after reset is 01_B for CPU1 (PE1) and 10_B for CPU2 (PE2).

(x) MCTL — Machine Control Register

This register is used to control the CPU.

	31																													2	1	0	
MCTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MA	UIC	Value after reset $8000\ 0002_H$

Table 3A.30 MCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
30 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	MA	This bit is used to control the misaligned access. 0: In the event of a misaligned access, an exception is always generated.*1 1: The correct operation is controlled by hardware.*2	R/W	1
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction in user mode becomes possible.	R/W	0

Note 1. Excluding LD.DW, and ST.DW for word boundary allocation.

Note 2. Exception still occurs in case of LD.DW or ST.DW for misaligned access except word boundary allocation.

(3) Interrupt Function Registers

Table 3A.31 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR7, 1	FPIPR	FPI exception interrupt priority setting register	SV
SR10, 2	ISPR	Priority of interrupt being serviced register	SV
SR11, 2	PMR	Interrupt priority masking register	SV
SR12, 2	ICSR	Interrupt control status register	SV
SR13, 2	INTCFG	Interrupt function setting register	SV

(a) FPIPR — FPI Exception Interrupt Priority Setting Register

This register is used to set the interrupt priority of FPI exception.

	31																										5	4 0	
FPIPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FPIPR	Value after reset $0000\ 0000_H$

Table 3A.32 FPIPR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	FPIPR	These bits are used to specify the interrupt priority of floating-point operation exceptions (imprecise) (FPI). Specify values from 0 to 16. Specifying 17 or greater is prohibited. FPI exceptions are handled using the specified interrupt priority. If an FPI exception occurs at the same time as an interrupt that has the same priority, the FPI exception is prioritized. **NOTE:** A set value of more than 16 is treated as 16.	R/W	0

(b) ISPR — Priority of Interrupt being Serviced Register

This register retains the priority of the *EIINTn* interrupt being serviced by the CPU. This priority value is then used to perform priority ceiling processing when multiple interrupts occur.

	31															16	15 0
ISPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISP15 to ISP0

Value after reset
0000 0000$_H$

Table 3A.33 ISPR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	ISP15 to ISP0	These bits indicate the acknowledgment status of an *EIINTn* interrupt with a priority that corresponds to the relevant bit position. 0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged. 1: An interrupt request for an interrupt whose priority corresponds to the relevant bit position is being serviced by the CPU core. The bit positions correspond to the following priority levels. (see table below) When an interrupt request (*EIINTn*) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15 to ISP0 bits that are set to 1 (0 is the highest priority) is cleared to 0.*[1] While a bit in this register is set to 1, same or lower priority interrupts (*EIINTn*) and FPI exceptions*[2] are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged. When performing software-based priority control using the PMR register, be sure to clear this register by using the INTCFG.ISPC bit.	R*[3]	0

Bit	Priority
0	Priority 0 (highest)
1	Priority 1
:	
14	Priority 14
15	Priority 15 (lowest)

Note 1. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) to the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.

Note 2. Since FPI exceptions have the same level of priority as EIINTn interrupts, they are affected by interrupts in the same way as the ISPR. The priority of FPI exceptions is set by the FPIPR register.

Note 3. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

(c) PMR — Interrupt Priority Masking Register

This register is used to mask the specified interrupt priority.

	31															16	15 0	
PMR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PM15 to PM0	Value after reset 0000 0000$_H$

Table 3A.34 PMR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	PM15 to PM0	These bits mask an interrupt request with a priority level that corresponds to the relevant bit position. 0: Servicing of an interrupt with a priority that corresponds to the relevant bit position is enabled. 1: Servicing of an interrupt with a priority that corresponds to the relevant bit position is disabled. The bit positions correspond to the following priority levels: Bit 0: Priority 0 (highest) Bit 1: Priority 1 : Bit 14: Priority 14 Bit 15: Priority 15 and priority 16 (lowest) While a bit in this register is set to 1, interrupts (*EIINTn*) and FPI exceptions*[1] with the priority corresponding to that bit are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged*[2].	R/W	0

Note 1. Since FPI exceptions are specified as the same level of priority as that of interrupts (*EIINTn*), it is affected by the PMR like interrupts. The priority of FPI exceptions is set by the FPIPR register.

Note 2. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, FF00$_H$ can be set, but F0F0$_H$ or 00FF$_H$ cannot.

(d) ICSR — Interrupt Control Status Register

This register indicates the interrupt control status in the CPU.

	31																													2	1	0	
ICSR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PMFP	PMEI	Value after reset 0000 0000$_H$

Table 3A.35 ICSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	PMFP	This bit indicates that an FPI exception with the priority level masked by the PMR register exists.	R	0
0	PMEI	This bit indicates that an interrupt (*EIINTn*) with the priority level masked by the PMR register exists.	R	0

(e) INTCFG — Interrupt Function Setting Register

This register is used to specify settings related to the CPU’s internal interrupt function.

	31																														1	0
INTCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISPC

Value after reset
0000 0000$_H$

Table 3A.36 INTCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	ISPC	This bit specifies how the ISPR register is updated. 0: The ISPR register is automatically updated. Updates triggered by the program (via execution of LDSR instruction) are ignored. 1: The ISPR register is not automatically updated. Updates triggered by the program (via execution of LDSR instruction) are performed. If this bit is cleared to 0, the bits of the ISPR register are automatically set to 1 when an interrupt (*EIINTn*) is acknowledged, and cleared to 0 when the EIRET instruction is executed. In this case, updating by the program (via execution of an LDSR instruction) is ignored. If this bit is set to 1, the bits of the ISPR register are not updated by the acknowledgement of an interrupt (*EIINTn*) or by execution of the EIRET instruction. In this case, the bits can be updated by an LDSR instruction executed by the program. In normal cases, the ISPC bit should be cleared to 0. When performing software-based control of interrupt priorities, however, set this bit (1) and perform priority control by using the PMR register.	R/W	0

(4) FPU Function Registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations. The RH850/F1KH supports single-precision floating-point instruction and thirty-two 32-bit registers can be specified.

These registers correspond to general-purpose registers r0 to r31. The FPU can use the following system registers to control floating-point operation

Table 3A.37 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Floating-point operation setting/status register	CU and SV
SR7, 0	FPEPC	Floating-point exception program counter register	CU and SV
SR8, 0	FPST	Floating-point operation status register	CU
SR9, 0	FPCC	Floating-point operation comparison result register	CU
SR10, 0	FPCFG	Floating-point operation configuration register	CU
SR11, 0	FPEC	Floating-point exception control register	CU and SV

(a) FPSR — Floating-point Operation Setting/Status Register

This register indicates the execution status of floating-point operations and any exceptions that occur.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPSR	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	FN	IF	PEM	0	RM		FS	0	E	V	Z	O	U	I	V	Z	O	U	I	V	Z	O	U	I
																			*1						*2					*3		

Value after reset
See below

Note 1. Cause bits (XC)

Note 2. Enable bits(XE)

Note 3. Preservation bits (XP)

Table 3A.38 FPSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 24	CC[7:0]	These are the CC (condition) bits. They store the results of floating-point comparison instructions. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined
23	FN	This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number.	R/W	0
22	IF	This bit accumulates and indicates information about the flushing of input operands.	R/W	0
21	PEM	This bit specifies whether to handle an exception as a precise exception. If the PEM bit is 1, exceptions that are caused by the execution of a floating-point operation instruction are handled as precise exceptions.	R/W	0
20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
19, 18	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.	R/W	00

RM Bits		Mnemonic	Description
19	18		
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.
1	0	RP	Rounds the result toward +∞. The result is nearest to a value greater than the accurate result with infinite accuracy.
1	1	RM	Rounds the result toward −∞. The result is nearest to a value less than the accurate result with infinite accuracy.

Bit Position	Bit Name	Function	R/W	Value after Reset
17	FS	This bit enables values that cannot be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign. Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode.	R/W	1

Operation Result that is a Subnormal Number	Rounding Mode and Value after Flushing			
	RN*1	RZ	RP	RM
Positive	+0	+0	$+2^{Emin}$	+0
Negative	−0	−0	−0	-2^{Emin}

Note 1. If the rounding mode is RN and the FPSR.FN bit is set to 1, flushing will occur in the direction of higher accuracy.

Bit Position	Bit Name	Function	R/W	Value after Reset
16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 10	XC (E, V, Z, O, U, I)	These are the cause bits.	R/W	Undefined

Table 3A.38 FPSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
9 to 5	XE (V, Z, O, U, I)	These are the enable bits.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are the preservation bits.	R/W	Undefined

(b) FPEPC — Floating-point Exception Program Counter Register

When an exception that is enabled by an enable bit occurs, the program counter (PC) of the instruction that caused the exception is stored.

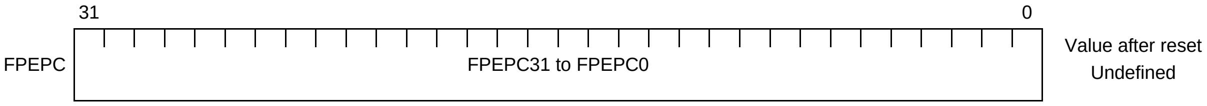
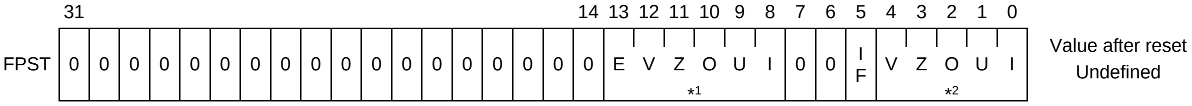


Table 3A.39 FPEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	FPEPC31 to FPEPC1	These bits store the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R/W	Undefined
0	FPEPC0	This bit stores the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs. Always set this bit to 0.	R	0

(c) FPST — Floating-point Operation Status Register

This register reflects the contents of the FPSR register bits related to the operation status.



Note 1. Cause bits (XC)

Note 2. Preservation bits (XP)

Table 3A.40 FPST Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E, V, Z, O, U, I)	These are cause bits. Values written to these bits are reflected in FPSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are preservation bits. Values written to these bits are reflected in FPSR.XP bits.	R/W	Undefined

(d) FPCC — Floating-point Operation Comparison Result Register

This register reflects the contents of the FPSR.CC[7:0] bits.

	31																							8	7	6	5	4	3	2	1	0	
FPCC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	Value after reset Undefined

Table 3A.41 FPCC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	CC[7:0]	These are CC (condition) bits. They store the result of a floating-point comparison instruction. The CC[7:0] bits are not affected by any instructions except the comparison instruction and LDSR instruction. Values written to these bits are reflected in the CC[7:0] bits of FPSR. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined

(e) FPCFG — Floating-point Operation Configuration Register

This register reflects the contents of the FPSR register bits related to the operation settings.

	31																					10	9	8	7		5	4	3	2	1	0	
FPCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RM		0	0	0	V	Z	O *1	U	I	Value after reset 0000 0000$_H$

Note 1. Enable bits (XE)

Table 3A.42 FPCFG Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value after Reset</th></tr>
<tr><td>31 to 10</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>9, 8</td><td>RM</td><td>These are rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.
Values written to these bits are reflected in RM bits of FPSR.
<table>
<tr><th colspan="2">RM Bits</th><th rowspan="2">Mnemonic</th><th rowspan="2">Description</th></tr>
<tr><th>9</th><th>8</th></tr>
<tr><td>0</td><td>0</td><td>RN</td><td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td></tr>
<tr><td>0</td><td>1</td><td>RZ</td><td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td></tr>
<tr><td>1</td><td>0</td><td>RP</td><td>Rounds the result toward +∞. The result is nearest to a value greater than the accurate result with infinite accuracy.</td></tr>
<tr><td>1</td><td>1</td><td>RM</td><td>Rounds the result toward −∞. The result is nearest to a value less than the accurate result with infinite accuracy.</td></tr>
</table></td><td>R/W</td><td>0</td></tr>
<tr><td>7 to 5</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>4 to 0</td><td>XE
(V, Z, O, U, I)</td><td>These are the enable bits.</td><td>R/W</td><td>0</td></tr>
</table>

(f) FPEC — Floating-point Exception Control Register

This register controls the floating-point operation exception.

	31																														1	0	
FPEC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FPIVD	Value after reset $0000\ 0000_H$

Table 3A.43 FPEC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	FPIVD*[1]	This bit indicates the status of reporting the FPI exception. If this bit is set to 1, the FPI exception is reported to the CPU but is not acknowledged. It is automatically cleared to 0 when the CPU acknowledges the FPI exception. While this bit is set to 1, all the floating-point instructions are invalidated. Report of the FPI exception can be canceled by clearing (0) this bit by the LDSR instruction while it is set to 1. When report of the FPI exception is canceled, the CPU does not acknowledge the FPI exception. 0: FPI exception is not reported. 1: FPI exception is reported.	R/W	0

Note 1. The FPIVD bit can only be cleared to 0 by the write operation of the LDSR instruction. It cannot be set to 1.

(5) MPU Function Registers

Table 3A.44 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	SV
SR1, 5	MPRC	MPU region control	SV
SR4, 5	MPBRGN	MPU base region number	SV
SR5, 5	MPTRGN	MPU end region number	SV
SR8, 5	MCA	Memory protection setting check address	SV
SR9, 5	MCS	Memory protection setting check size	SV
SR10, 5	MCC	Memory protection setting check command	SV
SR11, 5	MCR	Memory protection setting check result	SV
SR0, 6	MPLA0	Protection area lower limit address	SV
SR1, 6	MPUA0	Protection area upper limit address	SV
SR2, 6	MPAT0	Protection area attribute	SV
SR4, 6	MPLA1	Protection area lower limit address	SV
SR5, 6	MPUA1	Protection area upper limit address	SV
SR6, 6	MPAT1	Protection area attribute	SV
SR8, 6	MPLA2	Protection area lower limit address	SV
SR9, 6	MPUA2	Protection area upper limit address	SV
SR10, 6	MPAT2	Protection area attribute	SV
SR12, 6	MPLA3	Protection area lower limit address	SV
SR13, 6	MPUA3	Protection area upper limit address	SV
SR14, 6	MPAT3	Protection area attribute	SV
SR16, 6	MPLA4	Protection area lower limit address	SV
SR17, 6	MPUA4	Protection area upper limit address	SV
SR18, 6	MPAT4	Protection area attribute	SV
SR20, 6	MPLA5	Protection area lower limit address	SV
SR21, 6	MPUA5	Protection area upper limit address	SV
SR22, 6	MPAT5	Protection area attribute	SV
SR24, 6	MPLA6	Protection area lower limit address	SV
SR25, 6	MPUA6	Protection area upper limit address	SV
SR26, 6	MPAT6	Protection area attribute	SV
SR28, 6	MPLA7	Protection area lower limit address	SV
SR29, 6	MPUA7	Protection area upper limit address	SV
SR30, 6	MPAT7	Protection area attribute	SV
SR0, 7	MPLA8	Protection area lower limit address	SV
SR1, 7	MPUA8	Protection area upper limit address	SV
SR2, 7	MPAT8	Protection area attribute	SV
SR4, 7	MPLA9	Protection area lower limit address	SV
SR5, 7	MPUA9	Protection area upper limit address	SV
SR6, 7	MPAT9	Protection area attribute	SV
SR8, 7	MPLA10	Protection area lower limit address	SV
SR9, 7	MPUA10	Protection area upper limit address	SV
SR10, 7	MPAT10	Protection area attribute	SV
SR12, 7	MPLA11	Protection area lower limit address	SV
SR13, 7	MPUA11	Protection area upper limit address	SV

Table 3A.44 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR14, 7	MPAT11	Protection area attribute	SV
SR16, 7	MPLA12	Protection area lower limit address	SV
SR17, 7	MPUA12	Protection area upper limit address	SV
SR18, 7	MPAT12	Protection area attribute	SV
SR20, 7	MPLA13	Protection area lower limit address	SV
SR21, 7	MPUA13	Protection area upper limit address	SV
SR22, 7	MPAT13	Protection area attribute	SV
SR24, 7	MPLA14	Protection area lower limit address	SV
SR25, 7	MPUA14	Protection area upper limit address	SV
SR26, 7	MPAT14	Protection area attribute	SV
SR28, 7	MPLA15	Protection area lower limit address	SV
SR29, 7	MPUA15	Protection area upper limit address	SV
SR30, 7	MPAT15	Protection area attribute	SV

(a) MPM — Memory Protection Operation Mode Register

The memory protection mode register is used to define the basic operating state of the memory protection function.

	31																													2	1	0
MPM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SVP	MPE

Value after reset
0000 0000$_H$

Table 3A.45 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	SVP	In SV mode (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*1 0: As usual, implicitly enable all access in SV mode. 1: Restrict access according to the SX, SW, and SR bits even in SV mode.*2	R/W	0
0	MPE	This bit is used to specify whether to enable or disable the MPU function. 0: Disable 1: Enable	R/W	0

Note 1. If the SVP bit is set to 1, access will be restricted in accordance with the setting for each protection area, even in SV mode. Therefore, specify the protection area beforehand so that the access from the program which set the SVP bit is not restricted.

Note 2. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(b) MPRC — MPU Region Control Register

	31															16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Value after reset
0000 0000$_H$

Table 3A.46 MPRC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	E15 to E0	These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 15 to 0).	R/W	0

(c) MPBRGN — MPU Base Region Register

This register indicates the minimum usable MPU area number.

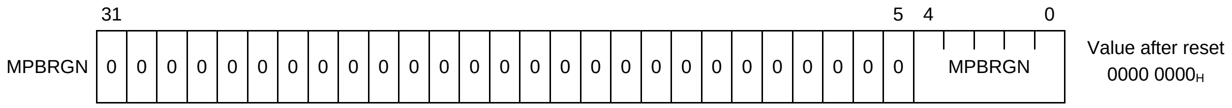


Table 3A.47 MPBRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	MPBRGN	These bits indicate the smallest number of an MPU area. These bits always indicate 0.	R	0

(d) MPTRGN — MPU End Region Register

This register indicates the maximum usable MPU area number + 1.

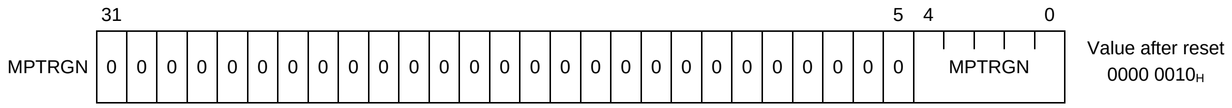


Table 3A.48 MPTRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	MPTRGN	These bits indicate the largest number of an MPU area + 1. These bits indicate the maximum number of MPU areas incorporated into the hardware.	R	10000_B

(e) MCA — Memory Protection Setting Check Address Register

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

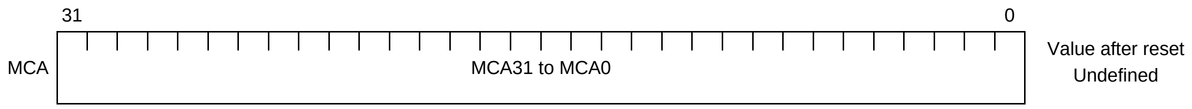


Table 3A.49 MCA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCA31 to MCA0	These bits are used to specify the start address of the memory area that is subject to a memory protection setting check in bytes.	R/W	Undefined

(f) MCS — Memory Protection Setting Check Size Register

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

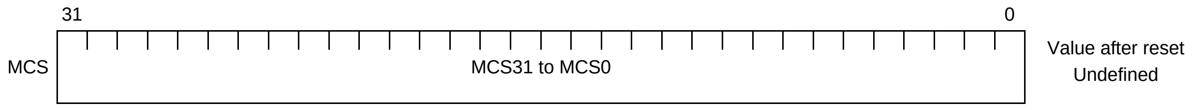


Table 3A.50 MCS Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCS31 to MCS0	These bits are used to specify the size of the memory area that is subject to a memory protection setting check and the size of the target area in bytes. Because the specified size is assumed to represent an unsigned integer, it is not possible to check an area in the direction in which the address value decreases relative to the MCA register value. Do not specify 0000 0000_H for the MCS register.	R/W	Undefined

(g) MCC — Memory Protection Setting Check Command Register

This register is used to specify the base address of the area where memory protection settings are checked.

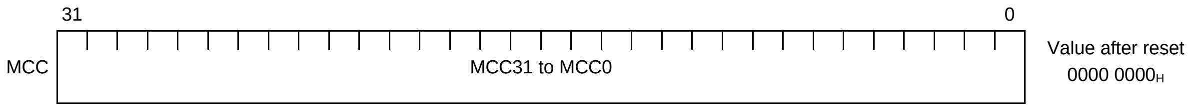


Table 3A.51 MCC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCC31 to MCC0	When any value is written to the MCC register, a memory protection setting check starts. By setting up the MCA/MCS register and then writing to the MCC register, results are stored in MCR. Because the check is started by any written value, a check can be started by using r0 as the source register without using any unnecessary registers. Note that, for the check, the results are applied according to each area setting regardless of the state of the PSW.UM bit. When the MCC register is read, value 0000 0000_H is always returned.	R/W	0

(h) MCR — Memory Protection Setting Check Result Register

This register is used to store the results of a memory protection setting check.

Be sure to clear bits 31 to 9, 7 and 6.

	31																						9	8	7	6	5	4	3	2	1	0	
MCR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OV	0	0	SXE	SWE	SRE	UXE	UWE	URE	Value after reset Undefined

Table 3A.52 MCR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	OV	If the specified area includes 0000 0000$_H$ or 7FFF FFFF$_H$, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
4	SWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
3	SRE	If the specified area is contained within one of the protection areas and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
2	UXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
1	UWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
0	URE	If the specified area is contained within one of the protection areas and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

(i) MPLAn — Protection Area Lower Limit Address Register

These registers indicate the lower limit address of area n (where n = 0 to 15).

	31	2	1	0	
MPLAn	MPLAn		0	0	Value after reset Undefined

Table 3A.53 MPLAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPLA31 to MPLA2	These bits indicate the lower limit address of area n. For MPLA1 and MPLA0, 0 is used implicitly.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(j) MPUAn — Protection Area Upper Limit Address Register

These registers indicate the upper limit address of area n (where n = 0 to 15).

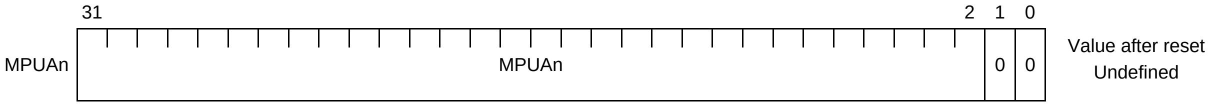


Table 3A.54 MPUAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPUA31 to MPUA2	These bits indicate the upper limit address of area n. For MPUAn.MPUA1 and MPUA0, 1 is used implicitly.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(k) MPATn — Protection Area Attribute Register

These registers indicate the attributes of area n (where n = 0 to 15).

	31					26	25 16	15							8	7	6	5	4	3	2	1	0	
MPATn	0	0	0	0	0	0	ASID	0	0	0	0	0	0	0	0	E	G	SX	SW	SR	UX	UW	UR	Value after reset Undefined

Table 3A.55 MPATn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 26	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
25 to 16	ASID	These bits indicate the ASID value to be used as the area match condition.	R/W	Undefined
15 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	E	This bit indicates whether area n is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled.	R/W	0
6	G	0: Areas match only if ASIDs are equal. 1: Areas match even if ASIDs are not equal. If this bit is 0, MPATn.ASID = ASID.ASID is used as the area match condition. If this bit is 1, areas may match even if the values of MPATn.ASID and ASID.ASID are not equal.	R/W	Undefined
5	SX	This bit indicates the execution privilege for the supervisor mode.*1 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
4	SW	This bit indicates whether writing is enabled in the supervisor mode.*1 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
3	SR	This bit indicates whether writing is enabled in the supervisor mode.*1 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined
2	UX	This bit indicates the execution privilege for the user mode. 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
1	UW	This bit indicates whether writing is enabled in the user mode. 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
0	UR	This bit indicates whether writing is enabled in the user mode. 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Note 1. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(6) Cache Operation Function Registers

The RH850/F1KH does not include a cache operation function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

Table 3A.56 Cache Operation Function Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR12, 4	BWERRL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR13, 4	BWERRH		SV
SR14, 4	BRERRL		SV
SR15, 4	BRERRH		SV
SR16, 4	ICTAGL		SV
SR17, 4	ICTAGH		SV
SR18, 4	ICDATL		SV
SR19, 4	ICDATH		SV
SR20, 4	DCTAGL		SV
SR21, 4	DCTAGH		SV
SR22, 4	DCDATL		SV
SR23, 4	DCDATH		SV
SR24, 4	ICCTRL		SV
SR25, 4	DCCTRL		SV
SR26, 4	ICCFG		SV
SR27, 4	DCCFG		SV
SR28, 4	ICERR		SV
SR29, 4	DCERR		SV

3A.2.1.3 Instruction

See “*Instruction*” in *Software Manual*.

A snooze instruction halts operation of the CPU1/2 core for 32 clocks.

3A.2.2 Buffers for Code Flash

3A.2.2.1 Features

CPUn (n = 1, 2) accesses Code flash by two paths; instruction fetch access is direct to Flash interface, and data access is via System interconnect to Code Flash. Both paths equip buffers, which can be cleared by software. See also **Figure 40A.1, Block Diagram of Code Flash ECC** for ECC decoders in these paths.

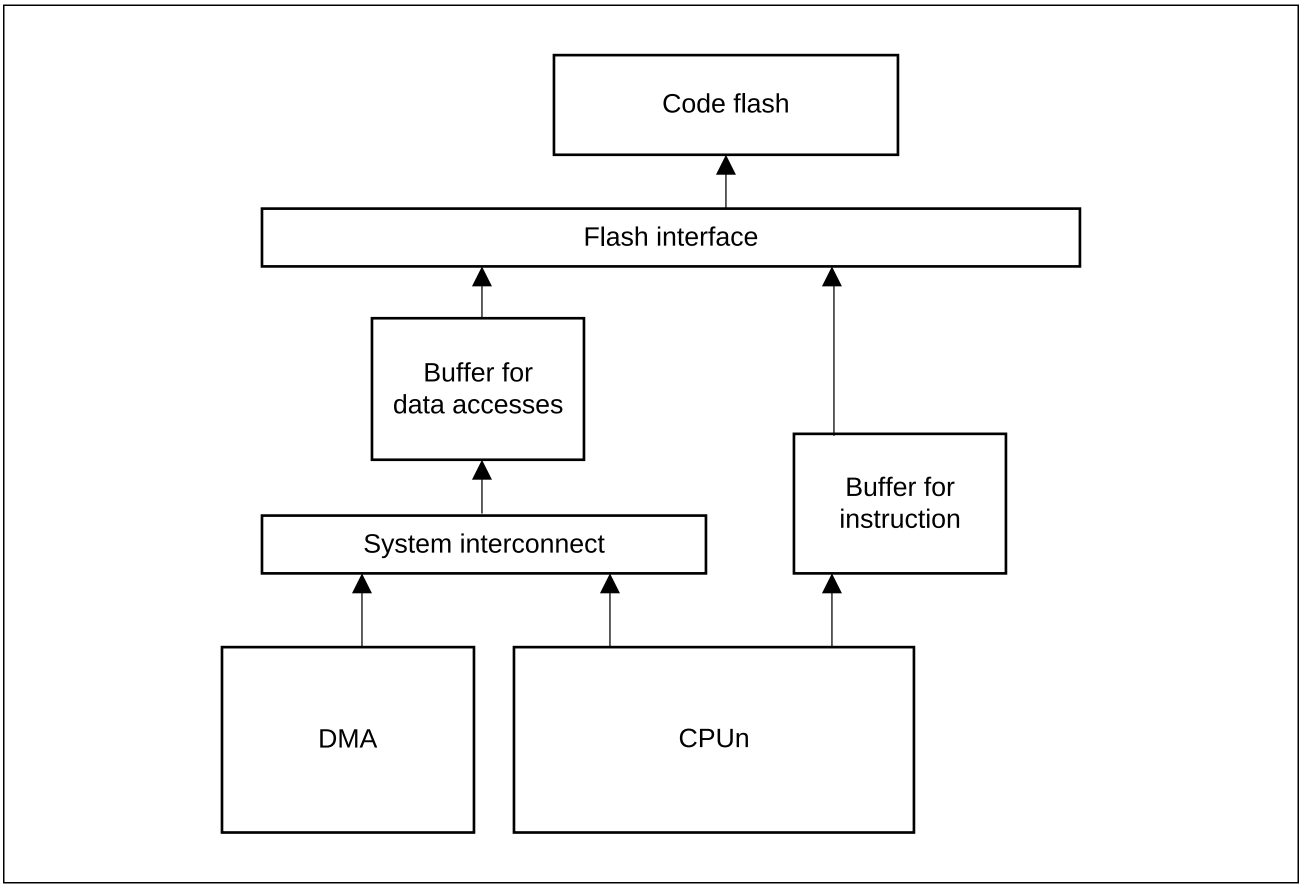


Figure 3A.2 Buffers for Code Flash

3A.2.2.2 Function of Buffers

One-line buffer with 128 bits is mounted for instruction fetches to code flash. The data is read out from the buffer after the next access to the same address, so the code flash is not accessed again within 128 bits location.

Two-line buffer with 128 bits is mounted as a data buffer. This buffer is not only used by the CPU but also used by DMA via system interconnect. The data is read out from the buffer if the next access is within the same 128 bits boundary.

16 entry branch history buffer is mounted to decrease branch penalty.
One buffer control register named FBUFCCTL is equipped. Using the FBUFCCTL register, the software can clear these three buffers.

3A.2.2.3 Registers for Buffer Control

(1) List of Buffer Control Registers

Table 3A.57 Buffer Control Register (Base Address: FFC5 B000$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after Reset
							1	8	16	32	
FBUF_CTRL	+000$_H$	4	Flash buffer clear control register	FBUFCCTL	—	R/W	—	✓	✓	✓	0000 0000$_H$

(2) Register Sets

Access: FBUFCCTL register can be read or written in 32-bit units.
FBUFCCTLL register can be read or written in 16-bit units.
FBUFCCTLLL register can be read or written in 8-bit units.

Address: FBUFCCTL: FFC5 B000$_H$
FBUFCCTLL: FFC5 B000$_H$
FBUFCCTLLL: FFC5 B000$_H$

	31																														1	0	
FBUFCCTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FBUFCLR	Value after reset 0000 0000$_H$

Table 3A.58 FBUFCCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	R	0
0	FBUFCLR	Buffer clear bits. To clear buffers, write 1 to this bit and then write 0. 0: Buffers are valid 1: Buffers are invalid (cleared)	R/W	0

Please do following procedures when you want to clear the buffers.

Step 1: Write 0 to FBUFCCTL.FBUFCLR

Step 2: Write 1 to FBUFCCTL.FBUFCLR

Step 3: Write 0 to FBUFCCTL.FBUFCLR

Step 4: Read the FBUFCCTL register (dummy read)

Step 5: Execute the SYNCP instruction

Step 6: Execute the SYNCI instruction

If you do not do Step 3 after Step 2, the buffers are kept invalid during FBUFCCTL.FBUFCLR = 1.

3A.2.3 Inter-Processor Interrupts

Four registers (IPIR_CHn) are provided for communicating four channels of interrupts between the CPUs (PEs).

IPIR_CH0 to IPIR_CH3 are assigned to CH0 to CH3 of user interrupts (EIINT). An interrupt for specific PEs (including own PE) can be requested by manipulating the bits corresponding to each PE.

3A.2.3.1 Inter-Processor Interrupt Control Registers

These registers are located in the CPU Peripheral.

Table 3A.59 List of Registers

Module Name	Address	Register Name	Symbol	R/W	Access Size				Value after Reset
					1	8	16	32	
IPIRSS	FFFE EC80$_H$	Inter-PE interrupt register 0	IPIR_CH0	R/W	—	✓	✓	✓	0000 0000$_H$
	FFFE EC84$_H$	Inter-PE interrupt register 1	IPIR_CH1	R/W	—	✓	✓	✓	0000 0000$_H$
	FFFE EC88$_H$	Inter-PE interrupt register 2	IPIR_CH2	R/W	—	✓	✓	✓	0000 0000$_H$
	FFFE EC8C$_H$	Inter-PE interrupt register 3	IPIR_CH3	R/W	—	✓	✓	✓	0000 0000$_H$

(1) IPIR_CHn — Inter-PE Interrupt Register n (n = 0 to 3)

Access: IPIR_CHn can be read or written in 32-bit units.
IPIR_CHnL can be read or written in 16-bit units.
IPIR_CHnLL can be read or written in 8-bit units.

Address: IPIR_CHn: FFFE EC80$_H$ + n × 4$_H$
IPIR_CHnL: FFFE EC80$_H$ + n × 4$_H$
IPIR_CHnLL: FFFE EC80$_H$ + n × 4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PE2	PE1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3A.60 IPIR_CHn Registers Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PE2	Inter-PE Interrupt Request to PE2 Writing 1 to this bit enables an interrupt request to PE2. This bit is automatically cleared to 0 when the interrupt request has been notified. 0: Inter-PE interrupt request output is not specified or an interrupt request is not being output. 1: Interrupt request output is specified or an interrupt request is being output.
0	PE1	Inter-PE Interrupt Request to PE1 Writing 1 to this bit enables an interrupt request to PE1. This bit is automatically cleared to 0 when the interrupt request has been notified. 0: Inter-PE interrupt request output is not specified or an interrupt request is not being output. 1: Interrupt request output is specified or an interrupt request is being output.

3A.2.4 Reliability Functions

3A.2.4.1 PE Guard Function (PEG)

(1) Overview of the PEG Function

The PEG is a constituent of the slave guard system to prevent unauthorized access to the resources in the CPU (PE) from an external master. This function protects access to the local RAM in the PE. In the initial state after a reset, access by masters other than own PE is disabled. Setting the registers listed in (3) List of PEG Protection Setting Registers enables access by masters other than own PE.

(1) Detecting PE guard violation
If an external master makes an unauthorized access to the resource area in a PE for which PE guard is set, the access is detected as a PE guard violation.

(2) Blocking unauthorized access
When a PE guard violation is detected, unauthorized access to the internal resources of the PE are blocked to prevent unauthorized modification of the contents of PE resources.

(3) Notifying occurrence of violation
An error response to an unauthorized access is sent to the request source of external master. When DMA Controller makes an unauthorized access, meanwhile, a DMA transfer error is detected.
A PE guard violation is notified as INTGUARD interrupt request which is a source of FEINT.

(2) Protection Made by SPID

- Setting PEG Protection
  - Up to four areas can be set depending on the local RAM address of the own PE.
  - The area range is specified by the base address and the mask bit (4 kbytes to 4 Gbytes).
  - “Read enable” and “write enable” can be set for each area.
  - “Enable” or “disable” can be selected based on the system protection identifier (SPID) for each area.
- Procedure for permitting access by using the system protection identifier (SPID)

1. Is the area subject to access is the local RAM area? If so, go to step 2.
2. Is the area subject to access is within the range of valid areas 0, 1, 2, or 3? If so, go to step 3.
   Otherwise, return an error response.
3. Are all the conditions below for the relevant area satisfied? If so, permit access.
   - The system protection identifier (SPID) is enabled.
   - Required operations (read/write) are enabled.

   Otherwise, return an error response.

(3) List of PEG Protection Setting Registers

Specify the necessary settings in the registers below to protect PE resources from unauthorized access by an external master.

- Whether to permit access to the local RAM in the PE can be specified.

Table 3A.61 PEG Registers (Base Address: FFFE E600$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after Reset
							1	8	16	32	
PEG	+00C$_H$	4	PEG SPID control register	PEGSP	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+080$_H$	4	PEG area 0 mask setting register	PEGG0MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+084$_H$	4	PEG area 0 base setting register	PEGG0BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+090$_H$	4	PEG area 1 mask setting register	PEGG1MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+094$_H$	4	PEG area 1 base setting register	PEGG1BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0A0$_H$	4	PEG area 2 mask setting register	PEGG2MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0A4$_H$	4	PEG area 2 base setting register	PEGG2BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0B0$_H$	4	PEG area 3 mask setting register	PEGG3MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0B4$_H$	4	PEG area 3 base setting register	PEGG3BA	—	R/W	—	✓	✓	✓	0000 0000$_H$

(4) Register Set

(a) PEGSP — PEG SPID Control Register

Access: PEGSP register can be read or written in 32-bit units.
PEGSPL register can be read or written in 16-bit units.
PEGSPLL register can be read or written in 8-bit units.

Address: PEGSP: FFFE E60C$_H$
PEGSPL: FFFE E60C$_H$
PEGSPLL: FFFE E60C$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3A.62 PEGSP Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SPEN	Access permission to external master with specified SPID. 0: Not permit. 1: Permit.

(b) PEGGnMK — PEG Area n Mask Setting Register (n = 0 to 3)

The PEGGnMK register defines which bits of PEGGnBA.GnBASE are compared with the access address. If bit PEGGnMK.GnMASK[m] is cleared, bit PEGGnBA.GnBASE[m] is compared with bit m of the access address.

Access: PEGGnMK register can be read or written in 32-bit units.
PEGGnMKL, PEGGnMKH registers can be read or written in 16-bit units.
PEGGnMKLH, PEGGnMKHL, PEGGnMKHH registers can be read or written in 8-bit units.

Address: PEGGnMK: FFFE $E680_H$ + (10_H × n)
PEGGnMKL: FFFE $E680_H$ + (10_H × n),
PEGGnMKH: FFFE $E682_H$ + (10_H × n)
PEGGnMKLH: FFFE $E681_H$ + (10_H × n),
PEGGnMKHL: FFFE $E682_H$ + (10_H × n),
PEGGnMKHH: FFFE $E683_H$ + (10_H × n)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnMASK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnMASK				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 3A.63 PEGGnMK Register Contents

Bit Position	Bit Name	Function
31 to 12	GnMASK	0: Target bits are compared when determining the PE guard area. 1: Target bits are not compared when determining the PE guard area.
11 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

NOTE

When you write to the PEGGnMK register, the corresponding GnEN bit in the PEGGnBA register is cleared automatically.

(c) PEGGnBA — PEG Area n Base Setting Register (n = 0 to 3)

In combination with the PEGGnMK register, this register specifies a range or ranges within PE guard protection area n. Setting the GnEN bit to 1 validates the access enable conditions specified by this register and the PEGGnMK register.

Access: PEGGnBA register can be read or written in 32-bit units.

PEGGnBAL, PEGGnBAH registers can be read or written in 16-bit units.

PEGGnBALL, PEGGnBALH, PEGGnBAHL, PEGGnBAHH registers can be read or written in 8-bit units.

Address: PEGGnBA: FFFE E684$_H$ + (10$_H$ × n)

PEGGnBAL: FFFE E684$_H$ + (10$_H$ × n),

PEGGnBAH: FFFE E686$_H$ + (10$_H$ × n)

PEGGnBALL FFFE E684$_H$ + (10$_H$ × n),

PEGGnBALH FFFE E685$_H$ + (10$_H$ × n),

PEGGnBAHL: FFFE E686$_H$ + (10$_H$ × n),

PEGGnBAHH: FFFE E687$_H$ + (10$_H$ × n)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnBASE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnBASE				—	—	—	—	GnSP3	GnSP2	GnSP1	GnSP0	—	GnWR	GnRD	GnEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 3A.64 PEGGnBA Register Contents

Bit Position	Bit Name	Function
31 to 12	GnBASE	Base address that specifies the range of PE guard protection area n.
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	GnSP3	Access permission setting from SPID = 3 external master to PE guard protection area n. 0: Not permit. 1: Permit.
6	GnSP2	Access permission setting from SPID = 2 (CPU2*1) external master to PE guard protection area n. 0: Not permit. 1: Permit.
5	GnSP1	Access permission setting from SPID = 1 (CPU1*1) external master to PE guard protection area n. 0: Not permit. 1: Permit.
4	GnSP0	Access permission setting from SPID = 0 (peripheral device connected to H-BUS) external master to PE guard protection area n. 0: Not permit. 1: Permit.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	GnWR	Write access permission to PE guard protection area n. 0: Write access is disabled. 1: Write access is enabled.

Table 3A.64 PEGGnBA Register Contents

Bit Position	Bit Name	Function
1	GnRD	Read access permission to PE guard protection area n. 0: Read access is disabled. 1: Read access is enabled.
0	GnEN	Enables or Disables the setting for the access enable conditions to PE guard protection area n. 0: Settings for access enable conditions are disabled. 1: Settings for access enable conditions are enabled.

Note 1. Setting value of MCFG0.SPID

NOTE

When you write to the PEGGnMK register, the corresponding GnEN bit in the PEGGnBA register is cleared automatically.

3A.2.4.2 PE’s Internal Peripheral Device Protection Function (IPG)

(1) Overview of the IPG Function

The IPG is a function to prevent unauthorized accesses to peripheral devices from the CPU core equipped with the IPG. The IPG achieves the following functions. The IPG covers accesses to the SEG, the PEG, the IPIR, the MEV, the INTC1 and P-Bus.

(a) Detecting Violation of Peripheral Device Protection

If the CPU makes an unauthorized access to an area (peripheral device) for which peripheral device protection is set, the access is detected as “violation of peripheral device protection”.

(b) Storing Unauthorized Access Information

When a violation of peripheral device protection is detected, the unauthorized-access information is stored in the IPG’s internal register.

(c) Blocking Unauthorized Accesses

When a violation of peripheral device protection is detected, unauthorized accesses to peripheral devices are blocked to prevent contents of peripheral devices from being modified illegally.

(d) Notifying Violation

When a violation of peripheral device protection is detected, a request for generating an exception is made to ask the CPU to stop the processing.

NOTE

Even if a request for generating an exception is immediately sent to the CPU in step **(d) Notifying Violation** above, a subsequent access issued (before receiving a request from the IPG) by the CPU that does not know an occurrence of violation may illegally modify contents of peripheral devices. (Accesses after a violation has occurred result in unauthorized accesses.)

(2) IPG Function

- This function invalidates accesses according to their attributes (including address, transfer type, and access right).
- After an access right violation is detected until the error flag (described later) is cleared by writing by the software, subsequent accesses are invalidated. However, invalidation is applied only to accesses from the CPU and is not applied to accesses from outside the CPU core. Invalidation is performed independently of addresses.
- When a request for accessing different peripheral devices simultaneously is made due to misalignment or double-word access, the access is executed when all such accesses are enabled.

(3) IPG Protection Setting Registers for Illegal Users

To protect peripheral devices from unauthorized accesses by programs in user mode, necessary settings are required for the registers listed below.

- Accesses in user mode are to be detected.

Table 3A.65 IPG Registers (Base Address: FFFE $E000_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right*1	R/W	Operable Bit				Value after Reset
							1	8	16	32	
IPG	$+002_H$	2	Peripheral device protection violation access information register	IPGECRUM	SV	R/W	—	✓	✓	—	Undefined (retained)
	$+008_H$	4	Peripheral device protection violation access address register	IPGADRUM	SV	R/W	—	✓	✓	✓	Undefined (retained)
	$+00D_H$	1	Peripheral device protection enable register	IPGENUM	SV	R/W	—	✓	—	—	00_H
	$+020_H$	1	Peripheral device protection setting register 0	IPGPMTUM0	SV	R/W	—	✓	—	—	00_H
	$+022_H$	1	Peripheral device protection setting register 2	IPGPMTUM2	SV	R/W	—	✓	—	—	00_H
	$+023_H$	1	Peripheral device protection setting register 3	IPGPMTUM3	SV	R/W	—	✓	—	—	00_H
	$+024_H$	1	Peripheral device protection setting register 4	IPGPMTUM4	SV	R/W	—	✓	—	—	00_H

Note 1. Registers for which “SV” is described are accessible by accesses with SV right (UM = 0).

(4) Register Set

(a) IPGECRUM — Peripheral Device Protection Violation Access Information Register

Access: IPGECRUM register can be read or written in 16-bit units.
IPGECRUML register can be read or written in 8-bit units.

Address: IPGECRUM: FFFE E002$_{\text{H}}$
IPGECRUML: FFFE E002$_{\text{H}}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WD	HW	BY	EX	WR	RD	VD
Value after reset	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3A.66 IPGECRUM Register Contents

Bit Position	Bit Name	Function
15 to 7	Reserved	These bits are always read as 0. The write value should always be 0.
6	WD	This bit is set to 1 when a violation occurred in read word, instruction fetch read access, write word, CAXI, LDL or STC. In other cases, this bit is cleared to 0.
5	HW	This bit is set to 1 when a violation occurred in read halfword or write haflword. In other cases, this bit is cleared to 0.
4	BY	This bit is set to 1 when a violation occurred in read byte, write byte or bit operation. In other cases, this bit is cleared to 0.
3	EX	This bit is set to 1 when a violation occurred in an instruction fetch read access. In other cases, this bit is cleared to 0.
2	WR	This bit is set to 1 when a violation occurred in a write access, bit operation, or execution of the CAXI instruction. In other cases, this bit is cleared to 0.
1	RD	This bit is set to 1 when a violation occurred in a read access, bit operation, or execution of the CAXI instruction. In other cases, this bit is cleared to 0.
0	VD	This bit is set to 1 when a violation of peripheral device protection is detected by a program with the relevant right. Even if another violation of peripheral device protection is detected while this bit is 1, data of this IPGECRUM register and the IPGADRUM register is not updated and is retained.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(b) IPGADRUM — Peripheral Device Protection Violation Access Address Register

Access: IPGADRUM register can be read or written in 32-bit units.

IPGADRUML, IPGADRUMH registers can be read or written in 16-bit units.

IPGADRUMLL, IPGADRUMLH, IPGADRUMHL, IPGADRUMHH registers can be read or written in 8-bit units.

Address: IPGADRUM: FFFE E008$_H$

IPGADRUML: FFFE E008$_H$, IPGADRUMH: FFFE E00A$_H$

IPGADRUMLL: FFFE E008$_H$, IPGADRUMLH: FFFE E009$_H$,

IPGADRUMHL: FFFE E00A$_H$, IPGADRUMHH: FFFE E00B$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EADR[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3A.67 IPGADRUM Register Contents

Bit Position	Bit Name	Function
31 to 0	EADR	These bits store the address of the access in which a violation occurred.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(c) IPGENUM — Peripheral Device Protection Enable Register

Access: IPGENUM register can be read or written in 8-bit units.

Address: IPGENUM: FFFE E00D$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IRE	E
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3A.68 IPGENUM Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	IRE	This bit sets whether to store the access information in the peripheral device protection violation access address register and the peripheral device protection violation access information register when a violation of peripheral device protection occurred in an instruction fetch access. 0: Instruction fetch access information is not stored. (value after reset) 1: Instruction fetch access information is stored. **CAUTION:** If you do not want to detect speculative instruction fetches (no instruction is executed in some cases), clear this bit to 0.
0	E	This bit enables or disables the peripheral devices protection function against accesses by the relevant access right. 0: The peripheral device protection function is disabled. (Value after reset) 1: The peripheral device protection function is enabled.

(d) IPGPMTUM0 — Peripheral Device Protection Setting Register 0

Access: IPGPMTUM0 register can be read or written in 8-bit units.

Address: IPGPMTUM0: FFFE E020$_H$

Bit	7	6	5	4	3	2	1	0
	—	X1	W1	R1	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R

Table 3A.69 IPGPMTUM0 Register Contents

Bit Position	Bit Name	Function
7	Reserved	These bits are always read as 0. The write value should always be 0.
6	X1	This bit sets whether to enable instruction fetch read access to P-Bus. 0: Instruction fetch read access to P-Bus is treated as violation. (Value after reset) 1: Instruction fetch read access to P-Bus is not restricted.
5	W1	This bit sets whether to enable write access to P-Bus. 0: Write access to P-Bus is treated as violation. (Value after reset) 1: Write access to P-Bus is not restricted.
4	R1	This bit sets whether to enable read access to P-Bus. 0: Read access to P-Bus is treated as violation. (Value after reset) 1: Read access to P-Bus is not restricted.
3 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

(e) IPGPMTUM2 — Peripheral Device Protection Setting Register 2

Access: IPGPMTUM2 register can be read or written in 8-bit units.

Address: IPGPMTUM2: FFFE E022$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	W1	R1	—	—	W0	R0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 3A.70 IPGPMTUM2 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	These bits are always read as 0. The write value should always be 0.
5	W1	This bit sets whether to enable write access to G0MEVm/IPIR_CHn. 0: Write access to G0MEVm/IPIR_CHn is treated as violation. (Value after reset) 1: Write access to G0MEVm/IPIR_CHn is not restricted.
4	R1	This bit sets whether to enable read access to G0MEVm/IPIR_CHn. 0: Read access to G0MEVm/IPIR_CHn is treated as violation. (Value after reset) 1: Read access to G0MEVm/IPIR_CHn is not restricted.
3, 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	W0	This bit sets whether to enable write access to INTC1. 0: Write access to INTC1 is treated as violation. (Value after reset) 1: Write access to INTC1 is not restricted
0	R0	This bit sets whether to enable read access to INTC1. 0: Read access to INTC1 is treated as violation. (Value after reset) 1: Read access to INTC1 is not restricted.

(f) IPGPMTUM3 — Peripheral Device Protection Setting Register 3

Access: IPGPMTUM3 register can be read or written in 8-bit units.

Address: IPGPMTUM3: FFFE E023$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	W1	R1	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 3A.71 IPGPMTUM3 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	These bits are always read as 0. The write value should always be 0.
5	W1	This bit sets whether to enable write access to SysErrGen. 0: Write access to SysErrGen is treated as violation. (Value after reset) 1: Write access to SysErrGen is not restricted.
4	R1	This bit sets whether to enable read access to SysErrGen. 0: Read access to SysErrGen is treated as violation. (Value after reset) 1: Read access to SysErrGen is not restricted
3 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

(g) IPGPMTUM4 — Peripheral Device Protection Setting Register 4

Access: IPGPMTUM4 register can be read or written in 8-bit units.

Address: IPGPMTUM4: FFFE E024$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	W0	R0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3A.72 IPGPMTUM4 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	W0	This bit sets whether to enable write access to its own PEG. 0: Write access to its own PEG is treated as violation. (Value after reset) 1: Write access to its own PEG is not restricted.
0	R0	This bit sets whether to enable read access to its own PEG. 0: Read access to its own PEG is treated as violation. (Value after reset) 1: Read access to its own PEG is not restricted

3A.2.4.3 System Error Generator Function (SEG)

SEG (SysErrGen) controls interrupt notification and recording after a system error occurred by a data access.

Multiple error occurrence inputs are categorized according to error factor, and are processed sequentially from the highest-priority error factor, generating an FE-level exception (SYSERR).

The bit position of the SEGFLAG register becomes the error factor priority. Error factors of lower bits take precedence over error factors of upper bits.

Error address information is recorded only once regardless of error frequency.

The error with the highest priority among the error factors is valid when errors occur simultaneously. Recorded error address information is not overwritten by subsequent errors.

(1) List of SEG Function Control Registers

Table 3A.73 SEG Register (Base Address: FFFE E980$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right*1	R/W	Operable Bit				Value after Reset
							1	8	16	32	
SEG	+00$_H$	2	SEG error control register	SEGCONT	SV	R/W	—	—	✓	—	0000$_H$
	+02$_H$	2	SEG error flag register	SEGFLAG	SV	R/W	—	—	✓	—	0000$_H$
	+08$_H$	4	SEG error address information register	SEGADDR	SV	R/W	—	—	—	✓	Undefined (retained)

Note 1. Registers for which “SV” is described are writable with the SV right (UM = 0). Attempting to write, if these conditions do not hold, leads to a SYSERR exception with setting VCIF flag. No restriction is provided for read accesses.

(2) Register Set

(a) SEGCONT — SEG Error Control Register

This register is used to enable (= 1) or disable (= 0) notification of SysErr request in response to error flags that store the error occurrence status for each factor.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VCSE	APIE	IPGE	—	TCME	—	VCIE	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R	R	R	R

Table 3A.74 SEGCONT Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	VCSE	This bit enables notification of an error response detected inside system interconnect: ● Error response from external bus masters in write access ● Illegal response to local RAM or peripherals or Flash from optional master in access (except instruction fetch from CPU). And illegal response to optional slave from external AHB master in access.
9	APIE	This bit enables notification of an error response from peripherals. The error notification includes the following cases: ● Error response from peripherals in write access ● PBG error in write access
8	IPGE	This bit enables notification of IPG illegal access detection.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCME	This bit enables notification of an error during data access to its own local RAM from PE master. The error notification includes the following cases: ● ECC uncorrectable error (DED or SED & SECDIS=1) ● Detection of an access to RAM unimplemented area
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	VCIE	This bit enables notification of an error response during access to CPU peripherals, P-Bus (read access), global RAM, retention RAM, other PE local RAM, H-Bus and CodeFlash by PE. The error notification includes the following cases: ● IPG error from CPU peripherals and P-Bus ● Error response from H-Bus peripherals ● PBG error / HBG error from P-Bus and H-Bus ● GRG error from global RAM and retention RAM ● PEG error from other PE local RAM ● ECC uncorrectable error from CodeFlash, global RAM, retention RAM and other PE local RAM (DED or SED & SECDIS = 1) This bit enables notification of an error response when accessing to a part of access prohibited areas in address map.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(b) SEGFLAG — SEG Error Flag Register

This register indicates error flags that store error occurrence status of each factors. The flags are set to 1 by an error occurrence input. The flags are not automatically cleared to 0. Both setting and clearing of each flag are supported in writing to the register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VCSF	APIF	IPGF	—	TCMF	—	VCIF	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R	R	R	R

Table 3A.75 SEGFLAG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	VCSF	Flag corresponding to bit 10 of the SEGCONT register
9	APIF	Flag corresponding to bit 9 of the SEGCONT register
8	IPGF	Flag corresponding to bit 8 of the SEGCONT register
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCMF	Flag corresponding to bit 6 of the SEGCONT register
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	VCIF	Flag corresponding to bit 4 of the SEGCONT register
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

NOTE

An error may lead to setting of multiple error flags in SEG. For example, if an IPG error occurs at peripheral registers read, both IPGF bit and VCIF bit in SEGFLAG are set.

(c) SEGADDR — Error Address Information Register

Address information (one record) which is notified with error occurrence is stored in the register. The register is not updated while one or more bits in SEGFLAG register are set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Address[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Address[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3A.76 SEGADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	Address	These bits store the error address information.

CAUTIONS

1. SEGADDR stores error address information in case of an error occurrence related to VCIF bit or TCMF bit in SEGFLAG register. SEGADDR register stores all 0 data in case of an error occurrence related to VCSF bit, APIF bit or IPGF bit in SEGFLAG register.
2. In case of an error occurrence related to TCMF bit in SEGFLAG register, bit[18:0] of the error address are stored in SEGADDR[18:0] and SEGADDR[31:19] are filled with 0.

(3) SEG Function

(a) SEG Function: SYSERR Request Notification by Error Flag

- Setting an error flag takes precedence over clearing the same flag.
  - A simultaneous clearing operation is ignored.
- Priority of error factors
  - The bit position of each flags in SEGFLAG register which error notification is enabled by SEGCONT register becomes the error factor priority. Error factors of lower bits take precedence over error factors of upper bits. Notification is made from the highest-priority error factor.
  - The bit position of error factors is reported as a “SysErr factor code.”
- Conditions for starting SysErr request notification
  - Even if a flag which error notification is disabled by SEGCONT register is set to 1, notification is not made.
  - Notification is made immediately after a flag which error notification is enabled by SEGCONT register is set to 1.
  - After clearing of a flag, notification is made if an other flag which error notification is enabled by SEGCONT register remains set.
- Finishing notification at a SysErr acknowledgement
  - Even after notification is finished, the flag is not cleared automatically.
  - Notification is not made until setting or clearing the flag again.
  - If an error flag that is prioritized higher than the error factor is set prior to an acknowledgement, the notification information may be replaced with a higher prioritized SysErr factor code.

(b) SEG Function: Recording Error Address Information

- When an error which error notification is enabled by SEGCONT register occurs, the error address is retained in the SEGADDR register.
  - No information is retained by setting or clearing an error flag in SEGFLAG register.
  - When multiple error occurrence inputs are present simultaneously, information other than the prioritized error factor is not retained.
- While a flag which error notification is enabled by SEGCONT register is set to 1, overwrite to the SEGADDR register is inhibited.
  - If error occurrence input continues, information of subsequent error factors is not retained.
  - To reset the inhibition of overwrite to the register, clear either SEGCONT or SEGFLAG register (or both of them).

(c) Supplementary Notes on SYSERR Exception

- Even when a SYSERR exception occurs, the value of the PSW.EBV bit is held, and the base address of the exception handler does not switch.

3A.3 Inter-CPU Functions

3A.3.1 Processor Element Identifier

The PEID, processor element ID number, can be read from the PEID field in the HTCFG0 register. Which CPU core is executing a specific program can be checked by referring to the PEID. The following shows the PEID of this product.

CPU Core	PEID
CPU1 (PE1)	001_B
CPU2 (PE2)	010_B

3A.3.2 Inter-Processor Interrupt Function

Each CPU has the IPIR register as a peripheral function. Setting of the IPIR register enables an EI-level interrupt request from a PE to another PE. For details, see **Section 3A.2.3, Inter-Processor Interrupts**.

3A.3.3 Exclusive Control

The local RAM, global RAM, retention RAM, and exclusive control registers (G0MEVm; m = 0 to 31) are available as a resource for exclusive control. Atomic operation instructions LDL/STC, CAXI, SET1, CLR1, and NOT1 can be executed on the local RAM, global RAM, and retention RAM. CAXI, SET1, CLR1, and NOT1 can be executed on the exclusive control registers (G0MEVm). Note that the LD and ST instructions can also access these resources, but the access is not atomic.

3A.3.3.1 Exclusive Control Register (G0MEVm; m = 0 to 31)

This register supports exclusive control for variable shared between PEs (common resources). (MEV: Mutual Exclusion Variable Register)

- Thirty-two 32-bit G0MEVm registers are included.
- G0MEVm can be accessed in 8-, 16-, or 32-bit units.
- Access from CPU1 (PE1) and CPU2 (PE2) can be made.
- Atomic operation instructions CAXI, SET1, CLR1, and NOT1 can be executed.

CPU1 (PE1) and CPU2 (PE2) each have an independent access path for G0MEVm registers. Therefore, when CPU1 (PE1) and CPU2 (PE2) each access different G0MEVm registers, they do not need to wait for access. When they access the same G0MEVm register, however, waiting for access is required.

Table 3A.77 List of Registers

Module Name	Address*1	Register Name	Symbol	R/W	Access Size				Value after Reset
					1	8	16	32	
MEV	$+00_H$	Exclusive control register 0	G0MEV0	R/W	—	✓	✓	✓	$0000\ 0000_H$
	$+04_H$	Exclusive control register 1	G0MEV1	R/W	—	✓	✓	✓	$0000\ 0000_H$
	$+08_H$	Exclusive control register 2	G0MEV2	R/W	—	✓	✓	✓	$0000\ 0000_H$
	:	:	:	:	:	:	:	:	:
	$+7C_H$	Exclusive control register 31	G0MEV31	R/W	—	✓	✓	✓	$0000\ 0000_H$

Note 1. Base address: FFFE EC00_H

(1) G0MEVm — Exclusive Control Register m (m = 0 to 31)

Access: G0MEVm can be read or written in 32-bit units.

G0MEVmL and G0MEVmH can be read or written in 16-bit units.

G0MEVmLL, G0MEVmLH, G0MEVmHL, G0MEVmHH can be read or written in 8-bit units.

Address: G0MEVm: FFFE EC00$_H$ + m × 4$_H$,

G0MEVmL: FFFE EC00$_H$ + m × 4$_H$,

G0MEVmH: FFFE EC00$_H$ + m × 4$_H$ + 2$_H$,

G0MEVmLL: FFFE EC00$_H$ + m × 4$_H$,

G0MEVmLH: FFFE EC00$_H$ + m × 4$_H$ + 1$_H$,

G0MEVmHL: FFFE EC00$_H$ + m × 4$_H$ + 2$_H$,

G0MEVmHH: FFFE EC00$_H$ + m × 4$_H$ + 3$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	G0MEVm[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	G0MEVm[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3A.78 G0MEVm Registers Contents

Bit Position	Bit Name	Function
31 to 0	G0MEVm[31:0]	These bits set the value for exclusive control.

3A.3.3.2 Operation of the LDL.W and STC.W Instructions

The LDL.W and STC.W instructions can be used to perform atomic read-modify-write operations for accurate processing in the updating of memory in multicore systems. The LDL.W and STC.W instructions operate as follows.

- Link generation: The CPU is capable of generating links to both the local RAM and global RAM (including the retention RAM). Executing the LDL.W instruction on the target RAM for the operation leads to the link flag being set and a link being generated in response to reading by the instruction. Two link flag systems are provided, one each for the following two areas of RAM.
  (1) The local RAM for the given processor
  (2) The global RAM (including the retention RAM)
- Success in storing: After a link has been generated, storing will only proceed in response to executing an STC.W instruction corresponding to the generated link.
- Failure in storing: If a link is lost, storing does not proceed even when an STC.W instruction for the corresponding address is processed. Storing also does not proceed when an STC.W instruction that does not correspond to the link is processed.
- Condition for successful storing: If the following condition is met, the STC.W instruction is judged to be for the address corresponding to the link.
  - The address for the LDL.W instruction which generated the link matches that for the STC.W instruction.
- Loss of the link: If any of the following occurs, the link flag is cleared and the link is lost.
  - Any of the following processing by the CPU for which the link was generated:
    - Execution of a STC.W instruction. The corresponding link (for (1) or (2) above) will be lost irrespective of the success or failure of storing.
    - Occurrence of an exception or the CPU executing an instruction to return from an exception processing routine (FERET or EIRET). The link flags for both areas of RAM are cleared.
    - Successive execution of LDL.W instructions for a location with the same link flags. The link generated in
    - response to the preceding LDL.W instruction will be lost. Do not execute such processing
    - Execution of a storing operation other than an STC.W instruction for the address generating the link. Do not execute such processing.
  - Access as described below by another bus master:
    Any storing operation, including execution of an STC.W instruction for the address generating the link.
    The corresponding link will be lost.

Success of the STC.W instruction means that the LDL.W and STC.W instructions have realized an atomic read-modify-write operation.

3A.4 CPU2 Boot Up Operation

After reset is released, CPU1 starts to fetch instruction. Initial condition of CPU2 is decided by option byte setting and its boot up condition can be set and confirmed by BOOTCTRL register.

(1) List of Register

Table 3A.79 BOOTCTRL Register (Base Address: FFC5 8000$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after Reset
							1	8	16	32	
—	+00$_H$	4	BOOTCTRL register	BOOTCTRL	—	R/W	—	—	—	✓	0000 0006$_H$ or 0000 0002$_H$

(2) BOOTCTRL — Boot Control Register

This register is used to control the start-up of CPU2. By setting bit to “1”, CPU2 starts its pipeline operation. When the value of CPU2EN is “1”, it cannot be overwritten to “0” excluding reset.

Access: BOOTCTRL register can be read or written in 32-bit units.

Address: FFC5 8000$_H$

Value after reset: The value after reset depends on the option byte setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU2 EN	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1[*1]	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Note 1. The setting of the option byte OPBT0.PE2BOOTEN is reflected. For details on the option byte, see **Section 44.9, Option Bytes**.

Table 3A.80 BOOTCTRL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CPU2EN	CPU2 operation setting 0: Invalid boot up of CPU2. 1: Valid boot up of CPU2. When the value of PE2EN (option byte) is “0”, it cannot be overwritten to “1”.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

3A.5 Notes

3A.5.1 Synchronization of Store Instruction Completion and Subsequent Instruction Execution

When a control register is updated by a store instruction, there is a time lag after the CPU executes the store instruction and before the control register is actually updated. Therefore, if the updated content of the control register is to be used by the instruction following the store instruction, the appropriate synchronization is required. How to perform synchronization processing is shown below.

For the procedures to synchronize updating system registers by LDSR instruction and the subsequent instruction execution, see *APPENDIX A, Hazard Resolution Procedure for System Registers* in *the RH850G3KH User's Manual: Software*.

When the updated results in the control registers are to be used by the subsequent instruction:

Example 1: An interrupt is enabled by execution of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits.

Proceed as follows in this case.

(1) Execute the store instruction to update a control register (ST.W, etc.).

(2) Perform a dummy read of the above control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute the subsequent instruction (EI).

In case of RH850/F1KH, SYNCM instruction has the same effects as above-mentioned (2) and (3). (Excludes RCFDCn, RCFDCn ECC register access.)

Example 2: When you must wait until a control register (control register A) has been completely updated before accessing another control register (control register B), execute similar processing. For example, different peripheral functions are linked, or the interrupt mask for INTC is cleared after the peripheral function is set. Note that this processing is not required if the control registers A and B belong to the same peripheral group.

(1) Execute the store instruction to update the control register A (ST.W, etc.).

(2) Perform a dummy read of the above control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute the store instruction to access the control register B (ST.W, LD.W, etc.).

In case of RH850/F1KH, SYNCM instruction has the same effects as above-mentioned (2) and (3). (Excludes RCFDCn, RCFDCn ECC register access.)

The similar processing is also required when starting to access a memory or control register to be protected is started after a safety function (such as some kind of memory protection and ECC) has been completely set up.

When the updated results of the control register or memory to be used in the instruction fetch of the subsequent instruction:

(a) In case of writing the instructions to the RAM before jumping to the RAM to execute instructions from the RAM, take the following procedure.

(1) Execute the store instruction to update a memory (ST.W, etc.).

(2) Perform a dummy read of the above memory (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute SYNCI.

(5) Execute the subsequent instruction (branch instruction, etc.).

(b) In case of updating control registers for memory protection and ECC functions before jumping to the memory to be controlled by the registers, take the following procedure.

(1) Execute the store instruction to update a control register (ST.W, etc.).

(2) Perform a dummy read of the control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute SYNCI.

(5) Execute the subsequent instruction (branch instruction, etc.).

When switching the code flash memory area:

In this case, see *Section 10, Usage Notes, (7) Updating the BFASELR register* in *the RH850/F1KH, F1KM, F1K Flash Memory User’s Manual: Hardware Interface*.

3A.5.2 Ensure Coherency after Rewriting the Code Flash

The CPU1 is equipped with the buffer for the code flash area as described in **Section 3A.2.2, Buffers for Code Flash**.

Therefore, clear the buffer to ensure coherency after rewriting the code flash by self-programming.

3A.5.3 Access to Registers by Using Bit-Manipulation Instructions

Writing bit-manipulation instructions consists of read-modify-write processing in 8-bit units. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. If a register includes multiple flag bits, the read-modify-write operation may lead to the clearing of flags that were not actually targets for clearing.

3A.5.4 Caution of Prefetching

There is a possibility that the reading of the memory occurs by pre-fetch from the area where instruction codes do not exist. Secure more than 40-byte initialized area after the area where instruction codes are stored.

3A.5.5 Overwriting Context upon Acceptance of Multiple Exceptions

Acceptance of an exception depends on the type of exception source, regardless of the states of the ID and NP bits in the PSW register. When multiple exceptions are generated, the contents of the system register which hold the context information are overwritten. For the conditions for acceptance and whether correct return or recovery is possible for each exception source, see *the List of Exception Sources* in *the RH850G3KH User’s Manual: Software*.

Section 3BC CPU System of RH850/F1KM

3BC.1 Overview

3BC.1.1 Block Configuration

Figure 3BC.1, Block Configuration Diagram of the RH850/F1KM-S4 shows the block configuration diagram of RH850/F1KM-S4.

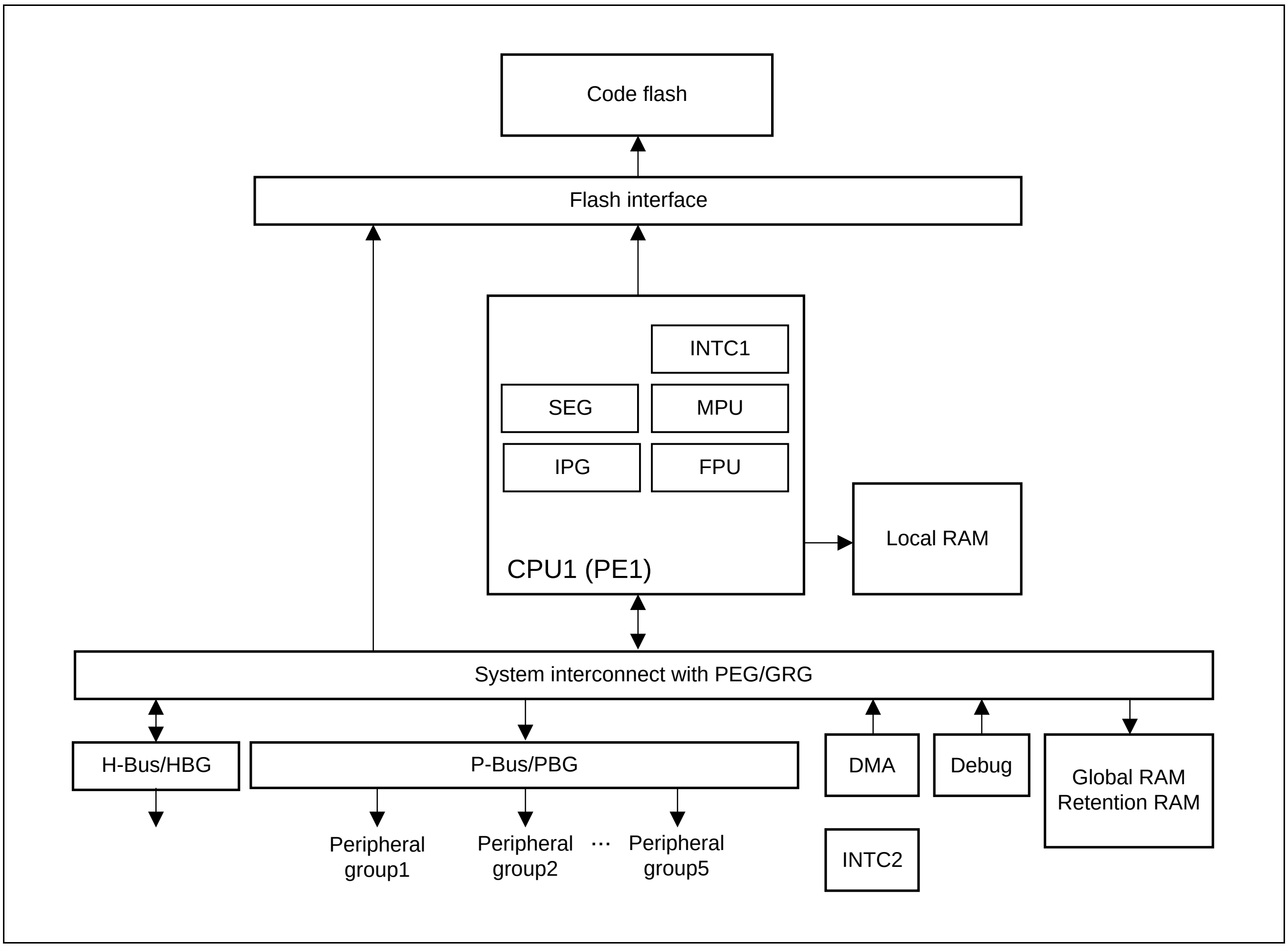


Figure 3BC.1 Block Configuration Diagram of the RH850/F1KM-S4

Figure 3BC.2, Block Configuration Diagram of the RH850/F1KM-S1 shows the block configuration diagram of RH850/F1KM-S1.

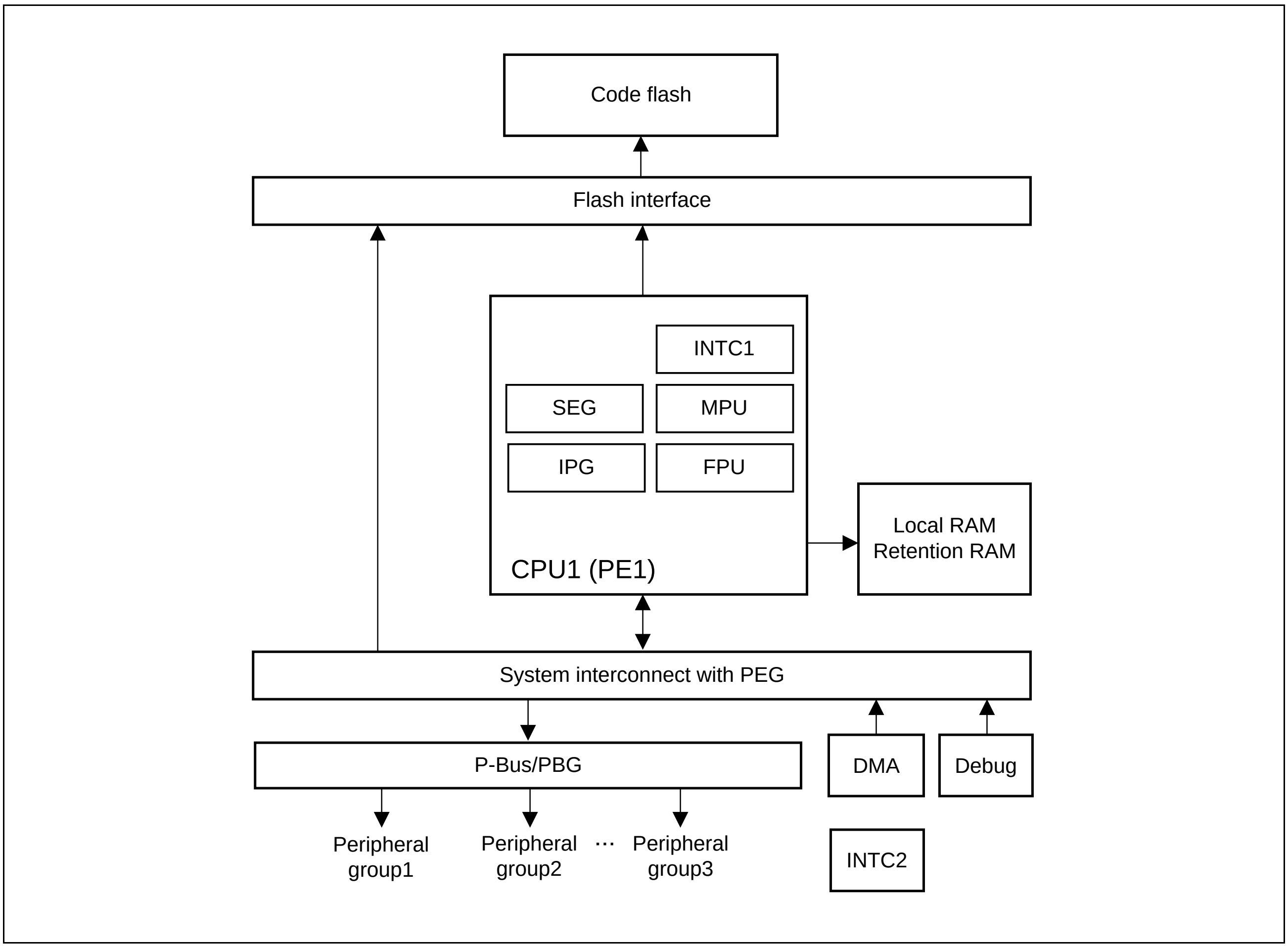


Figure 3BC.2 Block Configuration Diagram of the RH850/F1KM-S1

CPU1 (PE1)

The RH850G3KH2.0 Core is used as the main CPU.

Local RAM

This is a high-speed accessible RAM

Global RAM

The global RAM is a large-capacity RAM for data sharing among CPU and with DMA.

This is not supported with RH850/F1KM-S1.

Retention RAM

RH850/F1KM-S4:
The retention RAM is used to retain values in DeepSTOP mode. Since the continuous global RAM area is assigned for the retention RAM, the retention RAM can also serve as a global RAM for sharing data with the DMA.

RH850/F1KM-S1:
The retention RAM is used to retain values in DeepSTOP mode. Since the continuous local RAM area is assigned for the retention RAM, the retention RAM can also serve as a local RAM for sharing data with the DMA.

Code flash

The code flash memory is included for program storage. It is connected with CPU1 via the flash interface.

Data flash

The data flash memory can be rewritten by the CPU1. It has a greater write endurance than the code flash memory.

P-Bus and H-Bus (RH850/F1KM-S4)

The P-Bus connects the peripheral IPs. The P-Bus is divided into five peripheral groups, 1 to 5.

P-Bus (RH850/F1KM-S1)

The P-Bus connects the peripheral IPs. The P-Bus is divided into three peripheral groups, 1 to 3.

INTC1, INTC2

There are two interrupt controllers, INTC1 and INTC2.

DMA

The DMA transfer module (PDMA) is included.

Slave guard

The slave guard is a function to prevent unauthorized access from the specific bus master, and consists of the following guard structures:

(1) PE guard (PEG)
The PE guard is a function to prevent unauthorized access to the resources (local RAM) in the PE from an external master. After reset is released, access from other than the own PE is prohibited.

(2) Internal Peripheral Guard (IPG)
The PE with system interconnects supports “Internal Peripheral Guard” (IPG) that protects the registers of peripherals against invalid accesses.

(3) Global RAM guard (GRG) (RH850/F1KM-S4)
The global RAM guard is a function to prevent unauthorized access to the global RAM and retention RAM from an external master. The global RAM is in the unprotected state (accessible from all bus master) after reset is released. For details, see **Section 40B, Functional Safety of RH850/F1KM-S4**.

(4) Peripheral guard (PBG / HBG (RH850/F1KM-S4))
The peripheral guard is a function to prevent unauthorized access to peripherals. The control registers in the peripheral circuits are protected against illegal accesses.
For details, see **Section 40B, Functional Safety of RH850/F1KM-S4** and **Section 40C, Functional Safety of RH850/F1KM-S1**.

3BC.2 CPU

3BC.2.1 Core Functions

3BC.2.1.1 Features

Table 3BC.1, Features of the RH850G3KH2.0 Core lists features of the RH850G3KH2.0 core.

Table 3BC.1 Features of the RH850G3KH2.0 Core

Item	Feature
CPU	● Advanced 32-bit architecture for embedded control ● 32-bit internal data bus ● Thirty-two 32-bit general-purpose registers – RISC-type instruction set – Long-/short-format load/store instructions – Three-operand instructions – Instruction set based on C language ● CPU operating modes – User mode and supervisor mode ● Address space: 4-Gbyte linear address space for both data and instructions
Coprocessor	● Floating-point operation coprocessor (FPU) – Supports single precision (32 bits) – Supports data types and exceptions conforming to IEEE754. – Rounding mode: Neighborhood, 0 direction, +∞ direction, and −∞ direction – Handling of denormalized numbers: Rounding down to 0 or exception notification to conform to IEEE754
Exception/Interrupt	● 16 interrupt priority levels settable for each channel ● Vector selection method selectable according to performance request or memory usage – Direct branching exception vectors – Indirect branching exception vectors referring to the address table ● Supports the high-speed save/return processing of the context by the dedicated instructions (PUSHSP and POPSP) at the generation of an interrupt
Memory management	● Memory protection function (MPU): 16 areas settable
Cache	● No cache memory is equipped.

3BC.2.1.2 Register Set

This subsection explains the program registers and system registers incorporated in this CPU.

(1) Program Registers

Program registers include the general-purpose registers (r0 to r31) and program counter (PC).

Table 3BC.2 Program Registers

Program Register	Name	Function	Description
General-purpose registers	r0	Zero register	Always retains “0”
	r1	Assembler reserved register	Used as working register for generating addresses
	r2	Register for address and data variables (used when the real-time OS used does not use this register)	
	r3	Stack pointer (SP)	Used for generating a stack frame when a function is called
	r4	Global pointer (GP)	Used for accessing a global variable in the data area
	r5	Text pointer (TP)	Used as a register that indicates the start of the text area (area where program code is placed)
	r6 to r29	Register for address and data variables	
	r30	Element pointer (EP)	Used as a base pointer for generating addresses when accessing memory
	r31	Link pointer (LP)	Used when the compiler calls a function
Program counter	PC	Retains instruction addresses during execution of programs	

NOTE

For further descriptions of r1, r3 to r5, and r31 used by the assembler and/or C compiler, see the specification of each software development environment.

(a) General-Purpose Registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables. Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

1. r0, r3, r30

These registers are implicitly used by instructions.
r0 is a register that always retains “0”. It is used for operations that use 0 and addressing with base address being 0.
r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.
r30 is used as a base pointer when the SLD or SST instruction accesses memory.

2. r1, r4, r5, r31

These registers are implicitly used by the assembler and C compiler.
When using these registers, register contents must first be saved so they are not lost and can be restored after the registers are used.

3. r2

This register might be used by a real-time OS in some cases. If the real-time OS that is being used does not use r2, r2 can be used as a register for address variables or data variables.

(b) PC — Program Counter

The PC retains the address of the instruction being executed. Bit 0 is fixed to 0, and branching to an odd number address is disabled.

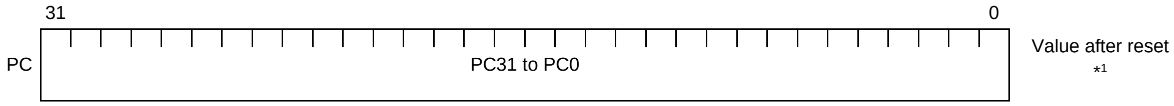


Table 3BC.3 PC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	PC31 to PC1	These bits indicate the address of the instruction being executed.	R/W	*1
0	PC0	This bit is fixed to 0. Branching to an odd number address is disabled.	R/W	0

Note 1. The value after reset differs depending on the setting value of the reset vector. For details, see **(q) RBASE — Reset Vector Base Address Register**.

(2) Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

System registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and a selection ID.

Table 3BC.4 Basic System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 0	EIPC	Status save registers when acknowledging EI level exception	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	SV
SR2, 0	FEPC	Status save registers when acknowledging FE level exception	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	SV
SR5, 0	PSW	Program status word	*1
SR6, 0	FPSR	(Refer to FPU function registers.)	CU and SV
SR7, 0	FPEPC	(Refer to FPU function registers.)	CU and SV
SR8, 0	FPST	(Refer to FPU function registers.)	CU
SR9, 0	FPCC	(Refer to FPU function registers.)	CU
SR10, 0	FPCFG	(Refer to FPU function registers.)	CU
SR11, 0	FPEC	(Refer to FPU function registers.)	CU and SV
SR13, 0	EIIC	EI level exception source register	SV
SR14, 0	FEIC	FE level exception source register	SV
SR16, 0	CTPC	CALLT execution status save register	UM
SR17, 0	CTPSW	CALLT execution status save register	UM
SR20, 0	CTBP	CALLT base pointer register	UM
SR28, 0	EIWR	EI level exception working register	SV
SR29, 0	FEWR	FE level exception working register	SV
SR0, 1	MCFG0	Machine configuration register	SV
SR2, 1	RBASE	Reset vector base address register	SV
SR3, 1	EBASE	Exception handler vector address register	SV
SR4, 1	INTBP	Base address register of the interrupt handler address table	SV
SR5, 1	MCTL	CPU control	SV
SR6, 1	PID	Processor ID register	SV
SR11, 1	SCCFG	SYSCALL operation setting register	SV
SR12, 1	SCBP	SYSCALL base pointer register	SV
SR0, 2	HTCFG0	Thread configuration register	SV
SR6, 2	MEA	Memory error address register	SV
SR7, 2	ASID	Address space ID register	SV
SR8, 2	MEI	Memory error information register	SV

Note 1. The access permission differs depending on the bit.

(a) EIPC — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see “*Types of Exceptions*” in *Software Manual*).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address cannot be specified.

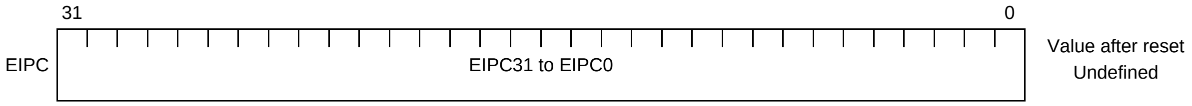


Table 3BC.5 EIPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	EIPC31 to EIPC1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC saved when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

(b) EIPSW — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

	31	30	29												17	16	15	14						8	7	6	5	4	3	2	1	0	
EIPSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z	Value after reset 0000 0020$_H$

Table 3BC.6 EIPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit stores the PSW.CU field setting when an EI level exception is acknowledged.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

(c) FEPC — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see “*Types of Exceptions*” in *Software Manual*).

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address cannot be specified.

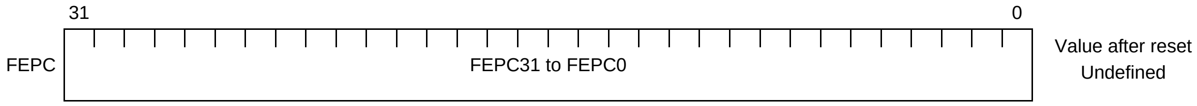


Table 3BC.7 FEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	FEPC31 to FEPC1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	Undefined

(d) FEPSW — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

	31	30	29												17	16	15	14						8	7	6	5	4	3	2	1	0
FEPSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

Value after reset
$0000\ 0020_H$

Table 3BC.8 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit stores the PSW.CU field setting when an FE level exception is acknowledged.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

(e) PSW — Program Status Word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by conditional instructions (Bcond, CMOV, etc.)).

CAUTIONS

1. When the LDSR instruction is used to change the contents of bit7 to 0 in this register, the changed contents become valid immediately after completion of the LDSR instruction execution. See “*APPENDIX A. Hazard Resolution Procedure for System Registers*” in *Software Manual* when the content of the other bits in this register is changed.
2. The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See **Table 3BC.9, Access Permission for PSW Register** for the access permission for each bit.

Table 3BC.9 Access Permission for PSW Register

Bit		Access Permission when Reading	Access Permission when Writing
30	UM	UM	SV*1
16	CU	UM	SV*1
15	EBV	UM	SV*1
7	NP	UM	SV*1
6	EP	UM	SV*1
5	ID	UM	SV*1
4	SAT	UM	UM
3	CY	UM	UM
2	OV	UM	UM
1	S	UM	UM
0	Z	UM	UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

	31	30	29												17	16	15	14						8	7	6	5	4	3	2	1	0
PSW	0	UM	0	0	0	0	0	0	0	0	0	0	0	0	0	CU	EBV	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z

Value after reset
0000 0020H

Table 3BC.10 PSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit indicates that the CPU is in user mode (in UM mode) 0: Supervisor mode 1: User mode	R/W	0
29 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	CU	This bit indicates the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor unusable exception occurs if an instruction for the coprocessor is executed or a coprocessor resource (system register) is accessed. CU bit 16: FPU	R/W	0

Table 3BC.10 PSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
15	EBV	This bit indicates the reset vector and exception vector operation. See the description on RBASE (**(q) RBASE — Reset Vector Base Address Register**) and EBASE (**(r) EBASE — Exception Handler Vector Address Register**) in this section.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit disables the acknowledgement of FE level exception. When an FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level and FE level exceptions. As for the exceptions which the NP bit disables the acknowledgment, see **Table 7BC.1, List of Exception Sources**. 0: The acknowledgement of FE level exception is enabled. 1: The acknowledgement of FE level exception is disabled.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt controlled by the interrupt controller is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0
5	ID	This bit disables the acknowledgement of EI level exception. When an EI level or FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level exception. As for the exceptions which the ID bit disables the acknowledgment, see **Table 7BC.1, List of Exception Sources**. This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. The change of the ID bit by the EI or ID instruction will be enabled from the next instruction. 0: EI level exception is not being processed or the section is not a critical section (after execution of EI instruction). 1: EI level exception is being processed or the section is a critical section (after execution of DI instruction).	R/W	1
4	SAT*1	This bit indicates that the operation result is saturated because the operation result of a saturated operation instruction has overflowed. This is a cumulative flag, so when the operation result of the saturated operation instruction becomes saturated, this bit is set to 1, but it is not cleared to 0 when the operation result for a subsequent instruction is not saturated. This bit is cleared to 0 by the LDSR instruction. This bit is neither set to 1 nor cleared to 0 when an arithmetic operation instruction is executed. 0: Not saturated 1: Saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV*1	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S*1	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative.	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. The operation result of the saturation processing is determined in accordance with the contents of the OV flag and S flag during a saturated operation. The SAT flag is set to 1 only when the OV flag is set to 1 in a saturated operation.

Operation Result Status	Flag Status			Operation Result after Saturation Processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF_H
Exceeded negative maximum value	1	1	1	8000 0000_H
Positive (maximum value not exceeded)	Value prior to operation is retained.	0	0	Operation result itself
Negative (maximum value not exceeded)			1	

(f) EIIC — EI Level Exception Source Register

The EIIC register retains the source of any EI level exception that occurs. The value retained in this register is an exception source code corresponding to a specific exception source.

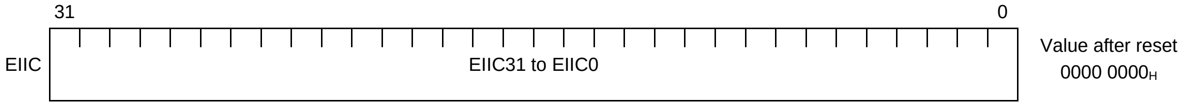


Table 3BC.11 EIIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIIC31 to EIIC0	These bits store the exception source code when an EI level exception is acknowledged. The EIIC15 to EIIC0 field stores the lower 16 bits of the exception source code. The EIIC31 to EIIC16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception, these bits are set to 0.	R/W	0

(g) FEIC — FE Level Exception Source Register

The FEIC register retains the source of any FE level exception that occurs. The value retained in this register is an exception source code corresponding to a specific exception source.

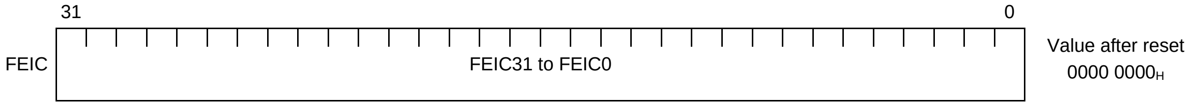


Table 3BC.12 FEIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEIC31 to FEIC0	These bits store the exception source code when an FE level exception is acknowledged. The FEIC15 to FEIC0 field stores the lower 16 bits of the exception source code. The FEIC31 to FEIC16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception, these bits are set to 0.	R/W	0

(h) CTPC — Status Save Register when Executing CALLT Instruction

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC. Be sure to set an even-numbered address to the CTPC register. An odd-numbered address cannot be specified.

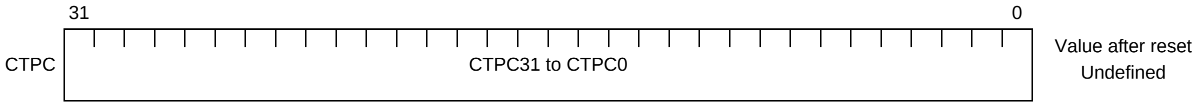


Table 3BC.13 CTPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTPC31 to CTPC1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

(i) CTPSW — Status Save Register when Executing CALLT Instruction

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

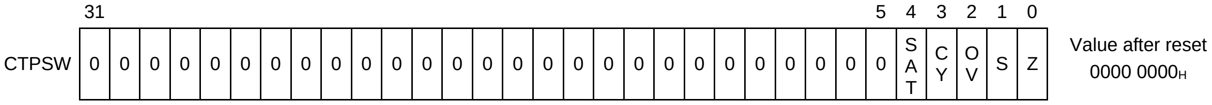


Table 3BC.14 CTPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4	SAT	This bit stores the PSW.SAT bit setting when the CALLT instruction is executed.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when the CALLT instruction is executed.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when the CALLT instruction is executed.	R/W	0
1	S	This bit stores the PSW.S bit setting when the CALLT instruction is executed.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when the CALLT instruction is executed.	R/W	0

(j) CTBP — CALLT Base Pointer Register

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

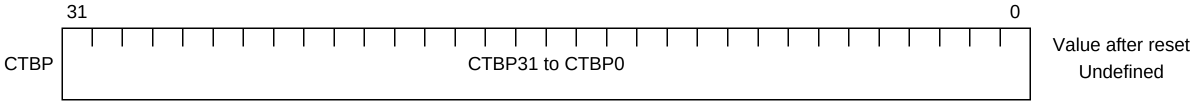


Table 3BC.15 CTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTBP31 to CTBP1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the start address of the table used by the CALLT instruction.	R/W	Undefined
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the start address of the table used by the CALLT instruction. Always set this bit to 0.	R	0

(k) ASID — Address Space ID Register

This register indicates the address space ID. This is used to identify the address space provided by the memory management function.

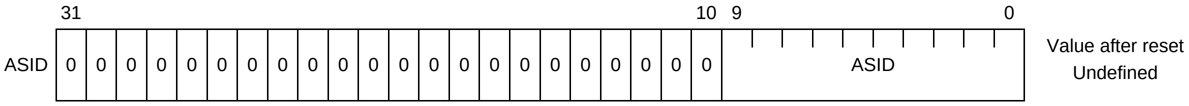


Table 3BC.16 ASID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
9 to 0	ASID	These bits indicate the address space ID.	R/W	Undefined

(l) EIWR — EI Level Exception Working Register

The EIWR register is used as a working register when an EI level exception has occurred.

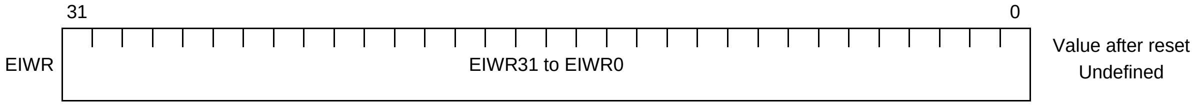


Table 3BC.17 EIWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIWR31 to EIWR0	These bits constitute a working register that can be used for any purpose during the servicing of an EI level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(m) FEWR — FE Level Exception Working Register

The FEWR register is used as a working register when an FE level exception has occurred.

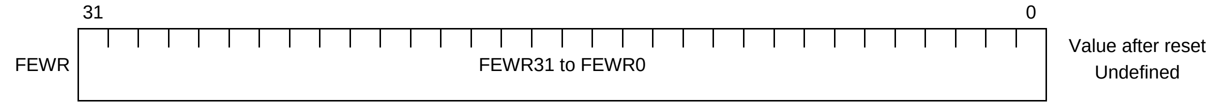


Table 3BC.18 FEWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEWR31 to FEWR0	These bits constitute a working register that can be used for any purpose during the servicing of an FE level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(n) HTCFG0 — Thread Configuration Register

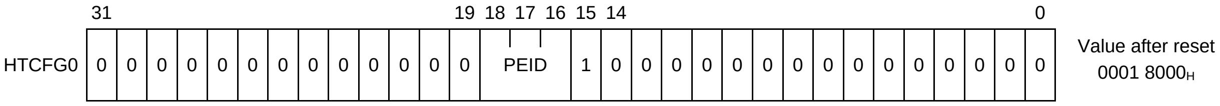


Table 3BC.19 HTCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	PEID	These bits indicate the processor element number.	R	001_B
15	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
14 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(o) MEA — Memory Error Address Register

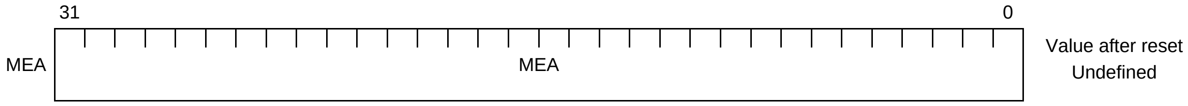


Table 3BC.20 MEA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MEA	These bits store the violation address when an MAE (misaligned) or MPU occurs.	R/W	Undefined

(p) MEI — Memory Error Information Register

This register is used to store information about the instruction that caused a misaligned (MAE) or memory protection (MDP) exception when such an exception occurred. This information is used during emulation.

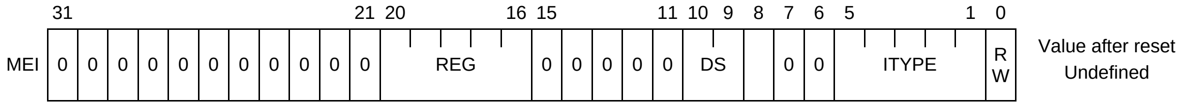


Table 3BC.21 MEI Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	REG	These bits indicate the number of the source or destination register accessed by the instruction that caused the exception. For details, see **Table 3BC.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
15 to 11	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
10, 9	DS	These bits indicate the data type of the instruction that caused the exception.*1 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) For details, see **Table 3BC.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
8	U	This bit indicates the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see **Table 3BC.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 1	ITYPE	These bits indicate the instruction that caused the exception. For details, see **Table 3BC.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
0	RW	This bit indicates whether the operation of the instruction that caused the exception was read (Load-memory) or write (Store-memory). 0: Read (Load-memory) 1: Write (Store-memory) For details, see **Table 3BC.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3BC.22 Instructions Causing Exceptions and Values of MEI Register

Instruction	REG	DS	U	RW	ITYPE
SLD.B	dst	0 (byte)	0 (signed)	0 (read)	00000_B
SLD.BU	dst	0 (byte)	1 (unsigned)	0 (read)	00000_B
SLD.H	dst	1 (halfword)	0 (signed)	0 (read)	00000_B
SLD.HU	dst	1 (halfword)	1 (unsigned)	0 (read)	00000_B
SLD.W	dst	2 (word)	0 (signed)	0 (read)	00000_B
SST.B	src	0 (byte)	0 (signed)	1 (write)	00000_B
SST.H	src	1 (halfword)	0 (signed)	1 (write)	00000_B

Table 3BC.22 Instructions Causing Exceptions and Values of MEI Register

Instruction	REG	DS	U	RW	ITYPE
SST.W	src	2 (word)	0 (signed)	1 (write)	00000_B
LD.B (disp16)	dst	0 (byte)	0 (signed)	0 (read)	00001_B
LD.BU (disp16)	dst	0 (byte)	1 (unsigned)	0 (read)	00001_B
LD.H (disp16)	dst	1 (halfword)	0 (signed)	0 (read)	00001_B
LD.HU (disp16)	dst	1 (halfword)	1 (unsigned)	0 (read)	00001_B
LD.W (disp16)	dst	2 (word)	0 (signed)	0 (read)	00001_B
ST.B (disp16)	src	0 (byte)	0 (signed)	1 (write)	00001_B
ST.H (disp16)	src	1 (halfword)	0 (signed)	1 (write)	00001_B
ST.W (disp16)	src	2 (word)	0 (signed)	1 (write)	00001_B
LD.B (disp23)	dst	0 (byte)	0 (signed)	0 (read)	00010_B
LD.BU (disp23)	dst	0 (byte)	1 (unsigned)	0 (read)	00010_B
LD.H (disp23)	dst	1 (halfword)	0 (signed)	0 (read)	00010_B
LD.HU (disp23)	dst	1 (halfword)	1 (unsigned)	0 (read)	00010_B
LD.W (disp23)	dst	2 (word)	0 (signed)	0 (read)	00010_B
ST.B (disp23)	src	0 (byte)	0 (signed)	1 (write)	00010_B
ST.H (disp23)	src	1 (halfword)	0 (signed)	1 (write)	00010_B
ST.W (disp23)	src	2 (word)	0 (signed)	1 (write)	00010_B
LD.DW (disp23)	dst	3 (double-word)	0 (signed)	0 (read)	00010_B
ST.DW (disp23)	src	3 (double-word)	0 (signed)	1 (write)	00010_B
LDL.W	dst	2 (word)	0 (signed)	0 (read)	00111_B
STC.W	src	2 (word)	0 (signed)	1 (write)	00111_B
CAXI	dst	2 (word)	0 (signed)	0 (read)/1 (write)	01000_B
SET1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
CLR1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
NOT1	—	0 (byte)	0 (signed)	0 (read)/1 (write)	01001_B
TST1	—	0 (byte)	0 (signed)	0 (read)	01001_B
PREPARE	—	2 (word)	0 (signed)	1 (write)	01100_B
DISPOSE	—	2 (word)	0 (signed)	0 (read)	01100_B
PUSHSP	—	2 (word)	0 (signed)	1 (write)	01101_B
POPSP	—	2 (word)	0 (signed)	0 (read)	01101_B
SWITCH	—	1 (halfword)	0 (signed)	0 (read)	10000_B
CALLT	—	1 (halfword)	1 (unsigned)	0 (read)	10001_B
SYSCALL	—	2 (word)	0 (signed)	0 (read)	10010_B
CACHE	—	—	—	—	—
Interrupt (table reference)*1	—	2 (word)	0 (signed)	0 (read)	10101_B

Note 1. When reading the interrupt vector by using the table reference method.

NOTE

dst: destination register number, src: source register number

(q) RBASE — Reset Vector Base Address Register

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this vector address is also used as the exception vector address.

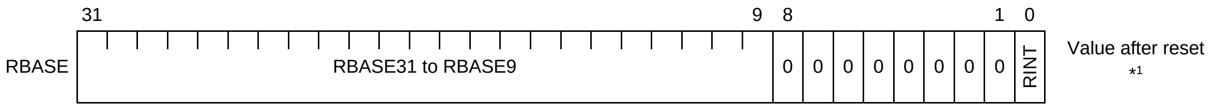


Table 3BC.23 RBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	RBASE31 to RBASE9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. For RBASE8 to RBASE0, 0 is used implicitly.	R	0000 0000 0000 0000 0000 000$_B$*1
8 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	RINT	When the RINT bit is set, the exception handler address for interrupt servicing is reduced. See **Section 7BC.10.1, Direct Vector Method**. This bit is valid when PSW.EBV = 0.	R	0

Note 1. The value depends on the reset vector. The values set at shipment are shown in the table. When the reset vector is modified, the address will be changed.

(r) EBASE — Exception Handler Vector Address Register

This register indicates the exception handler vector address. This register is valid when the PSW.EBV bit is 1.

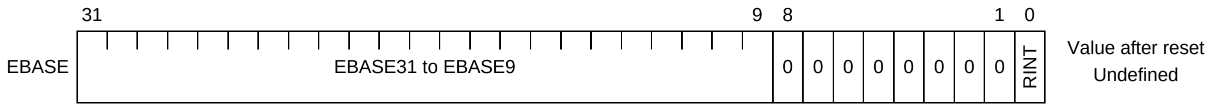


Table 3BC.24 EBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	EBASE31 to EBASE9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. For EBASE8 to EBASE0, 0 is used implicitly.	R/W	Undefined
8 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	RINT	When the RINT bit is set, the exception handler address for interrupt servicing is reduced. See **Section 7BC.10.1, Direct Vector Method**.	R/W	Undefined

(s) INTBP — Base Address Register of the Interrupt Handler Address Table

This register indicates the base address of the address table when the table reference method is selected as the interrupt handler address selection method.

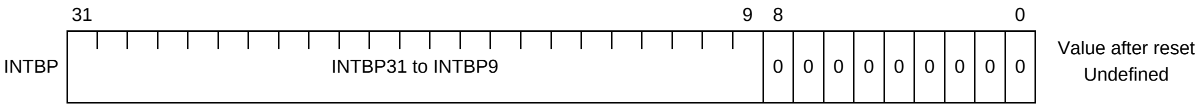


Table 3BC.25 INTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	INTBP31 to INTBP9	These bits indicate the base pointer address for an interrupt when the table reference method is used. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt specified by the table reference method (EIINT0 to EIINT511) is acknowledged. For INTBP8 to INTBP0, 0 is used implicitly.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(t) PID — Processor ID Register

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

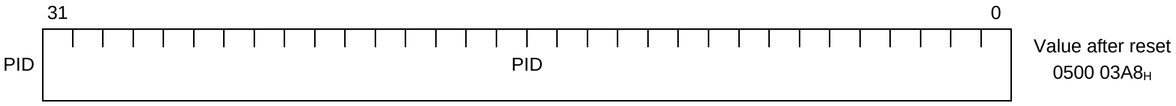


Table 3BC.26 PID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.	R	05_H
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bits 23 to 11: Reserved Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operation function Bit 8: Memory protection function (MPU)	R	0003_H
7 to 0		Version Identifier This identifier indicates the version of the processor.	R	$A8_H$

(u) SCCFG — SYSCALL Operation Setting Register

This register is used to specify operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

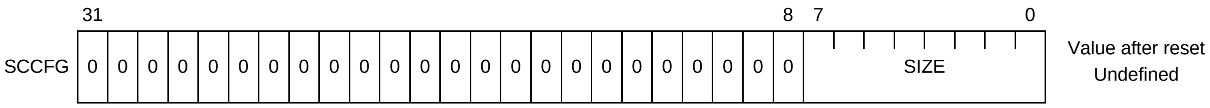


Table 3BC.27 SCCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SIZE	These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If vectors exceeding the maximum number of entries are specified for the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry.	R/W	Undefined

(v) SCBP — SYSCALL Base Pointer Register

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

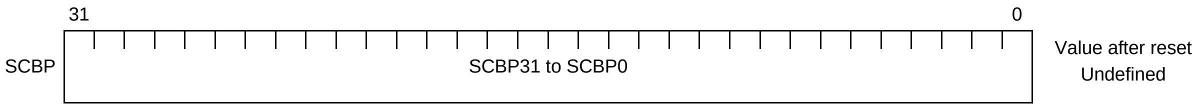


Table 3BC.28 SCBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	SCBP31 to SCBP2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction.	R/W	Undefined
1, 0	SCBP1, SCBP0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction. Always set these bits to 0.	R	0

(w) MCFG0 — Machine Configuration Register

This register indicates the CPU configuration.

	31													18	17	16	15												3	2	1	0	
MCFG0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPID		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Value after reset 0001 0004_H

Table 3BC.29 MCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 18	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
17, 16	SPID	These bits indicate the system protection number.	R/W	01_B
15 to 3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(x) MCTL — Machine Control Register

This register is used to control the CPU.

	31																													2	1	0	
MCTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MA	UIC	Value after reset 8000 0002_H

Table 3BC.30 MCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
30 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	MA	This bit is used to control the misaligned access. 0: In the event of a misaligned access, an exception is always generated.*[1] 1: The correct operation is controlled by hardware.*[2]	R/W	1
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction in user mode becomes possible.	R/W	0

Note 1. Excluding LD.DW, and ST.DW for word boundary allocation.

Note 2. Exception still occurs in case of LD.DW or ST.DW for misaligned access except word boundary allocation.

(3) Interrupt Function Registers

Table 3BC.31 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR7, 1	FPIPR	FPI exception interrupt priority setting register	SV
SR10, 2	ISPR	Priority of interrupt being serviced register	SV
SR11, 2	PMR	Interrupt priority masking register	SV
SR12, 2	ICSR	Interrupt control status register	SV
SR13, 2	INTCFG	Interrupt function setting register	SV

(a) FPIPR — FPI Exception Interrupt Priority Setting Register

This register is used to set the interrupt priority of FPI exception.

	31																										5	4 0
FPIPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FPIPR

Value after reset
0000 0000$_H$

Table 3BC.32 FPIPR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	FPIPR	These bits are used to specify the interrupt priority of floating-point operation exceptions (imprecise) (FPI). Specify values from 0 to 16. Specifying 17 or greater is prohibited. FPI exceptions are handled using the specified interrupt priority. If an FPI exception occurs at the same time as an interrupt that has the same priority, the FPI exception is prioritized. **NOTE:** A set value of more than 16 is treated as 16.	R/W	0

(b) ISPR — Priority of Interrupt being Serviced Register

This register retains the priority of the *EIINTn* interrupt being serviced by the CPU. This priority value is then used to perform priority ceiling processing when multiple interrupts occur.

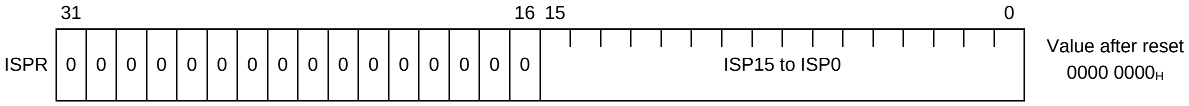


Table 3BC.33 ISPR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value after Reset</th></tr>
<tr><td>31 to 16</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>15 to 0</td><td>ISP15 to ISP0</td><td>These bits indicate the acknowledgment status of an <i>EIINTn</i> interrupt with a priority that corresponds to the relevant bit position.
0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged.
1: An interrupt request for an interrupt whose priority corresponds to the relevant bit position is being serviced by the CPU core.

The bit positions correspond to the following priority levels.
<table>
<tr><th>Bit</th><th>Priority</th></tr>
<tr><td>0</td><td>Priority 0 (highest)</td></tr>
<tr><td>1</td><td>Priority 1</td></tr>
<tr><td colspan="2">:</td></tr>
<tr><td>14</td><td>Priority 14</td></tr>
<tr><td>15</td><td>Priority 15 (lowest)</td></tr>
</table>
When an interrupt request (<i>EIINTn</i>) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15 to ISP0 bits that are set to 1 (0 is the highest priority) is cleared to 0.*1
While a bit in this register is set to 1, same or lower priority interrupts (<i>EIINTn</i>) and FPI exceptions*2 are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged.
When performing software-based priority control using the PMR register, be sure to clear this register by using the INTCFG.ISPC bit.</td><td>R*3</td><td>0</td></tr>
</table>

Note 1. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) to the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.

Note 2. Since FPI exceptions have the same level of priority as EIINTn interrupts, they are affected by interrupts in the same way as the ISPR. The priority of FPI exceptions is set by the FPIPR register.

Note 3. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

(c) PMR — Interrupt Priority Masking Register

This register is used to mask the specified interrupt priority.

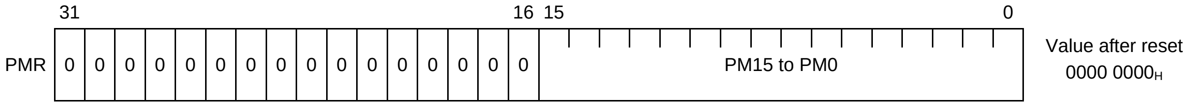


Table 3BC.34 PMR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value after Reset</th></tr>
<tr><td>31 to 16</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>15 to 0</td><td>PM15 to PM0</td><td>These bits mask an interrupt request with a priority level that corresponds to the relevant bit position.
0: Servicing of an interrupt with a priority that corresponds to the relevant bit position is enabled.
1: Servicing of an interrupt with a priority that corresponds to the relevant bit position is disabled.

The bit positions correspond to the following priority levels:
<table>
<tr><th>Bit</th><th>Priority</th></tr>
<tr><td>0</td><td>Priority 0 (highest)</td></tr>
<tr><td>1</td><td>Priority 1</td></tr>
<tr><td colspan="2">:</td></tr>
<tr><td>14</td><td>Priority 14</td></tr>
<tr><td>15</td><td>Priority 15 and priority 16 (lowest)</td></tr>
</table>
While a bit in this register is set to 1, interrupts (*EIINTn*) and FPI exceptions*[1] with the priority corresponding to that bit are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged*[2].</td><td>R/W</td><td>0</td></tr>
</table>

Note 1. Since FPI exceptions are specified as the same level of priority as that of interrupts (*EIINTn*), it is affected by the PMR like interrupts. The priority of FPI exceptions is set by the FPIPR register.

Note 2. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, $FF00_H$ can be set, but $F0F0_H$ or $00FF_H$ cannot.

(d) ICSR — Interrupt Control Status Register

This register indicates the interrupt control status in the CPU.

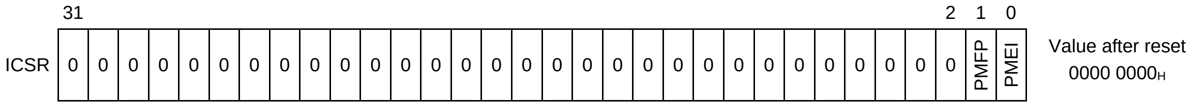


Table 3BC.35 ICSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	PMFP	This bit indicates that an FPI exception with the priority level masked by the PMR register exists.	R	0
0	PMEI	This bit indicates that an interrupt (*EIINTn*) with the priority level masked by the PMR register exists.	R	0

(e) INTCFG — Interrupt Function Setting Register

This register is used to specify settings related to the CPU’s internal interrupt function.

	31																														1	0
INTCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISPC

Value after reset
0000 0000$_H$

Table 3BC.36 INTCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	ISPC	This bit specifies how the ISPR register is updated. 0: The ISPR register is automatically updated. Updates triggered by the program (via execution of LDSR instruction) are ignored. 1: The ISPR register is not automatically updated. Updates triggered by the program (via execution of LDSR instruction) are performed. If this bit is cleared to 0, the bits of the ISPR register are automatically set to 1 when an interrupt (*EIINTn*) is acknowledged, and cleared to 0 when the EIRET instruction is executed. In this case, updating by the program (via execution of an LDSR instruction) is ignored. If this bit is set to 1, the bits of the ISPR register are not updated by the acknowledgement of an interrupt (*EIINTn*) or by execution of the EIRET instruction. In this case, the bits can be updated by an LDSR instruction executed by the program. In normal cases, the ISPC bit should be cleared to 0. When performing software-based control of interrupt priorities, however, set this bit (1) and perform priority control by using the PMR register.	R/W	0

(4) FPU Function Registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations. The RH850/F1KM supports single-precision floating-point instruction and thirty-two 32-bit registers can be specified.

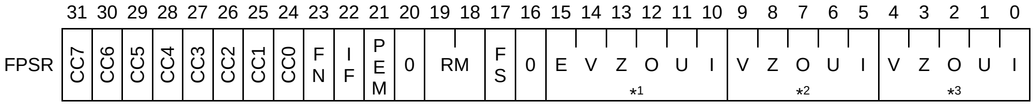
These registers correspond to general-purpose registers r0 to r31. The FPU can use the following system registers to control floating-point operation

Table 3BC.37 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Floating-point operation setting/status register	CU and SV
SR7, 0	FPEPC	Floating-point exception program counter register	CU and SV
SR8, 0	FPST	Floating-point operation status register	CU
SR9, 0	FPCC	Floating-point operation comparison result register	CU
SR10, 0	FPCFG	Floating-point operation configuration register	CU
SR11, 0	FPEC	Floating-point exception control register	CU and SV

(a) FPSR — Floating-point Operation Setting/Status Register

This register indicates the execution status of floating-point operations and any exceptions that occur.



Value after reset
See below

Note 1. Cause bits (XC)

Note 2. Enable bits(XE)

Note 3. Preservation bits (XP)

Table 3BC.38 FPSR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value after Reset</th></tr>
<tr><td>31 to 24</td><td>CC[7:0]</td><td>These are the CC (condition) bits. They store the results of floating-point comparison instructions. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction.
0: Comparison result is false
1: Comparison result is true</td><td>R/W</td><td>Undefined</td></tr>
<tr><td>23</td><td>FN</td><td>This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number.</td><td>R/W</td><td>0</td></tr>
<tr><td>22</td><td>IF</td><td>This bit accumulates and indicates information about the flushing of input operands.</td><td>R/W</td><td>0</td></tr>
<tr><td>21</td><td>PEM</td><td>This bit specifies whether to handle an exception as a precise exception. If the PEM bit is 1, exceptions that are caused by the execution of a floating-point operation instruction are handled as precise exceptions.</td><td>R/W</td><td>0</td></tr>
<tr><td>20</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>19, 18</td><td>RM</td><td>These are the rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.
<table>
<tr><th colspan="2">RM Bits</th><th rowspan="2">Mnemonic</th><th rowspan="2">Description</th></tr>
<tr><th>19</th><th>18</th></tr>
<tr><td>0</td><td>0</td><td>RN</td><td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td></tr>
<tr><td>0</td><td>1</td><td>RZ</td><td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td></tr>
<tr><td>1</td><td>0</td><td>RP</td><td>Rounds the result toward +∞. The result is nearest to a value greater than the accurate result with infinite accuracy.</td></tr>
<tr><td>1</td><td>1</td><td>RM</td><td>Rounds the result toward −∞. The result is nearest to a value less than the accurate result with infinite accuracy.</td></tr>
</table></td><td>R/W</td><td>00</td></tr>
<tr><td>17</td><td>FS</td><td>This bit enables values that cannot be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign.
Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode.
<table>
<tr><th rowspan="2">Operation Result that is a Subnormal Number</th><th colspan="4">Rounding Mode and Value after Flushing</th></tr>
<tr><th>RN*1</th><th>RZ</th><th>RP</th><th>RM</th></tr>
<tr><td>Positive</td><td>+0</td><td>+0</td><td>$+2^{Emin}$</td><td>+0</td></tr>
<tr><td>Negative</td><td>−0</td><td>−0</td><td>−0</td><td>-2^{Emin}</td></tr>
</table>
Note 1. If the rounding mode is RN and the FPSR.FN bit is set to 1, flushing will occur in the direction of higher accuracy.</td><td>R/W</td><td>1</td></tr>
<tr><td>16</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>15 to 10</td><td>XC (E, V, Z, O, U, I)</td><td>These are the cause bits.</td><td>R/W</td><td>Undefined</td></tr>
</table>

Table 3BC.38 FPSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
9 to 5	XE (V, Z, O, U, I)	These are the enable bits.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are the preservation bits.	R/W	Undefined

(b) FPEPC — Floating-point Exception Program Counter Register

When an exception that is enabled by an enable bit occurs, the program counter (PC) of the instruction that caused the exception is stored.

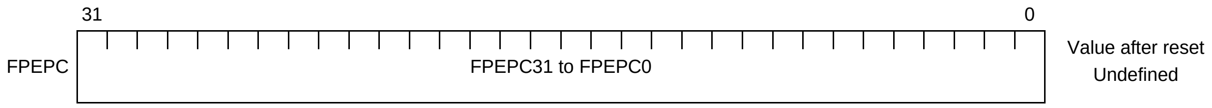
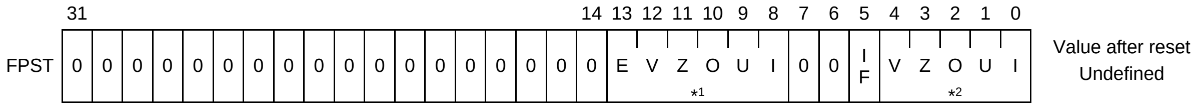


Table 3BC.39 FPEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	FPEPC31 to FPEPC1	These bits store the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R/W	Undefined
0	FPEPC0	This bit stores the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs. Always set this bit to 0.	R	0

(c) FPST — Floating-point Operation Status Register

This register reflects the contents of the FPSR register bits related to the operation status.



Note 1. Cause bits (XC)

Note 2. Preservation bits (XP)

Table 3BC.40 FPST Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E, V, Z, O, U, I)	These are cause bits. Values written to these bits are reflected in FPSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are preservation bits. Values written to these bits are reflected in FPSR.XP bits.	R/W	Undefined

(d) FPCC — Floating-point Operation Comparison Result Register

This register reflects the contents of the FPSR.CC[7:0] bits.

	31																							8	7	6	5	4	3	2	1	0	
FPCC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	Value after reset Undefined

Table 3BC.41 FPCC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	CC[7:0]	These are CC (condition) bits. They store the result of a floating-point comparison instruction. The CC[7:0] bits are not affected by any instructions except the comparison instruction and LDSR instruction. Values written to these bits are reflected in the CC[7:0] bits of FPSR. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined

(e) FPCFG — Floating-point Operation Configuration Register

This register reflects the contents of the FPSR register bits related to the operation settings.

	31																					10	9	8	7		5	4	3	2	1	0	
FPCFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RM		0	0	0	V	Z	O	U	I	Value after reset 0000 0000$_H$
																														*1			

Note 1. Enable bits (XE)

Table 3BC.42 FPCFG Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value after Reset</th></tr>
<tr><td>31 to 10</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>9, 8</td><td>RM</td><td>These are rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.
Values written to these bits are reflected in RM bits of FPSR.
<table>
<tr><th colspan="2">RM Bits</th><th rowspan="2">Mnemonic</th><th rowspan="2">Description</th></tr>
<tr><th>9</th><th>8</th></tr>
<tr><td>0</td><td>0</td><td>RN</td><td>Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.</td></tr>
<tr><td>0</td><td>1</td><td>RZ</td><td>Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.</td></tr>
<tr><td>1</td><td>0</td><td>RP</td><td>Rounds the result toward +∞. The result is nearest to a value greater than the accurate result with infinite accuracy.</td></tr>
<tr><td>1</td><td>1</td><td>RM</td><td>Rounds the result toward −∞. The result is nearest to a value less than the accurate result with infinite accuracy.</td></tr>
</table>
</td><td>R/W</td><td>0</td></tr>
<tr><td>7 to 5</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>4 to 0</td><td>XE
(V, Z, O, U, I)</td><td>These are the enable bits.</td><td>R/W</td><td>0</td></tr>
</table>

(f) FPEC — Floating-point Exception Control Register

This register controls the floating-point operation exception.

	31																														1	0
FPEC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FPIVD

Value after reset
$0000\ 0000_H$

Table 3BC.43 FPEC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	FPIVD*[1]	This bit indicates the status of reporting the FPI exception. If this bit is set to 1, the FPI exception is reported to the CPU but is not acknowledged. It is automatically cleared to 0 when the CPU acknowledges the FPI exception. While this bit is set to 1, all the floating-point instructions are invalidated. Report of the FPI exception can be canceled by clearing (0) this bit by the LDSR instruction while it is set to 1. When report of the FPI exception is canceled, the CPU does not acknowledge the FPI exception. 0: FPI exception is not reported. 1: FPI exception is reported.	R/W	0

Note 1. The FPIVD bit can only be cleared to 0 by the write operation of the LDSR instruction. It cannot be set to 1.

(5) MPU Function Registers

Table 3BC.44 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	SV
SR1, 5	MPRC	MPU region control	SV
SR4, 5	MPBRGN	MPU base region number	SV
SR5, 5	MPTRGN	MPU end region number	SV
SR8, 5	MCA	Memory protection setting check address	SV
SR9, 5	MCS	Memory protection setting check size	SV
SR10, 5	MCC	Memory protection setting check command	SV
SR11, 5	MCR	Memory protection setting check result	SV
SR0, 6	MPLA0	Protection area lower limit address	SV
SR1, 6	MPUA0	Protection area upper limit address	SV
SR2, 6	MPAT0	Protection area attribute	SV
SR4, 6	MPLA1	Protection area lower limit address	SV
SR5, 6	MPUA1	Protection area upper limit address	SV
SR6, 6	MPAT1	Protection area attribute	SV
SR8, 6	MPLA2	Protection area lower limit address	SV
SR9, 6	MPUA2	Protection area upper limit address	SV
SR10, 6	MPAT2	Protection area attribute	SV
SR12, 6	MPLA3	Protection area lower limit address	SV
SR13, 6	MPUA3	Protection area upper limit address	SV
SR14, 6	MPAT3	Protection area attribute	SV
SR16, 6	MPLA4	Protection area lower limit address	SV
SR17, 6	MPUA4	Protection area upper limit address	SV
SR18, 6	MPAT4	Protection area attribute	SV
SR20, 6	MPLA5	Protection area lower limit address	SV
SR21, 6	MPUA5	Protection area upper limit address	SV
SR22, 6	MPAT5	Protection area attribute	SV
SR24, 6	MPLA6	Protection area lower limit address	SV
SR25, 6	MPUA6	Protection area upper limit address	SV
SR26, 6	MPAT6	Protection area attribute	SV
SR28, 6	MPLA7	Protection area lower limit address	SV
SR29, 6	MPUA7	Protection area upper limit address	SV
SR30, 6	MPAT7	Protection area attribute	SV
SR0, 7	MPLA8	Protection area lower limit address	SV
SR1, 7	MPUA8	Protection area upper limit address	SV
SR2, 7	MPAT8	Protection area attribute	SV
SR4, 7	MPLA9	Protection area lower limit address	SV
SR5, 7	MPUA9	Protection area upper limit address	SV
SR6, 7	MPAT9	Protection area attribute	SV
SR8, 7	MPLA10	Protection area lower limit address	SV
SR9, 7	MPUA10	Protection area upper limit address	SV
SR10, 7	MPAT10	Protection area attribute	SV
SR12, 7	MPLA11	Protection area lower limit address	SV
SR13, 7	MPUA11	Protection area upper limit address	SV

Table 3BC.44 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR14, 7	MPAT11	Protection area attribute	SV
SR16, 7	MPLA12	Protection area lower limit address	SV
SR17, 7	MPUA12	Protection area upper limit address	SV
SR18, 7	MPAT12	Protection area attribute	SV
SR20, 7	MPLA13	Protection area lower limit address	SV
SR21, 7	MPUA13	Protection area upper limit address	SV
SR22, 7	MPAT13	Protection area attribute	SV
SR24, 7	MPLA14	Protection area lower limit address	SV
SR25, 7	MPUA14	Protection area upper limit address	SV
SR26, 7	MPAT14	Protection area attribute	SV
SR28, 7	MPLA15	Protection area lower limit address	SV
SR29, 7	MPUA15	Protection area upper limit address	SV
SR30, 7	MPAT15	Protection area attribute	SV

(a) MPM — Memory Protection Operation Mode Register

The memory protection mode register is used to define the basic operating state of the memory protection function.

	31																												2	1	0	
MPM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SVP	MPE	Value after reset 0000 0000$_H$

Table 3BC.45 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	SVP	In SV mode (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*[1] 0: As usual, implicitly enable all access in SV mode. 1: Restrict access according to the SX, SW, and SR bits even in SV mode.*[2]	R/W	0
0	MPE	This bit is used to specify whether to enable or disable the MPU function. 0: Disable 1: Enable	R/W	0

Note 1. If the SVP bit is set to 1, access will be restricted in accordance with the setting for each protection area, even in SV mode. Therefore, specify the protection area beforehand so that the access from the program which set the SVP bit is not restricted.

Note 2. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(b) MPRC — MPU Region Control Register

	31															16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MPRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0	Value after reset 0000 0000$_H$

Table 3BC.46 MPRC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	E15 to E0	These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 15 to 0).	R/W	0

(c) MPBRGN — MPU Base Region Register

This register indicates the minimum usable MPU area number.

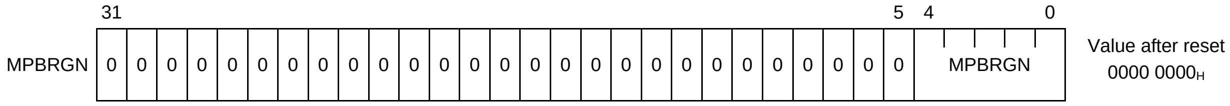


Table 3BC.47 MPBRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	MPBRGN	These bits indicate the smallest number of an MPU area. These bits always indicate 0.	R	0

(d) MPTRGN — MPU End Region Register

This register indicates the maximum usable MPU area number + 1.

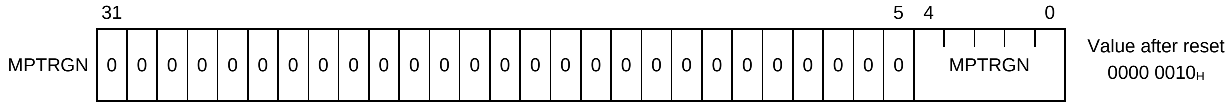


Table 3BC.48 MPTRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	MPTRGN	These bits indicate the largest number of an MPU area + 1. These bits indicate the maximum number of MPU areas incorporated into the hardware.	R	10000_B

(e) MCA — Memory Protection Setting Check Address Register

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

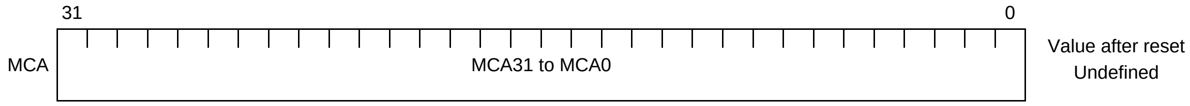


Table 3BC.49 MCA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCA31 to MCA0	These bits are used to specify the start address of the memory area that is subject to a memory protection setting check in bytes.	R/W	Undefined

(f) MCS — Memory Protection Setting Check Size Register

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

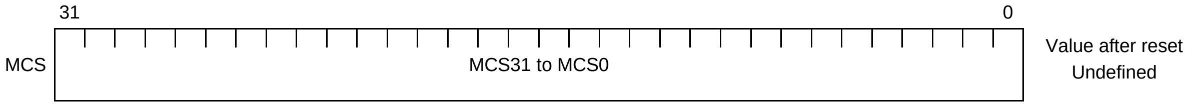


Table 3BC.50 MCS Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCS31 to MCS0	These bits are used to specify the size of the memory area that is subject to a memory protection setting check and the size of the target area in bytes. Because the specified size is assumed to represent an unsigned integer, it is not possible to check an area in the direction in which the address value decreases relative to the MCA register value. Do not specify 0000 0000_H for the MCS register.	R/W	Undefined

(g) MCC — Memory Protection Setting Check Command Register

This register is used to specify the base address of the area where memory protection settings are checked.

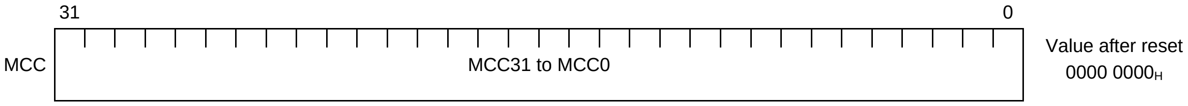


Table 3BC.51 MCC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCC31 to MCC0	When any value is written to the MCC register, a memory protection setting check starts. By setting up the MCA/MCS register and then writing to the MCC register, results are stored in MCR. Because the check is started by any written value, a check can be started by using r0 as the source register without using any unnecessary registers. Note that, for the check, the results are applied according to each area setting regardless of the state of the PSW.UM bit. When the MCC register is read, value 0000 0000_H is always returned.	R/W	0

(h) MCR — Memory Protection Setting Check Result Register

This register is used to store the results of a memory protection setting check.

Be sure to clear bits 31 to 9, 7 and 6.

	31																						9	8	7	6	5	4	3	2	1	0	
MCR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OV	0	0	SXE	SWE	SRE	UXE	UWE	URE	Value after reset Undefined

Table 3BC.52 MCR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	OV	If the specified area includes 0000 0000_H or 7FFF FFFF_H, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
4	SWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
3	SRE	If the specified area is contained within one of the protection areas and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
2	UXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
1	UWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
0	URE	If the specified area is contained within one of the protection areas and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

(i) MPLAn — Protection Area Lower Limit Address Register

These registers indicate the lower limit address of area n (where n = 0 to 15).

	31	2	1	0	
MPLAn	MPLAn		0	0	Value after reset Undefined

Table 3BC.53 MPLAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPLA31 to MPLA2	These bits indicate the lower limit address of area n. For MPLA1 and MPLA0, 0 is used implicitly.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(j) MPUAn — Protection Area Upper Limit Address Register

These registers indicate the upper limit address of area n (where n = 0 to 15).

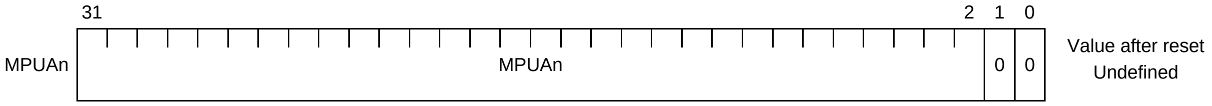


Table 3BC.54 MPUAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPUA31 to MPUA2	These bits indicate the upper limit address of area n. For MPUAn.MPUA1 and MPUA0, 1 is used implicitly.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(k) MPATn — Protection Area Attribute Register

These registers indicate the attributes of area n (where n = 0 to 15).

	31					26	25 16	15							8	7	6	5	4	3	2	1	0	Value after reset
MPATn	0	0	0	0	0	0	ASID	0	0	0	0	0	0	0	0	E	G	SX	SW	SR	UX	UW	UR	Undefined

Table 3BC.55 MPATn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 26	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
25 to 16	ASID	These bits indicate the ASID value to be used as the area match condition.	R/W	Undefined
15 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	E	This bit indicates whether area n is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled.	R/W	0
6	G	0: Areas match only if ASIDs are equal. 1: Areas match even if ASIDs are not equal. If this bit is 0, MPATn.ASID = ASID.ASID is used as the area match condition. If this bit is 1, areas may match even if the values of MPATn.ASID and ASID.ASID are not equal.	R/W	Undefined
5	SX	This bit indicates the execution privilege for the supervisor mode.*1 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
4	SW	This bit indicates whether writing is enabled in the supervisor mode.*1 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
3	SR	This bit indicates whether writing is enabled in the supervisor mode.*1 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined
2	UX	This bit indicates the execution privilege for the user mode. 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
1	UW	This bit indicates whether writing is enabled in the user mode. 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
0	UR	This bit indicates whether writing is enabled in the user mode. 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Note 1. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(6) Cache Operation Function Registers

The RH850/F1KM does not include a cache operation function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

Table 3BC.56 Cache Operation Function Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR12, 4	BWERRL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR13, 4	BWERRH		SV
SR14, 4	BRERRL		SV
SR15, 4	BRERRH		SV
SR16, 4	ICTAGL		SV
SR17, 4	ICTAGH		SV
SR18, 4	ICDATL		SV
SR19, 4	ICDATH		SV
SR20, 4	DCTAGL		SV
SR21, 4	DCTAGH		SV
SR22, 4	DCDATL		SV
SR23, 4	DCDATH		SV
SR24, 4	ICCTRL		SV
SR25, 4	DCCTRL		SV
SR26, 4	ICCFG		SV
SR27, 4	DCCFG		SV
SR28, 4	ICERR		SV
SR29, 4	DCERR		SV

3BC.2.1.3 Instruction

See “*Instruction*” in *Software Manual*.

A snooze instruction halts operation of the CPU core for 32 clocks.

3BC.2.2 Buffers for Code Flash

3BC.2.2.1 Features

CPU accesses Code flash by two paths; instruction fetch access is direct to Flash interface, and data access is via System interconnect to Code Flash. Both paths equip buffers, which can be cleared by software. See also **Figure 40B.1, Block Diagram of Code Flash ECC**, **Figure 40C.1, Block Diagram of Code Flash ECC** for ECC decoders in these paths.

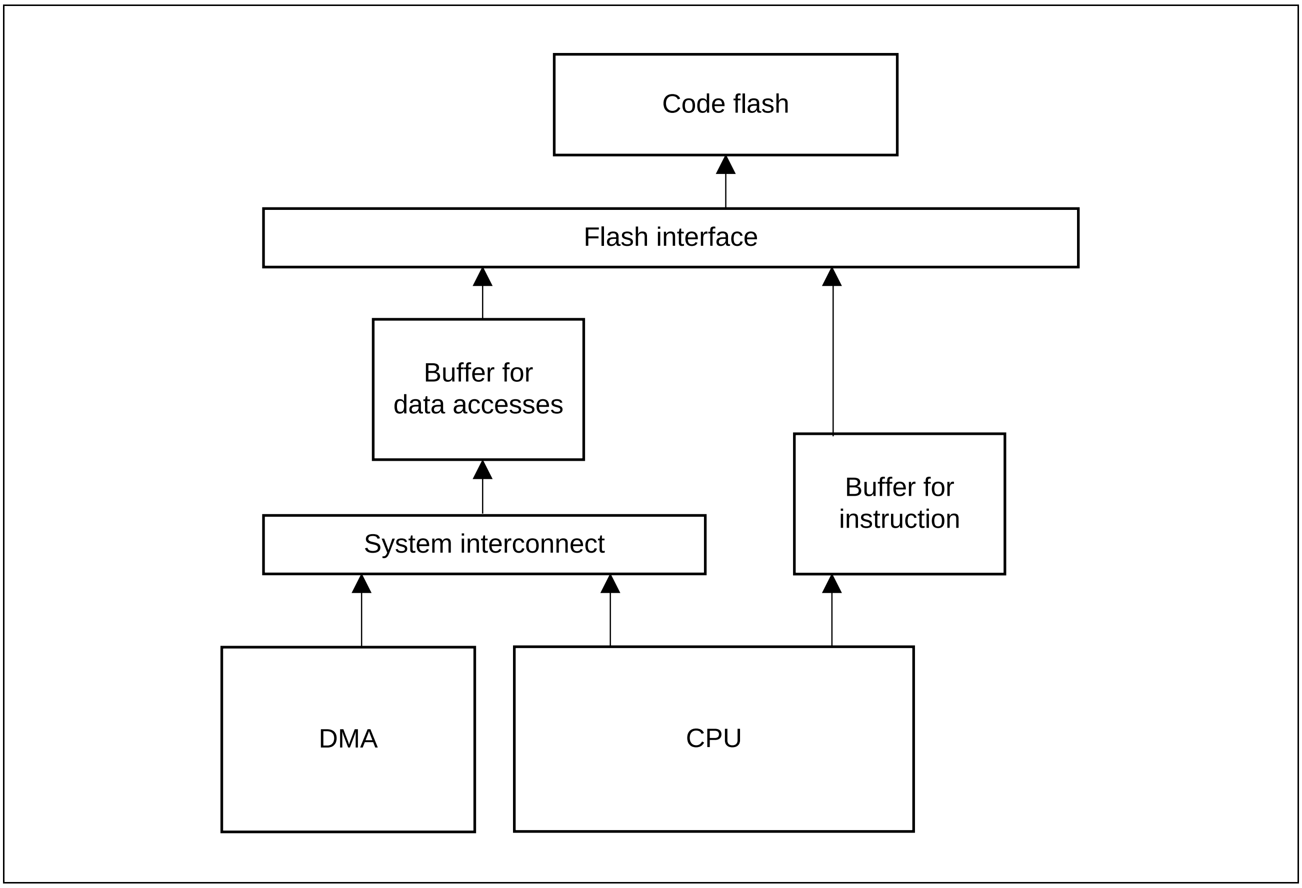


Figure 3BC.3 Buffers for Code Flash

3BC.2.2.2 Function of Buffers

One-line buffer with 128 bits is mounted for instruction fetches to code flash. The data is read out from the buffer after the next access to the same address, so the code flash is not accessed again within 128 bits location.

One-line buffer with 128 bits is mounted as a data buffer. This buffer is not only used by the CPU but also used by DMA via system interconnect. The data is read out from the buffer if the next access is within the same 128 bits boundary.

RH850/F1KM-S4:
16 entry branch history buffer is mounted to decrease branch penalty.
One buffer control register named FBUFCCTL is equipped. Using the FBUFCCTL register, the software can clear these three buffers.

RH850/F1KM-S1:
One buffer control register named FBUFCCTL is equipped. Using the FBUFCCTL register, the software can clear these two buffers.

3BC.2.2.3 Registers for Buffer Control

(1) List of Buffer Control Registers

Table 3BC.57 Buffer Control Register (Base Address: FFC5 B000$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after Reset
							1	8	16	32	
FBUF_CTRL	+000$_H$	4	Flash buffer clear control register	FBUFCCTL	—	R/W	—	✓	✓	✓	0000 0000$_H$

(2) Register Sets

Access: FBUFCCTL register can be read or written in 32-bit units.
FBUFCCTLL register can be read or written in 16-bit units.
FBUFCCTLLL register can be read or written in 8-bit units.

Address: FBUFCCTL: FFC5 B000$_H$
FBUFCCTLL: FFC5 B000$_H$
FBUFCCTLLL: FFC5 B000$_H$

	31																														1	0
FBUFCCTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FBUFCLR

Value after reset
0000 0000$_H$

Table 3BC.58 FBUFCCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	R	0
0	FBUFCLR	Buffer clear bits. To clear buffers, write 1 to this bit and then write 0. 0: Buffers are valid 1: Buffers are invalid (cleared)	R/W	0

Please do following procedures when you want to clear the buffers.

Step 1: Write 0 to FBUFCCTL.FBUFCLR

Step 2: Write 1 to FBUFCCTL.FBUFCLR

Step 3: Write 0 to FBUFCCTL.FBUFCLR

Step 4: Read the FBUFCCTL register (dummy read)

Step 5: Execute the SYNCP instruction

Step 6: Execute the SYNCI instruction

If you do not do Step 3 after Step 2, the buffers are kept invalid during FBUFCCTL.FBUFCLR = 1.

3BC.2.3 Reliability Functions

3BC.2.3.1 PE Guard Function (PEG)

(1) Overview of the PEG Function

The PEG is a constituent of the slave guard system to prevent unauthorized access to the resources in the CPU (PE) from an external master. This function protects access to the local RAM*[1] in the PE. In the initial state after a reset, access by masters other than own PE is disabled. Setting the registers listed in (3) List of PEG Protection Setting Registers enables access by masters other than own PE.

Note 1. It is the local RAM and the retention RAM in RH850/F1KM-S1.

(1) Detecting PE guard violation
If an external master makes an unauthorized access to the resource area in a PE for which PE guard is set, the access is detected as a PE guard violation.

(2) Blocking unauthorized access
When a PE guard violation is detected, unauthorized access to the internal resources of the PE are blocked to prevent unauthorized modification of the contents of PE resources.

(3) Notifying occurrence of violation
An error response to an unauthorized access is sent to the request source of external master. When DMA Controller makes an unauthorized access, meanwhile, a DMA transfer error is detected.
A PE guard violation is notified as INTGUARD interrupt request which is a source of FEINT.

(2) Protection Made by SPID

- Setting PEG Protection
  - Up to four areas can be set depending on the local RAM address*[1] of the own PE.
  - The area range is specified by the base address and the mask bit (4 kbytes to 4 Gbytes).
  - “Read enable” and “write enable” can be set for each area.
  - “Enable” or “disable” can be selected based on the system protection identifier (SPID) for each area.

Note 1. It is the local RAM address and the retention RAM address in RH850/F1KM-S1.

- Procedure for permitting access by using the system protection identifier (SPID)

1. Is the area subject to access is the local RAM area*[1]? If so, go to step 2.
2. Is the area subject to access is within the range of valid areas 0, 1, 2, or 3? If so, go to step 3.
   Otherwise, return an error response.
3. Are all the conditions below for the relevant area satisfied? If so, permit access.
   - The system protection identifier (SPID) is enabled.
   - Required operations (read/write) are enabled.

   Otherwise, return an error response.

Note 1. It is the local RAM area and the retention RAM area in RH850/F1KM-S1.

(3) List of PEG Protection Setting Registers

Specify the necessary settings in the registers below to protect PE resources from unauthorized access by an external master.

- Whether to permit access to the local RAM*[1] in the PE can be specified.

Note 1. It is local RAM and retention RAM in RH850/F1KM-S1.

Table 3BC.59 PEG Registers (Base Address: FFFE E600$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after Reset
							1	8	16	32	
PEG	+00C$_H$	4	PEG SPID control register	PEGSP	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+080$_H$	4	PEG area 0 mask setting register	PEGG0MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+084$_H$	4	PEG area 0 base setting register	PEGG0BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+090$_H$	4	PEG area 1 mask setting register	PEGG1MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+094$_H$	4	PEG area 1 base setting register	PEGG1BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0A0$_H$	4	PEG area 2 mask setting register	PEGG2MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0A4$_H$	4	PEG area 2 base setting register	PEGG2BA	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0B0$_H$	4	PEG area 3 mask setting register	PEGG3MK	—	R/W	—	✓	✓	✓	0000 0000$_H$
	+0B4$_H$	4	PEG area 3 base setting register	PEGG3BA	—	R/W	—	✓	✓	✓	0000 0000$_H$

(4) Register Set

(a) PEGSP — PEG SPID Control Register

Access:	PEGSP register can be read or written in 32-bit units. PEGSPL register can be read or written in 16-bit units. PEGSPLL register can be read or written in 8-bit units.
Address:	PEGSP: FFFE E60C$_H$ PEGSPL: FFFE E60C$_H$ PEGSPLL: FFFE E60C$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3BC.60 PEGSP Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SPEN	Access permission to external master with specified SPID. 0: Not permit. 1: Permit.

(b) PEGGnMK — PEG Area n Mask Setting Register (n = 0 to 3)

The PEGGnMK register defines which bits of PEGGnBA.GnBASE are compared with the access address. If bit PEGGnMK.GnMASK[m] is cleared, bit PEGGnBA.GnBASE[m] is compared with bit m of the access address.

Access: PEGGnMK register can be read or written in 32-bit units.
PEGGnMKL, PEGGnMKH registers can be read or written in 16-bit units.
PEGGnMKLH, PEGGnMKHL, PEGGnMKHH registers can be read or written in 8-bit units.

Address: PEGGnMK: FFFE $E680_H$ + (10_H × n)
PEGGnMKL: FFFE $E680_H$ + (10_H × n),
PEGGnMKH: FFFE $E682_H$ + (10_H × n)
PEGGnMKLH: FFFE $E681_H$ + (10_H × n),
PEGGnMKHL: FFFE $E682_H$ + (10_H × n),
PEGGnMKHH: FFFE $E683_H$ + (10_H × n)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnMASK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnMASK				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 3BC.61 PEGGnMK Register Contents

Bit Position	Bit Name	Function
31 to 12	GnMASK	0: Target bits are compared when determining the PE guard area. 1: Target bits are not compared when determining the PE guard area.
11 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

NOTE

When you write to the PEGGnMK register, the corresponding GnEN bit in the PEGGnBA register is cleared automatically.

(c) PEGGnBA — PEG Area n Base Setting Register (n = 0 to 3)

In combination with the PEGGnMK register, this register specifies a range or ranges within PE guard protection area n. Setting the GnEN bit to 1 validates the access enable conditions specified by this register and the PEGGnMK register.

Access: PEGGnBA register can be read or written in 32-bit units.
PEGGnBAL, PEGGnBAH registers can be read or written in 16-bit units.
PEGGnBALL, PEGGnBALH, PEGGnBAHL, PEGGnBAHH registers can be read or written in 8-bit units.

Address: PEGGnBA: FFFE E684$_H$ + (10$_H$ × n)
PEGGnBAL: FFFE E684$_H$ + (10$_H$ × n),
PEGGnBAH: FFFE E686$_H$ + (10$_H$ × n)
PEGGnBALL FFFE E684$_H$ + (10$_H$ × n),
PEGGnBALH FFFE E685$_H$ + (10$_H$ × n),
PEGGnBAHL: FFFE E686$_H$ + (10$_H$ × n),
PEGGnBAHH: FFFE E687$_H$ + (10$_H$ × n)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnBASE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnBASE				—	—	—	—	GnSP3	GnSP2	GnSP1	GnSP0	—	GnWR	GnRD	GnEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 3BC.62 PEGGnBA Register Contents

Bit Position	Bit Name	Function
31 to 12	GnBASE	Base address that specifies the range of PE guard protection area n.
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	GnSP3	Access permission setting from SPID = 3 external master to PE guard protection area n. 0: Not permit. 1: Permit.
6	GnSP2	Access permission setting from SPID = 2 external master to PE guard protection area n. 0: Not permit. 1: Permit.
5	GnSP1	Access permission setting from SPID = 1 (CPU1*[1]) external master to PE guard protection area n. 0: Not permit. 1: Permit.
4	GnSP0	Access permission setting from SPID = 0 (peripheral device connected to H-BUS*[2]) external master to PE guard protection area n. 0: Not permit. 1: Permit.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	GnWR	Write access permission to PE guard protection area n. 0: Write access is disabled. 1: Write access is enabled.

Table 3BC.62 PEGGnBA Register Contents

Bit Position	Bit Name	Function
1	GnRD	Read access permission to PE guard protection area n. 0: Read access is disabled. 1: Read access is enabled.
0	GnEN	Enables or Disables the setting for the access enable conditions to PE guard protection area n. 0: Settings for access enable conditions are disabled. 1: Settings for access enable conditions are enabled.

Note 1. Setting value of MCFG0.SPID

Note 2. H-BUS is only supported by RH850/F1KM-S4.

NOTE

When you write to the PEGGnMK register, the corresponding GnEN bit in the PEGGnBA register is cleared automatically.

3BC.2.3.2 PE’s Internal Peripheral Device Protection Function (IPG)

(1) Overview of the IPG Function

The IPG is a function to prevent unauthorized accesses to peripheral devices from the CPU core equipped with the IPG. The IPG achieves the following functions. The IPG covers accesses to the SEG, the PEG, the INTC1 and P-Bus.

(a) Detecting Violation of Peripheral Device Protection

If the CPU makes an unauthorized access to an area (peripheral device) for which peripheral device protection is set, the access is detected as “violation of peripheral device protection”.

(b) Storing Unauthorized Access Information

When a violation of peripheral device protection is detected, the unauthorized-access information is stored in the IPG’s internal register.

(c) Blocking Unauthorized Accesses

When a violation of peripheral device protection is detected, unauthorized accesses to peripheral devices are blocked to prevent contents of peripheral devices from being modified illegally.

(d) Notifying Violation

When a violation of peripheral device protection is detected, a request for generating an exception is made to ask the CPU to stop the processing.

NOTE

Even if a request for generating an exception is immediately sent to the CPU in step **(d) Notifying Violation** above, a subsequent access issued (before receiving a request from the IPG) by the CPU that does not know an occurrence of violation may illegally modify contents of peripheral devices. (Accesses after a violation has occurred result in unauthorized accesses.)

(2) IPG Function

- This function invalidates accesses according to their attributes (including address, transfer type, and access right).
- After an access right violation is detected until the error flag (described later) is cleared by writing by the software, subsequent accesses are invalidated. However, invalidation is applied only to accesses from the CPU and is not applied to accesses from outside the CPU core. Invalidation is performed independently of addresses.
- When a request for accessing different peripheral devices simultaneously is made due to misalignment or double-word access, the access is executed when all such accesses are enabled.

(3) IPG Protection Setting Registers for Illegal Users

To protect peripheral devices from unauthorized accesses by programs in user mode, necessary settings are required for the registers listed below.

- Accesses in user mode are to be detected.

Table 3BC.63 IPG Registers (Base Address: FFFE E000$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right*1	R/W	Operable Bit				Value after Reset
							1	8	16	32	
IPG	+002$_H$	2	Peripheral device protection violation access information register	IPGECRUM	SV	R/W	—	✓	✓	—	Undefined (retained)
	+008$_H$	4	Peripheral device protection violation access address register	IPGADRUM	SV	R/W	—	✓	✓	✓	Undefined (retained)
	+00D$_H$	1	Peripheral device protection enable register	IPGENUM	SV	R/W	—	✓	—	—	00$_H$
	+020$_H$	1	Peripheral device protection setting register 0	IPGPMTUM0	SV	R/W	—	✓	—	—	00$_H$
	+022$_H$	1	Peripheral device protection setting register 2	IPGPMTUM2	SV	R/W	—	✓	—	—	00$_H$
	+023$_H$	1	Peripheral device protection setting register 3	IPGPMTUM3	SV	R/W	—	✓	—	—	00$_H$
	+024$_H$	1	Peripheral device protection setting register 4	IPGPMTUM4	SV	R/W	—	✓	—	—	00$_H$

Note 1. Registers for which “SV” is described are accessible by accesses with SV right (UM = 0).

(4) Register Set

(a) IPGECRUM — Peripheral Device Protection Violation Access Information Register

Access:	IPGECRUM register can be read or written in 16-bit units. IPGECRUML register can be read or written in 8-bit units.
Address:	IPGECRUM: FFFE E002$_H$ IPGECRUML: FFFE E002$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WD	HW	BY	EX	WR	RD	VD
Value after reset	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3BC.64 IPGECRUM Register Contents

Bit Position	Bit Name	Function
15 to 7	Reserved	These bits are always read as 0. The write value should always be 0.
6	WD	This bit is set to 1 when a violation occurred in read word, instruction fetch read access, write word, CAXI, LDL or STC. In other cases, this bit is cleared to 0.
5	HW	This bit is set to 1 when a violation occurred in read halfword or write haflword. In other cases, this bit is cleared to 0.
4	BY	This bit is set to 1 when a violation occurred in read byte, write byte or bit operation. In other cases, this bit is cleared to 0.
3	EX	This bit is set to 1 when a violation occurred in an instruction fetch read access. In other cases, this bit is cleared to 0.
2	WR	This bit is set to 1 when a violation occurred in a write access, bit operation, or execution of the CAXI instruction. In other cases, this bit is cleared to 0.
1	RD	This bit is set to 1 when a violation occurred in a read access, bit operation, or execution of the CAXI instruction. In other cases, this bit is cleared to 0.
0	VD	This bit is set to 1 when a violation of peripheral device protection is detected by a program with the relevant right. Even if another violation of peripheral device protection is detected while this bit is 1, data of this IPGECRUM register and the IPGADRUM register is not updated and is retained.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(b) IPGADRUM — Peripheral Device Protection Violation Access Address Register

Access: IPGADRUM register can be read or written in 32-bit units.

IPGADRUML, IPGADRUMH registers can be read or written in 16-bit units.

IPGADRUMLL, IPGADRUMLH, IPGADRUMHL, IPGADRUMHH registers can be read or written in 8-bit units.

Address: IPGADRUM: FFFE E008$_H$

IPGADRUML: FFFE E008$_H$, IPGADRUMH: FFFE E00A$_H$

IPGADRUMLL: FFFE E008$_H$, IPGADRUMLH: FFFE E009$_H$,

IPGADRUMHL: FFFE E00A$_H$, IPGADRUMHH: FFFE E00B$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EADR[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3BC.65 IPGADRUM Register Contents

Bit Position	Bit Name	Function
31 to 0	EADR	These bits store the address of the access in which a violation occurred.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(c) IPGENUM — Peripheral Device Protection Enable Register

Access: IPGENUM register can be read or written in 8-bit units.

Address: IPGENUM: FFFE E00D$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IRE	E
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3BC.66 IPGENUM Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	IRE	This bit sets whether to store the access information in the peripheral device protection violation access address register and the peripheral device protection violation access information register when a violation of peripheral device protection occurred in an instruction fetch access. 0: Instruction fetch access information is not stored. (value after reset) 1: Instruction fetch access information is stored. **CAUTION:** If you do not want to detect speculative instruction fetches (no instruction is executed in some cases), clear this bit to 0.
0	E	This bit enables or disables the peripheral devices protection function against accesses by the relevant access right. 0: The peripheral device protection function is disabled. (Value after reset) 1: The peripheral device protection function is enabled.

(d) IPGPMTUM0 — Peripheral Device Protection Setting Register 0

Access: IPGPMTUM0 register can be read or written in 8-bit units.

Address: IPGPMTUM0: FFFE E020$_H$

Bit	7	6	5	4	3	2	1	0
	—	X1	W1	R1	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R

Table 3BC.67 IPGPMTUM0 Register Contents

Bit Position	Bit Name	Function
7	Reserved	These bits are always read as 0. The write value should always be 0.
6	X1	This bit sets whether to enable instruction fetch read access to P-Bus. 0: Instruction fetch read access to P-Bus is treated as violation. (Value after reset) 1: Instruction fetch read access to P-Bus is not restricted.
5	W1	This bit sets whether to enable write access to P-Bus. 0: Write access to P-Bus is treated as violation. (Value after reset) 1: Write access to P-Bus is not restricted.
4	R1	This bit sets whether to enable read access to P-Bus. 0: Read access to P-Bus is treated as violation. (Value after reset) 1: Read access to P-Bus is not restricted.
3 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

(e) IPGPMTUM2 — Peripheral Device Protection Setting Register 2

Access: IPGPMTUM2 register can be read or written in 8-bit units.

Address: IPGPMTUM2: FFFE E022$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	W0	R0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3BC.68 IPGPMTUM2 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	W0	This bit sets whether to enable write access to INTC1. 0: Write access to INTC1 is treated as violation. (Value after reset) 1: Write access to INTC1 is not restricted
0	R0	This bit sets whether to enable read access to INTC1. 0: Read access to INTC1 is treated as violation. (Value after reset) 1: Read access to INTC1 is not restricted.

(f) IPGPMTUM3 — Peripheral Device Protection Setting Register 3

Access: IPGPMTUM3 register can be read or written in 8-bit units.

Address: IPGPMTUM3: FFFE E023$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	W1	R1	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 3BC.69 IPGPMTUM3 Register Contents

Bit Position	Bit Name	Function
7 to 6	Reserved	These bits are always read as 0. The write value should always be 0.
5	W1	This bit sets whether to enable write access to SysErrGen. 0: Write access to SysErrGen is treated as violation. (Value after reset) 1: Write access to SysErrGen is not restricted.
4	R1	This bit sets whether to enable read access to SysErrGen. 0: Read access to SysErrGen is treated as violation. (Value after reset) 1: Read access to SysErrGen is not restricted
3 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

(g) IPGPMTUM4 — Peripheral Device Protection Setting Register 4

Access: IPGPMTUM4 register can be read or written in 8-bit units.

Address: IPGPMTUM4: FFFE $E024_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	W0	R0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 3BC.70 IPGPMTUM4 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	W0	This bit sets whether to enable write access to its own PEG. 0: Write access to its own PEG is treated as violation. (Value after reset) 1: Write access to its own PEG is not restricted.
0	R0	This bit sets whether to enable read access to its own PEG. 0: Read access to its own PEG is treated as violation. (Value after reset) 1: Read access to its own PEG is not restricted

3BC.2.3.3 System Error Generator Function (SEG)

SEG (SysErrGen) controls interrupt notification and recording after a system error occurred by a data access.

Multiple error occurrence inputs are categorized according to error factor, and are processed sequentially from the highest-priority error factor, generating an FE-level exception (SYSERR).

The bit position of the SEGFLAG register becomes the error factor priority. Error factors of lower bits take precedence over error factors of upper bits.

Error address information is recorded only once regardless of error frequency.

The error with the highest priority among the error factors is valid when errors occur simultaneously. Recorded error address information is not overwritten by subsequent errors.

(1) List of SEG Function Control Registers

Table 3BC.71 SEG Register (Base Address: FFFE E980$_H$)

Module Name	Address Offset	Size (Byte)	Register Name	Abbreviation	Right*1	R/W	Operable Bit				Value after Reset
							1	8	16	32	
SEG	+00$_H$	2	SEG error control register	SEGCONT	SV	R/W	—	—	✓	—	0000$_H$
	+02$_H$	2	SEG error flag register	SEGFLAG	SV	R/W	—	—	✓	—	0000$_H$
	+08$_H$	4	SEG error address information register	SEGADDR	SV	R/W	—	—	—	✓	Undefined (retained)

Note 1. Registers for which “SV” is described are writable with the SV right (UM = 0). Attempting to write, if these conditions do not hold, leads to a SYSERR exception with setting VCIF flag. No restriction is provided for read accesses.

(2) Register Set

(a) SEGCONT — SEG Error Control Register

This register is used to enable (= 1) or disable (= 0) notification of SysErr request in response to error flags that store the error occurrence status for each factor.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VCSE	APIE	IPGE	—	TCME	—	VCIE	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R	R	R	R

Table 3BC.72 SEGCONT Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	VCSE	This bit enables notification of an error response detected inside system interconnect: ● Error response from external bus masters in write access ● Illegal response to local RAM or peripherals or Flash from optional master in access (except instruction fetch from CPU). And illegal response to optional slave from external AHB master in access.
9	APIE	This bit enables notification of an error response from peripherals. The error notification includes the following cases: ● Error response from peripherals in write access ● PBG error in write access
8	IPGE	This bit enables notification of IPG illegal access detection.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCME	This bit enables notification of an error during data access to its own local RAM*[1] from PE master. The error notification includes the following cases: ● ECC uncorrectable error (DED or SED & SECDIS=1) ● Detection of an access to RAM unimplemented area
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 3BC.72 SEGCONT Register Contents

Bit Position	Bit Name	Function
4	VCIE	RH850/F1KM-S4: This bit enables notification of an error response during access to CPU peripherals, P-Bus (read access), global RAM, retention RAM, H-Bus and CodeFlash by PE. The error notification includes the following cases: ● IPG error from CPU peripherals and P-Bus ● Error response from H-Bus peripherals ● PBG error / HBG error from P-Bus and H-Bus ● GRG error from global RAM and retention RAM ● ECC uncorrectable error from CodeFlash, global RAM and retention RAM (DED or SED & SECDIS = 1) This bit enables notification of an error response when accessing to a part of access prohibited areas in address map. RH850/F1KM-S1: This bit enables notification of an error response during access to CPU peripherals, P-Bus (read access), global RAM and CodeFlash by PE. The error notification includes the following cases: ● IPG error from CPU peripherals and P-Bus ● PBG error from P-Bus ● ECC uncorrectable error from CodeFlash (DED or SED & SECDIS = 1) This bit enables notification of an error response when accessing to a part of access prohibited areas in address map.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. It is local RAM and retention RAM in RH850/F1KM-S1.

(b) SEGFLAG — SEG Error Flag Register

This register indicates error flags that store error occurrence status of each factors. The flags are set to 1 by an error occurrence input. The flags are not automatically cleared to 0. Both setting and clearing of each flag are supported in writing to the register.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VCSF	APIF	IPGF	—	TCMF	—	VCIF	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R	R	R	R

Table 3BC.73 SEGFLAG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	VCSF	Flag corresponding to bit 10 of the SEGCONT register
9	APIF	Flag corresponding to bit 9 of the SEGCONT register
8	IPGF	Flag corresponding to bit 8 of the SEGCONT register
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCMF	Flag corresponding to bit 6 of the SEGCONT register
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	VCIF	Flag corresponding to bit 4 of the SEGCONT register
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

NOTE

An error may lead to setting of multiple error flags in SEG. For example, if an IPG error occurs at peripheral registers read, both IPGF bit and VCIF bit in SEGFLAG are set.

(c) SEGADDR — Error Address Information Register

Address information (one record) which is notified with error occurrence is stored in the register. The register is not updated while one or more bits in SEGFLAG register are set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Address[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Address[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3BC.74 SEGADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	Address	These bits store the error address information.

CAUTIONS

1. SEGADDR stores error address information in case of an error occurrence related to VCIF bit or TCMF bit in SEGFLAG register. SEGADDR register stores all 0 data in case of an error occurrence related to VCSF bit, APIF bit or IPGF bit in SEGFLAG register.
2. In case of an error occurrence related to TCMF bit in SEGFLAG register, bit[18:0] of the error address are stored in SEGADDR[18:0] and SEGADDR[31:19] are filled with 0.

(3) SEG Function

(a) SEG Function: SYSERR Request Notification by Error Flag

- Setting an error flag takes precedence over clearing the same flag.
  - A simultaneous clearing operation is ignored.
- Priority of error factors
  - The bit position of each flags in SEGFLAG register which error notification is enabled by SEGCONT register becomes the error factor priority. Error factors of lower bits take precedence over error factors of upper bits. Notification is made from the highest-priority error factor.
  - The bit position of error factors is reported as a “SysErr factor code.”
- Conditions for starting SysErr request notification
  - Even if a flag which error notification is disabled by SEGCONT register is set to 1, notification is not made.
  - Notification is made immediately after a flag which error notification is enabled by SEGCONT register is set to 1.
  - After clearing of a flag, notification is made if an other flag which error notification is enabled by SEGCONT register remains set.
- Finishing notification at a SysErr acknowledgement
  - Even after notification is finished, the flag is not cleared automatically.
  - Notification is not made until setting or clearing the flag again.
  - If an error flag that is prioritized higher than the error factor is set prior to an acknowledgement, the notification information may be replaced with a higher prioritized SysErr factor code.

(b) SEG Function: Recording Error Address Information

- When an error which error notification is enabled by SEGCONT register occurs, the error address is retained in the SEGADDR register.
  - No information is retained by setting or clearing an error flag in SEGFLAG register.
  - When multiple error occurrence inputs are present simultaneously, information other than the prioritized error factor is not retained.
- While a flag which error notification is enabled by SEGCONT register is set to 1, overwrite to the SEGADDR register is inhibited.
  - If error occurrence input continues, information of subsequent error factors is not retained.
  - To reset the inhibition of overwrite to the register, clear either SEGCONT or SEGFLAG register (or both of them).

(c) Supplementary Notes on SYSERR Exception

- Even when a SYSERR exception occurs, the value of the PSW.EBV bit is held, and the base address of the exception handler does not switch.

3BC.3 Notes

3BC.3.1 Synchronization of Store Instruction Completion and Subsequent Instruction Execution

When a control register is updated by a store instruction, there is a time lag after the CPU executes the store instruction and before the control register is actually updated. Therefore, if the updated content of the control register is to be used by the instruction following the store instruction, the appropriate synchronization is required. How to perform synchronization processing is shown below.

For the procedures to synchronize updating system registers by LDSR instruction and the subsequent instruction execution, see *APPENDIX A, Hazard Resolution Procedure for System Registers* in *the RH850G3KH User's Manual: Software*.

When the updated results in the control registers are to be used by the subsequent instruction:

Example 1: An interrupt is enabled by execution of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits.

Proceed as follows in this case.

(1) Execute the store instruction to update a control register (ST.W, etc.).

(2) Perform a dummy read of the above control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute the subsequent instruction (EI).

In case of RH850/F1KM, SYNCM instruction has the same effects as above-mentioned (2) and (3). (Excludes RCFDCn, RCFDCn ECC register access.)

Example 2: When you must wait until a control register (control register A) has been completely updated before accessing another control register (control register B), execute similar processing. For example, different peripheral functions are linked, or the interrupt mask for INTC is cleared after the peripheral function is set. Note that this processing is not required if the control registers A and B belong to the same peripheral group.

(1) Execute the store instruction to update the control register A (ST.W, etc.).

(2) Perform a dummy read of the above control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute the store instruction to access the control register B (ST.W, LD.W, etc.).

In case of RH850/F1KM, SYNCM instruction has the same effects as above-mentioned (2) and (3). (Excludes RCFDCn, RCFDCn ECC register access.)

The similar processing is also required when starting to access a memory or control register to be protected is started after a safety function (such as some kind of memory protection and ECC) has been completely set up.

When the updated results of the control register or memory to be used in the instruction fetch of the subsequent instruction:

(a) In case of writing the instructions to the RAM before jumping to the RAM to execute instructions from the RAM, take the following procedure.

(1) Execute the store instruction to update a memory (ST.W, etc.).

(2) Perform a dummy read of the above memory (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute SYNCI.

(5) Execute the subsequent instruction (branch instruction, etc.).

(b) In case of updating control registers for memory protection and ECC functions before jumping to the memory to be controlled by the registers, take the following procedure.

(1) Execute the store instruction to update a control register (ST.W, etc.).

(2) Perform a dummy read of the control register (LD.W, etc.).

(3) Execute SYNCP.

(4) Execute SYNCI.

(5) Execute the subsequent instruction (branch instruction, etc.).

When switching the code flash memory area:

In this case, see *Section 10, Usage Notes, (7) Updating the BFASELR register* in *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

3BC.3.2 Ensure Coherency after Rewriting the Code Flash

The CPU1 is equipped with the buffer for the code flash area as described in **Section 3BC.2.2, Buffers for Code Flash**.

Therefore, clear the buffer to ensure coherency after rewriting the code flash by self-programming.

3BC.3.3 Access to Registers by Using Bit-Manipulation Instructions

Writing bit-manipulation instructions consists of read-modify-write processing in 8-bit units. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. If a register includes multiple flag bits, the read-modify-write operation may lead to the clearing of flags that were not actually targets for clearing.

3BC.3.4 Caution of Prefetching

There is a possibility that the reading of the memory occurs by pre-fetch from the area where instruction codes do not exist. Secure more than 40-byte initialized area after the area where instruction codes are stored.

3BC.3.5 Overwriting Context upon Acceptance of Multiple Exceptions

Acceptance of an exception depends on the type of exception source, regardless of the states of the ID and NP bits in the PSW register. When multiple exceptions are generated, the contents of the system register which hold the context information are overwritten. For the conditions for acceptance and whether correct return or recovery is possible for each exception source, see *the List of Exception Sources* in *the RH850G3KH User's Manual: Software*.

Section 4A Address Space of RH850/F1KH-D8

4A.1 Address Space

Table 4A.1, Address Space (8-MB 176/233/324-Pin Product) to **Table 4A.2, Address Space (6-MB 176/233/324-Pin Product)** show the address space of the RH850/F1KH.

CAUTION

Do not access an address with which no register is mapped in the on-chip I/O register space. In addition, do not access any access prohibited area specified in **Table 4A.1, Address Space (8-MB 176/233/324-Pin Product)** to **Table 4A.2, Address Space (6-MB 176/233/324-Pin Product)**. If such an address is accessed, operation is not guaranteed.

NOTE

The Local RAM is accessible through the following three address areas in the address space.
CPU1 area: Address area accessible from CPU1, CPU2, DMA, FlexRay and ETNB.
CPU2 area: Address area accessible from CPU1, CPU2, DMA, FlexRay and ETNB.
Self area: Mirrored address area, accessible only from CPU (CPU1, CPU2) to refer the CPU’s self resource.

Table 4A.1 Address Space (8-MB 176/233/324-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 003F FFFF$_H$	Code Flash (bank A)	4 MB
0040 0000$_H$ to 007F FFFF$_H$	Access prohibited area	
0080 0000$_H$ to 00BF FFFF$_H$	Code Flash (bank B)	4 MB
00C0 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area	
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	1 KB
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area	
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area(CS0)	16 MB^{*4}
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area	
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area(CS1)	16 MB^{*4}
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area	
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area(CS2)	16 MB^{*4}
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area	
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area(CS3)	16 MB^{*4}
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FE9C FFFF$_H$	Access prohibited area	
FE9D 0000$_H$ to FE9F FFFF$_H$	Local RAM (CPU2 area)	192 KB^{*1}
FEA0 0000$_H$ to FEBC FFFF$_H$	Access prohibited area	
FEBD 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	192 KB^{*1}

Table 4A.1 Address Space (8-MB 176/233/324-Pin Product)

Address	Address Space Type	Size
FEC0 0000_H to FEDC $FFFF_H$	Access prohibited area	
FEDD 0000_H to FEDF $FFFF_H$	Local RAM (self area)	192 KB*[1]
FEE0 0000_H to FEEB $7FFF_H$	Access prohibited area	
FEEB 8000_H to FEEF $FFFF_H$	Global RAM A	288 KB*[2]
FEF0 0000_H to FEF0 $FFFF_H$	Retention RAM	64 KB
FEF1 0000_H to FEFB $7FFF_H$	Access prohibited area	
FEFB 8000_H to FEFF $FFFF_H$	Global RAM B	288 KB*[2]
FF00 0000_H to FF1F $FFFF_H$	Access prohibited area	
FF20 0000_H to FF23 $FFFF_H$	Data flash	256 KB*[3]
FF24 0000_H to FF9F $FFFF_H$	Access prohibited area	
FFA0 0000_H to FFFD $FFFF_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000_H to FFFE $DFFF_H$	Access prohibited area	
FFFE $E000_H$ to FFFE $FFFF_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000_H to FFFF $4FFF_H$	Access prohibited area	
FFFF 5000_H to FFFF $FFFF_H$	On-chip peripheral I/O area	44 KB

Note 1. 160 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 256 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Note 4. 8 MB in 176/233-pin products: For details, see **Section 16, External Memory Access Controller (MEMC)**.

Table 4A.2 Address Space (6-MB 176/233/324-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 002F FFFF$_H$	Code Flash (bank A)	3 MB
0030 0000$_H$ to 007F FFFF$_H$	Access prohibited area	
0080 0000$_H$ to 00AF FFFF$_H$	Code Flash (bank B)	3 MB
00B0 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area	
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	1 KB
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area	
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area (CS0)	16 MB*4
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area	
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area (CS1)	16 MB*4
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area	
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area (CS2)	16 MB*4
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area	
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area (CS3)	16 MB*4
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FE9D 7FFF$_H$	Access prohibited area	
FE9D 8000$_H$ to FE9F FFFF$_H$	Local RAM (CPU2 area)	160 KB*1
FEA0 0000$_H$ to FEBD 7FFF$_H$	Access prohibited area	
FEBD 8000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	160 KB*1
FEC0 0000$_H$ to FEDD 7FFF$_H$	Access prohibited area	
FEDD 8000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	160 KB*1
FEE0 0000$_H$ to FEEB FFFF$_H$	Access prohibited area	
FEEC 0000$_H$ to FEEF FFFF$_H$	Global RAM A	256 KB*2
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFB FFFF$_H$	Access prohibited area	
FEFC 0000$_H$ to FEFF FFFF$_H$	Global RAM B	256 KB*2
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF23 FFFF$_H$	Data flash	256 KB*3
FF24 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. 160 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 48 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Note 4. 8 MB in 176/233-pin products: For details, see **Section 16, External Memory Access Controller (MEMC)**.

4A.2 Address Space Viewed from Each Bus Master

Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product) shows address spaces viewed from each bus master.

4A.2.1 Space in which Instructions can be Fetched

Instructions of the CPU can be fetched from the Code flash, local RAM, global RAM, and retention RAM.

4A.2.2 Data Space Accessible by CPU1

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from the CPU1.

4A.2.3 Data Space Accessible by CPU2

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from the CPU2.

4A.2.4 Data Space Accessible by DMA

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from the DMA.

4A.2.5 Data Space Accessible by Flexray

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from the Flexray.

4A.2.6 Data Space Accessible by ETNB

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from the ETNB.

4A.2.7 Data Space Accessible by Each Bus Master

See **Table 4A.3, Address Space Viewed from Each Bus Master (8-MB Product)** for the spaces accessible from Each Bus Master.

Table 4A.3 Address Space Viewed from Each Bus Master (8-MB Product)

Address	Resource	From CPU1	From CPU2	From DMA	From FlexRay	From ETNB
0000 0000$_H$ to 003F FFFF$_H$	Code Flash (bank A)	✓	✓	✓		
0040 0000$_H$ to 007F FFFF$_H$	Access prohibited area					
0080 0000$_H$ to 00BF FFFF$_H$	Code Flash (bank B)	✓	✓	✓		
00C0 0000$_H$ to 00FF FFFF$_H$	Access prohibited area					
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	✓	✓	✓		
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area					
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	✓	✓	✓		
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area					
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	✓	✓	✓		
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area					
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	✓	✓	✓		
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area					
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area (CS0)	✓	✓	✓		
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area					
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area (CS1)	✓	✓	✓		
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area					
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area (CS2)	✓	✓	✓		
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area					
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area (CS3)	✓	✓	✓		
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area					
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	✓	✓	✓		
3400 0000$_H$ to FE9C FFFF$_H$	Access prohibited area					
FE9D 0000$_H$ to FE9F FFFF$_H$	Local RAM (CPU2 area)	✓	✓	✓	✓	✓
FEA0 0000$_H$ to FEBC FFFF$_H$	Access prohibited area					
FEBD 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	✓	✓	✓	✓	✓
FEC0 0000$_H$ to FEDC FFFF$_H$	Access prohibited area					
FEDD 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	✓	✓			
FEE0 0000$_H$ to FEEB 7FFF$_H$	Access prohibited area					
FEEB 8000$_H$ to FEEF FFFF$_H$	Global RAM A	✓	✓	✓	✓	✓
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	✓	✓	✓	✓	✓
FEF1 0000$_H$ to FEFB 7FFF$_H$	Access prohibited area					
FEFB 8000$_H$ to FEFF FFFF$_H$	Global RAM B	✓	✓	✓	✓	✓
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area					
FF20 0000$_H$ to FF23 FFFF$_H$	Data flash	✓	✓	✓		
FF24 0000$_H$ to FF9F FFFF$_H$	Access prohibited area					
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	✓	✓	✓		
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area					
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	✓	✓			
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area					
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	✓	✓	✓		

Note: The following color coding is used in the map above.

🟧 Fetch and data access available
🟩 Data access available
⬜ Access prohibited

4A.3 Peripheral I/O Address Map

Table 4A.4, Peripheral I/O Address Map shows peripheral I/O address map.

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FF00 0000_H to FF1F $FFFF_H$	—	Access prohibited area
FF20 0000_H to FF23 $FFFF_H$	4	Data Flash
FF24 0000_H to FF9F $FFFF_H$	—	Access prohibited area
FFA0 0000_H to FFA0 $001F_H$	1	FLMD
FFA0 0020_H to FFA0 $0FFF_H$	—	Access prohibited area
FFA0 1000_H to FFA0 $103F_H$	—	Access prohibited area
FFA0 1040_H to FFA0 $1FFF_H$	—	Access prohibited area
FFA0 2000_H to FFA0 $201F_H$	—	Access prohibited area
FFA0 2020_H to FFA0 $7FFF_H$	—	Access prohibited area
FFA0 8000_H to FFA0 $801F_H$	—	Access prohibited area
FFA0 8020_H to FFA0 $FFFF_H$	—	Access prohibited area
FFA1 0000_H to FFA1 $1FFF_H$	1	FPSYS (Register)
FFA1 2000_H to FFA1 $FFFF_H$	—	Access prohibited area
FFA2 0000_H to FFA2 $FFFF_H$	1	FACI command-issuing area
FFA3 0000_H to FFBF $FFFF_H$	—	Access prohibited area
FFC0 0000_H to FFC0 $000F_H$	1	FENMI (ECON_NMI)
FFC0 0010_H to FFC0 $00FF_H$	—	Access prohibited area
FFC0 0100_H to FFC0 $010F_H$	1	FEINT (ECON_FEINT)
FFC0 0110_H to FFC0 $0FFF_H$	—	Access prohibited area
FFC0 1000_H to FFC0 1003_H	1	SELB_INTC (SL_INTC)
FFC0 1004_H to FFC0 $1FFF_H$	—	Access prohibited area
FFC0 2000_H to FFC0 $200F_H$	1	SELB_DMAC (SL_DMAC)
FFC0 2010_H to FFC0 $2FFF_H$	—	Access prohibited area
FFC0 3000_H to FFC0 $300F_H$	1	GRZF
FFC0 3010_H to FFC0 $FFFF_H$	—	Access prohibited area
FFC1 0000_H to FFC1 $FFFF_H$	1	PORTn
FFC2 0000_H to FFC2 $FFFF_H$	1	PORT (JTAG)
FFC3 0000_H to FFC3 $000F_H$	1	DNFA_TAUD0 (DNF)
FFC3 0010_H to FFC3 $001F_H$	—	Access prohibited area
FFC3 0020_H to FFC3 $002F_H$	1	DNFA_TAUB0 (DNF)
FFC3 0030_H to FFC3 $003F_H$	—	Access prohibited area
FFC3 0040_H to FFC3 $004F_H$	1	DNFA_TAUB1 (DNF)
FFC3 0050_H to FFC3 $005F_H$	—	Access prohibited area
FFC3 0060_H to FFC3 $006F_H$	1	DNFA_ENCA0 (DNF)
FFC3 0070_H to FFC3 $009F_H$	—	Access prohibited area
FFC3 $00A0_H$ to FFC3 $00AF_H$	1	DNFA_ADCA0 (DNF)
FFC3 $00B0_H$ to FFC3 $00BF_H$	—	Access prohibited area
FFC3 $00C0_H$ to FFC3 $00CF_H$	1	DNFA_ADCA1 (DNF)
FFC3 $00D0_H$ to FFC3 $00DF_H$	—	Access prohibited area
FFC3 $00E0_H$ to FFC3 $00EF_H$	1	DNFA_SENT (DNF)
FFC3 $00F0_H$ to FFC3 $00FF_H$	—	Access prohibited area

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC3 0100$_H$ to FFC3 010F$_H$	—	Access prohibited area
FFC3 0110$_H$ to FFC3 3FFF$_H$	—	Access prohibited area
FFC3 4000$_H$ to FFC3 401F$_H$	1	FCLA_NMI (FCLA0)
FFC3 4020$_H$ to FFC3 403F$_H$	1	FCLA_INTPL (FCLA0)
FFC3 4040$_H$ to FFC3 405F$_H$	1	FCLA_INTPH (FCLA0)
FFC3 4060$_H$ to FFC3 407F$_H$	1	FCLA_ADCA0 (FCLA0)
FFC3 4080$_H$ to FFC3 409F$_H$	1	FCLA_ADCA1 (FCLA0)
FFC3 40A0$_H$ to FFC3 40BF$_H$	1	FCLA_INTPU (FCLA0)
FFC3 40C0$_H$ to FFC3 FFFF$_H$	—	Access prohibited area
FFC4 0000$_H$ to FFC4 004F$_H$	1	P-Bus guard (PBG10)
FFC4 0050$_H$ to FFC4 00FF$_H$	—	Access prohibited area
FFC4 0100$_H$ to FFC4 014F$_H$	1	P-Bus guard (PBG11)
FFC4 0150$_H$ to FFC4 03FF$_H$	—	Access prohibited area
FFC4 0400$_H$ to FFC4 044F$_H$	1	P-Bus guard (PBG12)
FFC4 0450$_H$ to FFC4 04FF$_H$	—	Access prohibited area
FFC4 0500$_H$ to FFC4 054F$_H$	1	P-Bus guard (PBG13)
FFC4 0550$_H$ to FFC4 05FF$_H$	—	Access prohibited area
FFC4 0600$_H$ to FFC4 064F$_H$	1	P-Bus guard (PBG14)
FFC4 0650$_H$ to FFC4 8FFF$_H$	—	Access prohibited area
FFC4 9000$_H$ to FFC4 907F$_H$	2	Global RAM Guard Bank A (MGDGR)
FFC4 9080$_H$ to FFC4 91FF$_H$	—	Access prohibited area
FFC4 9200$_H$ to FFC4 927F$_H$	2	Global RAM Guard Bank B (MGDGR)
FFC4 9280$_H$ to FFC4 BFFF$_H$	—	Access prohibited area
FFC4 C000$_H$ to FFC4 C00F$_H$	2	P-Bus guard (PBGC0)
FFC4 C010$_H$ to FFC4 C10F$_H$	—	Access prohibited area
FFC4 C110$_H$ to FFC4 C13F$_H$	2	P-Bus guard (PBGC1)
FFC4 C140$_H$ to FFC4 C7FF$_H$	—	Access prohibited area
FFC4 C800$_H$ to FFC4 C80F$_H$	2	ERRSLV (PBGC0)
FFC4 C810$_H$ to FFC4 C8FF$_H$	—	Access prohibited area
FFC4 C900$_H$ to FFC4 C90F$_H$	2	ERRSLV (PBGC1)
FFC4 C910$_H$ to FFC5 7FFF$_H$	—	Access prohibited area
FFC5 8000$_H$ to FFC5 803F$_H$	2	BOOTCTRL
FFC5 8040$_H$ to FFC5 97FF$_H$	—	Access prohibited area
FFC5 9800$_H$ to FFC5 981F$_H$	4	EEPRDCYCL (DCIB)
FFC5 9820$_H$ to FFC5 9BFF$_H$	—	Access prohibited area
FFC5 9C00$_H$ to FFC5 9C4F$_H$	4	P-Bus guard (PBG40)
FFC5 9C50$_H$ to FFC5 AFFF$_H$	—	Access prohibited area
FFC5 B000$_H$ to FFC5 B003$_H$	2	FBUFCCTL (FBUF_CTRL)
FFC5 B004$_H$ to FFC6 21FF$_H$	—	Access prohibited area
FFC6 2200$_H$ to FFC6 23FF$_H$	2	Code Flash ECC (CFECC_VCI)
FFC6 2400$_H$ to FFC6 25FF$_H$	2	Code Flash ECC (CFECC_CPU1)
FFC6 2600$_H$ to FFC6 29FF$_H$	—	Access prohibited area
FFC6 2A00$_H$ to FFC6 2A3F$_H$	4	Data Flash ECC (DFECC)
FFC6 2A40$_H$ to FFC6 3FFF$_H$	—	Access prohibited area

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC6 4000$_H$ to FFC6 403F$_H$	2	Global RAM ECC Bank A (GRECC)
FFC6 4040$_H$ to FFC6 41FF$_H$	—	Access prohibited area
FFC6 4200$_H$ to FFC6 423F$_H$	2	Global RAM ECC Bank B (GRECC)
FFC6 4240$_H$ to FFC6 4FFF$_H$	—	Access prohibited area
FFC6 5000$_H$ to FFC6 501F$_H$	2	Local RAM ECC TEST PE1 (LRTST)
FFC6 5020$_H$ to FFC6 53FF$_H$	—	Access prohibited area
FFC6 5400$_H$ to FFC6 547F$_H$	2	Local RAM ECC PE1 (LRECC)
FFC6 5480$_H$ to FFC6 55FF$_H$	—	Access prohibited area
FFC6 5600$_H$ to FFC6 567F$_H$	2	Local RAM ECC PE2 (LRECC)
FFC6 5680$_H$ to FFC7 00FF$_H$	—	Access prohibited area
FFC7 0100$_H$ to FFC7 011F$_H$	3	ECCCSIH0
FFC7 0120$_H$ to FFC7 01FF$_H$	—	Access prohibited area
FFC7 0200$_H$ to FFC7 021F$_H$	3	ECCCSIH1
FFC7 0220$_H$ to FFC7 02FF$_H$	—	Access prohibited area
FFC7 0300$_H$ to FFC7 031F$_H$	3	ECCCSIH2
FFC7 0320$_H$ to FFC7 03FF$_H$	—	Access prohibited area
FFC7 0400$_H$ to FFC7 041F$_H$	3	ECCCSIH3
FFC7 0420$_H$ to FFC7 04FF$_H$	—	Access prohibited area
FFC7 0500$_H$ to FFC7 051F$_H$	3	ECCCSIH4
FFC7 0520$_H$ to FFC7 12FF$_H$	—	Access prohibited area
FFC7 1300$_H$ to FFC7 131F$_H$	5	ECCCFD0MB
FFC7 1320$_H$ to FFC7 13FF$_H$	—	Access prohibited area
FFC7 1400$_H$ to FFC7 141F$_H$	5	ECCCFD0AFL0
FFC7 1420$_H$ to FFC7 14FF$_H$	—	Access prohibited area
FFC7 1500$_H$ to FFC7 151F$_H$	5	ECCCFD0AFL1
FFC7 1520$_H$ to FFC7 17FF$_H$	—	Access prohibited area
FFC7 1800$_H$ to FFC7 184F$_H$	5	P-Bus guard (PBG60)
FFC7 1850$_H$ to FFC7 18FF$_H$	—	Access prohibited area
FFC7 1900$_H$ to FFC7 194F$_H$	5	P-Bus guard (PBG61)
FFC7 1950$_H$ to FFC7 19FF$_H$	—	Access prohibited area
FFC7 1A00$_H$ to FFC7 1A1F$_H$	5	ECCCFD1MB
FFC7 1A20$_H$ to FFC7 1AFF$_H$	—	Access prohibited area
FFC7 1B00$_H$ to FFC7 1B1F$_H$	5	ECCCFD1AFL0
FFC7 1B20$_H$ to FFC7 1BFF$_H$	—	Access prohibited area
FFC7 1C00$_H$ to FFC7 1C1F$_H$	5	ECCCFD1AFL1
FFC7 1C20$_H$ to FFC7 30FF$_H$	—	Access prohibited area
FFC7 3100$_H$ to FFC7 311F$_H$	3	ECCFLXA0
FFC7 3120$_H$ to FFC7 31FF$_H$	—	Access prohibited area
FFC7 3200$_H$ to FFC7 321F$_H$	3	ECCFLXA0T0
FFC7 3220$_H$ to FFC7 32FF$_H$	—	Access prohibited area
FFC7 3300$_H$ to FFC7 331F$_H$	3	ECCFLXA0T1
FFC7 3320$_H$ to FFC7 40FF$_H$	—	Access prohibited area
FFC7 4100$_H$ to FFC7 411F$_H$	3	ECCETNB0TX
FFC7 4120$_H$ to FFC7 41FF$_H$	—	Access prohibited area

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC7 4200$_H$ to FFC7 421F$_H$	3	ECCETNB0RX
FFC7 4220$_H$ to FFC7 42FF$_H$	—	Access prohibited area
FFC7 4300$_H$ to FFC7 431F$_H$	3	ECCETNB1TX
FFC7 4320$_H$ to FFC7 43FF$_H$	—	Access prohibited area
FFC7 4400$_H$ to FFC7 441F$_H$	3	ECCETNB1RX
FFC7 4420$_H$ to FFC7 7FFF$_H$	—	Access prohibited area
FFC7 8000$_H$ to FFC7 8003$_H$	1	SELB READ TEST (SL_READTEST)
FFC7 8004$_H$ to FFC9 FFFF$_H$	—	Access prohibited area
FFCA 0000$_H$ to FFCA 007F$_H$	1	RIIC0
FFCA 0080$_H$ to FFCA 00FF$_H$	1	RIIC1
FFCA 0100$_H$ to FFCC FFFF$_H$	—	Access prohibited area
FFCD 0000$_H$ to FFCD 01FF$_H$	1	PRDNAME/CHIPID (SCDS)
FFCD 0200$_H$ to FFCD FFFF$_H$	—	Access prohibited area
FFCE 0000$_H$ to FFCE 007F$_H$	1	RLN240
FFCE 0080$_H$ to FFCE 00FF$_H$	1	RLN241
FFCE 0100$_H$ to FFCE 017F$_H$	1	RLN242
FFCE 0180$_H$ to FFCE 01FF$_H$	1	RLN243
FFCE 0200$_H$ to FFCE 1FFF$_H$	—	Access prohibited area
FFCE 2000$_H$ to FFCE 203F$_H$	1	RLN30
FFCE 2040$_H$ to FFCE 207F$_H$	1	RLN31
FFCE 2080$_H$ to FFCE 20BF$_H$	1	RLN32
FFCE 20C0$_H$ to FFCE 20FF$_H$	1	RLN33
FFCE 2100$_H$ to FFCE 213F$_H$	1	RLN34
FFCE 2140$_H$ to FFCE 217F$_H$	1	RLN35
FFCE 2180$_H$ to FFCE 21BF$_H$	1	RLN36
FFCE 21C0$_H$ to FFCE 21FF$_H$	1	RLN37
FFCE 2200$_H$ to FFCE FFFF$_H$	—	Access prohibited area
FFCF 0000$_H$ to FFCF 007F$_H$	1	RSENT0
FFCF 0080$_H$ to FFCF 00FF$_H$	—	Access prohibited area
FFCF 0100$_H$ to FFCF 017F$_H$	1	RSENT1
FFCF 0180$_H$ to FFCF FFFF$_H$	—	Access prohibited area
FFD0 0000$_H$ to FFD1 FFFF$_H$	5	RCFDC0
FFD2 0000$_H$ to FFD3 FFFF$_H$	5	RCFDC1
FFD4 0000$_H$ to FFD6 CFFF$_H$	—	Access prohibited area
FFD6 D000$_H$ to FFD6 D7FF$_H$	3	ADCA1
FFD6 D800$_H$ to FFD6 DFFF$_H$	—	Access prohibited area
FFD6 E000$_H$ to FFD6 E7FF$_H$	3	ETNB0
FFD6 E800$_H$ to FFD6 EFFF$_H$	3	ETNB1
FFD6 F000$_H$ to FFD6 FFFF$_H$	—	Access prohibited area
FFD7 0000$_H$ to FFD7 003F$_H$	3	OSTM0
FFD7 0040$_H$ to FFD7 00FF$_H$	—	Access prohibited area
FFD7 0100$_H$ to FFD7 013F$_H$	3	OSTM1
FFD7 0140$_H$ to FFD7 01FF$_H$	—	Access prohibited area
FFD7 0200$_H$ to FFD7 023F$_H$	3	OSTM2
FFD7 0240$_H$ to FFD7 02FF$_H$	—	Access prohibited area

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFD7 0300$_H$ to FFD7 033F$_H$	3	OSTM3
FFD7 0340$_H$ to FFD7 03FF$_H$	—	Access prohibited area
FFD7 0400$_H$ to FFD7 043F$_H$	3	OSTM4
FFD7 0440$_H$ to FFD7 0FFF$_H$	—	Access prohibited area
FFD7 1000$_H$ to FFD7 103F$_H$	3	OSTM5
FFD7 1040$_H$ to FFD7 10FF$_H$	—	Access prohibited area
FFD7 1100$_H$ to FFD7 113F$_H$	3	OSTM6
FFD7 1140$_H$ to FFD7 11FF$_H$	—	Access prohibited area
FFD7 1200$_H$ to FFD7 123F$_H$	3	OSTM7
FFD7 1240$_H$ to FFD7 12FF$_H$	—	Access prohibited area
FFD7 1300$_H$ to FFD7 133F$_H$	3	OSTM8
FFD7 1340$_H$ to FFD7 13FF$_H$	—	Access prohibited area
FFD7 1400$_H$ to FFD7 143F$_H$	3	OSTM9
FFD7 1440$_H$ to FFD7 FFFF$_H$	—	Access prohibited area
FFD8 0000$_H$ to FFD8 001F$_H$	3	CSIH0 (CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU)
FFD8 0020$_H$ to FFD8 0FFF$_H$	—	Access prohibited area
FFD8 1000$_H$ to FFD8 107F$_H$	3	CSIH0 (CSIH0 registers other than above)
FFD8 1080$_H$ to FFD8 1FFF$_H$	—	Access prohibited area
FFD8 2000$_H$ to FFD8 201F$_H$	3	CSIH1 (CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU)
FFD8 2020$_H$ to FFD8 2FFF$_H$	—	Access prohibited area
FFD8 3000$_H$ to FFD8 307F$_H$	3	CSIH1 (CSIH1 registers other than above)
FFD8 3080$_H$ to FFD8 3FFF$_H$	—	Access prohibited area
FFD8 4000$_H$ to FFD8 401F$_H$	3	CSIH2 (CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU)
FFD8 4020$_H$ to FFD8 4FFF$_H$	—	Access prohibited area
FFD8 5000$_H$ to FFD8 507F$_H$	3	CSIH2 (CSIH2 registers other than above)
FFD8 5080$_H$ to FFD8 5FFF$_H$	—	Access prohibited area
FFD8 6000$_H$ to FFD8 601F$_H$	3	CSIH3 (CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU)
FFD8 6020$_H$ to FFD8 6FFF$_H$	—	Access prohibited area
FFD8 7000$_H$ to FFD8 707F$_H$	3	CSIH3 (CSIH3 registers other than above)
FFD8 7080$_H$ to FFD8 7FFF$_H$	—	Access prohibited area
FFD8 8000$_H$ to FFD8 801F$_H$	3	CSIG0 (CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU)
FFD8 8020$_H$ to FFD8 8FFF$_H$	—	Access prohibited area
FFD8 9000$_H$ to FFD8 901F$_H$	3	CSIG0 (CSIG0 registers other than above)
FFD8 9020$_H$ to FFD8 9FFF$_H$	—	Access prohibited area
FFD8 A000$_H$ to FFD8 A01F$_H$	3	CSIG1 (CSIG1CTL0-2, CSIG1STR0, CSIG1STCR0, CSIG1EMU)
FFD8 A020$_H$ to FFD8 AFFF$_H$	—	Access prohibited area
FFD8 B000$_H$ to FFD8 B01F$_H$	3	CSIG1 (CSIG1 registers other than above)
FFD8 B020$_H$ to FFD8 BFFF$_H$	—	Access prohibited area
FFD8 C000$_H$ to FFD8 C01F$_H$	3	CSIG2 (CSIG2CTL0-2, CSIG2STR0, CSIG2STCR0, CSIG2EMU)
FFD8 C020$_H$ to FFD8 CFFF$_H$	—	Access prohibited area
FFD8 D000$_H$ to FFD8 D01F$_H$	3	CSIG2 (CSIG2 registers other than above)
FFD8 D020$_H$ to FFD8 DFFF$_H$	—	Access prohibited area
FFD8 E000$_H$ to FFD8 E01F$_H$	3	CSIG3 (CSIG3CTL0-2, CSIG3STR0, CSIG3STCR0, CSIG3EMU)
FFD8 E020$_H$ to FFD8 EFFF$_H$	—	Access prohibited area
FFD8 F000$_H$ to FFD8 F01F$_H$	3	CSIG3 (CSIG3 registers other than above)

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFD8 F020$_H$ to FFD8 FFFF$_H$	—	Access prohibited area
FFD9 0000$_H$ to FFD9 001F$_H$	3	CSIH4 (CSIH4CTL0-2, CSIH4STR0, CSIH4STCR0, CSIH4EMU)
FFD9 0020$_H$ to FFD9 0FFF$_H$	—	Access prohibited area
FFD9 1000$_H$ to FFD9 107F$_H$	3	CSIH4 (CSIH4 registers other than above)
FFD9 1080$_H$ to FFD9 1FFF$_H$	—	Access prohibited area
FFD9 2000$_H$ to FFD9 201F$_H$	3	CSIG4 (CSIG4CTL0-2, CSIG4STR0, CSIG4STCR0, CSIG4EMU)
FFD9 2020$_H$ to FFD9 2FFF$_H$	—	Access prohibited area
FFD9 3000$_H$ to FFD9 301F$_H$	3	CSIG4 (CSIG4 registers other than above)
FFD9 3020$_H$ to FFDC FFFF$_H$	—	Access prohibited area
FFDD 0000$_H$ to FFDD 00FF$_H$	2	PIC0
FFDD 0100$_H$ to FFDD CFFF$_H$	—	Access prohibited area
FFDD D000$_H$ to FFDD D04F$_H$	2	P-Bus guard (PBG20)
FFDD D050$_H$ to FFDD D0FF$_H$	—	Access prohibited area
FFDD D100$_H$ to FFDD D14F$_H$	2	P-Bus guard (PBG21)
FFDD D150$_H$ to FFE1 FFFF$_H$	—	Access prohibited area
FFE2 0000$_H$ to FFE2 03FF$_H$	2	TAUD0
FFE2 0400$_H$ to FFE2 3FFF$_H$	—	Access prohibited area
FFE2 4000$_H$ to FFE2 4003$_H$	2	SELB_TAUD0 (SL_TAUD0)
FFE2 4004$_H$ to FFE2 FFFF$_H$	—	Access prohibited area
FFE3 0000$_H$ to FFE3 03FF$_H$	2	TAUB0
FFE3 0400$_H$ to FFE3 0FFF$_H$	—	Access prohibited area
FFE3 1000$_H$ to FFE3 13FF$_H$	2	TAUB1
FFE3 1400$_H$ to FFE3 1FFF$_H$	—	Access prohibited area
FFE3 2000$_H$ to FFE3 2003$_H$	2	SELB_TAUB0 (SL_TAUB0)
FFE3 2004$_H$ to FFE3 2FFF$_H$	—	Access prohibited area
FFE3 3000$_H$ to FFE3 3003$_H$	2	SELB_TAUB1 (SL_TAUB1)
FFE3 3004$_H$ to FFE4 FFFF$_H$	—	Access prohibited area
FFE5 0000$_H$ to FFE5 00FF$_H$	2	TAUJ0
FFE5 0100$_H$ to FFE5 01FF$_H$	2	TAUJ2
FFE5 0200$_H$ to FFE5 0FFF$_H$	—	Access prohibited area
FFE5 1000$_H$ to FFE5 10FF$_H$	2	TAUJ1
FFE5 1100$_H$ to FFE5 11FF$_H$	2	TAUJ3
FFE5 1200$_H$ to FFE5 3FFF$_H$	—	Access prohibited area
FFE5 4000$_H$ to FFE5 4003$_H$	2	SELB_TAUJ0 (SL_TAUJ0)
FFE5 4004$_H$ to FFE5 4007$_H$	2	SELB_TAUJ2 (SL_TAUJ2)
FFE5 4008$_H$ to FFE6 FFFF$_H$	—	Access prohibited area
FFE7 0000$_H$ to FFE7 03FF$_H$	2	PWSA0
FFE7 0400$_H$ to FFE7 0FFF$_H$	—	Access prohibited area
FFE7 1000$_H$ to FFE7 27FF$_H$	2	PWGAn
FFE7 2800$_H$ to FFE7 281F$_H$	2	PWBA0
FFE7 2820$_H$ to FFE7 2FFF$_H$	—	Access prohibited area
FFE7 3000$_H$ to FFE7 301F$_H$	2	SLPWG
FFE7 3020$_H$ to FFE7 30FF$_H$	—	Access prohibited area
FFE7 3100$_H$ to FFE7 315F$_H$	2	PWGA_INTF
FFE7 3160$_H$ to FFE7 7FFF$_H$	—	Access prohibited area

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFE7 8000$_H$ to FFE7 807F$_H$	2	RTCA0
FFE7 8080$_H$ to FFE7 FFFF$_H$	—	Access prohibited area
FFE8 0000$_H$ to FFE8 007F$_H$	2	ENCA0
FFE8 0080$_H$ to FFE8 FFFF$_H$	—	Access prohibited area
FFE9 0000$_H$ to FFE9 003F$_H$	2	TAPA0
FFE9 0040$_H$ to FFEC FFFF$_H$	—	Access prohibited area
FFED 0000$_H$ to FFED 000F$_H$	2	WDTA0
FFED 0010$_H$ to FFED 0FFF$_H$	—	Access prohibited area
FFED 1000$_H$ to FFED 100F$_H$	2	WDTA1
FFED 1010$_H$ to FFED 1FFF$_H$	—	Access prohibited area
FFED 2000$_H$ to FFED 200F$_H$	2	WDTA2
FFED 2010$_H$ to FFED 8FFF$_H$	—	Access prohibited area
FFED 9000$_H$ to FFED 907F$_H$	2	MMCA0
FFED 9080$_H$ to FFED 9FFF$_H$	—	Access prohibited area
FFED A000$_H$ to FFED A01F$_H$	2	ECCMMCA0A
FFED A020$_H$ to FFED A0FF$_H$	—	Access prohibited area
FFED A100$_H$ to FFED A11F$_H$	2	ECCMMCA0B
FFED A120$_H$ to FFF1 FFFF$_H$	—	Access prohibited area
FFF2 0000$_H$ to FFF2 07FF$_H$	1	ADCA0
FFF2 0800$_H$ to FFF6 FFFF$_H$	—	Access prohibited area
FFF7 0000$_H$ to FFF7 003F$_H$	1	DCRA0
FFF7 0040$_H$ to FFF7 0FFF$_H$	—	Access prohibited area
FFF7 1000$_H$ to FFF7 103F$_H$	1	DCRA1
FFF7 1040$_H$ to FFF7 1FFF$_H$	—	Access prohibited area
FFF7 2000$_H$ to FFF7 203F$_H$	1	DCRA2
FFF7 2040$_H$ to FFF7 2FFF$_H$	—	Access prohibited area
FFF7 3000$_H$ to FFF7 303F$_H$	1	DCRA3
FFF7 3040$_H$ to FFF7 7FFF$_H$	—	Access prohibited area
FFF7 8000$_H$ to FFF7 8003$_H$	1	KR0
FFF7 8004$_H$ to FFF7 FFFF$_H$	—	Access prohibited area
FFF8 0000$_H$ to FFF8 000F$_H$	1	Write protected register (WPROTR)
FFF8 0010$_H$ to FFF8 00FF$_H$	—	Access prohibited area
FFF8 0100$_H$ to FFF8 011F$_H$	1	STBC0
FFF8 0120$_H$ to FFF8 03FF$_H$	—	Access prohibited area
FFF8 0400$_H$ to FFF8 040F$_H$	1	STBC_WUF0
FFF8 0410$_H$ to FFF8 041F$_H$	1	STBC_WUF1
FFF8 0420$_H$ to FFF8 051F$_H$	—	Access prohibited area
FFF8 0520$_H$ to FFF8 052F$_H$	1	STBC_WUF20
FFF8 0530$_H$ to FFF8 075F$_H$	—	Access prohibited area
FFF8 0760$_H$ to FFF8 0AFF$_H$	1	Reset controller / Supply voltage monitor (LVI,VLVI)
FFF8 0B00$_H$ to FFF8 0FFF$_H$	1	STBC_IOHOLD
FFF8 1000$_H$ to FFF8 2FFF$_H$	1	Clock controller (CLKCTL)
FFF8 3000$_H$ to FFF8 307F$_H$	1	LPS0
FFF8 3080$_H$ to FFF8 30FF$_H$	—	Access prohibited area
FFF8 3100$_H$ to FFF8 3207$_H$	1	CVM (SVM)

Table 4A.4 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFF8 3208$_H$ to FFF8 3603$_H$	—	Access prohibited area
FFF8 3604$_H$ to FFF8 7FFF$_H$	—	Access prohibited area
FFF8 8000$_H$ to FFF8 800F$_H$	1	Write protected register (WPROTR)
FFF8 8010$_H$ to FFF8 810F$_H$	—	Access prohibited area
FFF8 8110$_H$ to FFF8 811F$_H$	1	STBC_WUFISO
FFF8 8120$_H$ to FFF8 BFFF$_H$	1	Clock controller (CLKCTL)
FFF8 C000$_H$ to FFF8 CFFF$_H$	1	CLMA0, CLMA
FFF8 D000$_H$ to FFF8 DFFF$_H$	1	CLMA1
FFF8 E000$_H$ to FFF8 EFFF$_H$	1	CLMA2
FFF8 F000$_H$ to FFF8 FFFF$_H$	1	CLMA3
FFF9 0000$_H$ to FFF9 004F$_H$	1	P-Bus guard (PBG50)
FFF9 0050$_H$ to FFF9 3FFF$_H$	—	Access prohibited area
FFF9 4000$_H$ to FFF9 404F$_H$	3	P-Bus guard (PBG30)
FFF9 4050$_H$ to FFF9 40FF$_H$	—	Access prohibited area
FFF9 4100$_H$ to FFF9 414F$_H$	3	P-Bus guard (PBG31)
FFF9 4150$_H$ to FFF9 41FF$_H$	—	Access prohibited area
FFF9 4200$_H$ to FFF9 424F$_H$	3	P-Bus guard (PBG32)
FFF9 4250$_H$ to FFF9 42FF$_H$	—	Access prohibited area
FFF9 4300$_H$ to FFF9 434F$_H$	3	P-Bus guard (PBG33)
FFF9 4350$_H$ to FFF9 BFFF$_H$	—	Access prohibited area
FFF9 C000$_H$ to FFF9 C04F$_H$	3	H-Bus guard (HBG00)
FFF9 C050$_H$ to FFF9 C0FF$_H$	—	Access prohibited area
FFF9 C100$_H$ to FFF9 C14F$_H$	3	H-Bus guard (HBG01)
FFF9 C150$_H$ to FFF9 C1FF$_H$	—	Access prohibited area
FFF9 C200$_H$ to FFF9 C24F$_H$	3	H-Bus guard (HBG02)
FFF9 C250$_H$ to FFFE DFFF$_H$	—	Access prohibited area
FFFE E000$_H$ to FFFE E03F$_H$	CPU local peripheral	IPG
FFFE E040$_H$ to FFFE E5FF$_H$		Access prohibited area
FFFE E600$_H$ to FFFE E6FF$_H$		PEG
FFFE E700$_H$ to FFFE E97F$_H$		Access prohibited area
FFFE E980$_H$ to FFFE E98F$_H$		SEG
FFFE E990$_H$ to FFFE E9FF$_H$		Access prohibited area
FFFE EA00$_H$ to FFFE EBFF$_H$		INTC1
FFFE EC00$_H$ to FFFE EC7F$_H$		MEV
FFFE EC80$_H$ to FFFE EC8F$_H$		IPIRSS
FFFE EC90$_H$ to FFFF 7FFF$_H$	—	Access prohibited area
FFFF 8000$_H$ to FFFF 8FFF$_H$	2	PDMA0
FFFF 9000$_H$ to FFFF 9FFF$_H$	2	PDMA1
FFFF A000$_H$ to FFFF AFFF$_H$	—	Access prohibited area
FFFF B000$_H$ to FFFF BFFF$_H$	2	INTC2
FFFF C000$_H$ to FFFF FFFF$_H$	—	Access prohibited area

Section 4B Address Space of RH850/F1KM-S4

4B.1 Address Space

Table 4B.1, Address Space (4-MB 176/233/272-Pin Product) to **Table 4B.6, Address Space (3-MB 100-Pin Product)** show the address space of the RH850/F1KM.

CAUTION

Do not access an address with which no register is mapped in the on-chip I/O register space. In addition, do not access any access prohibited area specified in **Table 4B.1, Address Space (4-MB 176/233/272-Pin Product)** to **Table 4B.6, Address Space (3-MB 100-Pin Product)**. If such an address is accessed, operation is not guaranteed.

NOTE

The Local RAM is accessible through the following two address areas in the address space.
CPU1 area: Address area accessible from CPU, DMA, FlexRay and ETNB.
Self area: Mirrored address area, accessible only from CPU to refer the CPU’s self resource.

Table 4B.1 Address Space (4-MB 176/233/272-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 003F FFFF$_H$	Code Flash	4 MB
0040 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area	
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	1 KB
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area	
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area (CS0)	16 MB*4
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area	
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area (CS1)	16 MB*4
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area	
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area (CS2)	16 MB*4
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area	
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area (CS3)	16 MB*4
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FEBB FFFF$_H$	Access prohibited area	
FEBC 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	256 KB*1
FEC0 0000$_H$ to FEDB FFFF$_H$	Access prohibited area	
FEDC 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	256 KB*1
FEE0 0000$_H$ to FEEE 7FFF$_H$	Access prohibited area	
FEEE 8000$_H$ to FEEF FFFF$_H$	Global RAM A	96 KB*2
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFE 7FFF$_H$	Access prohibited area	

Table 4B.1 Address Space (4-MB 176/233/272-Pin Product)

Address	Address Space Type	Size
FEFE 8000_H to FEFF $FFFF_H$	Global RAM B	96 KB*2
FF00 0000_H to FF1F $FFFF_H$	Access prohibited area	
FF20 0000_H to FF21 $FFFF_H$	Data flash	128 KB*3
FF22 0000_H to FF9F $FFFF_H$	Access prohibited area	
FFA0 0000_H to FFFD $FFFF_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000_H to FFFE $DFFF_H$	Access prohibited area	
FFFE $E000_H$ to FFFE $FFFF_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000_H to FFFF $4FFF_H$	Access prohibited area	
FFFF 5000_H to FFFF $FFFF_H$	On-chip peripheral I/O area	44 KB

Note 1. 192 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 64 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Note 4. 8 MB in 176/233-pin products: For details, see **Section 16, External Memory Access Controller (MEMC)**.

Table 4B.2 Address Space (4-MB 144-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 003F FFFF$_H$	Code Flash	4 MB
0040 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FEBB FFFF$_H$	Access prohibited area	
FEBC 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	256 KB*[1]
FEC0 0000$_H$ to FEDB FFFF$_H$	Access prohibited area	
FEDC 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	256 KB*[1]
FEE0 0000$_H$ to FEEE 7FFF$_H$	Access prohibited area	
FEEE 8000$_H$ to FEEF FFFF$_H$	Global RAM A	96 KB*[2]
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFE 7FFF$_H$	Access prohibited area	
FEFE 8000$_H$ to FEFF FFFF$_H$	Global RAM B	96 KB*[2]
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF21 FFFF$_H$	Data flash	128 KB*[3]
FF22 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. 192 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 64 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Table 4B.3 Address Space (4-MB 100-Pin Product)

Address	Address Space Type	Size
0000 0000_H to 003F $FFFF_H$	Code Flash	4 MB
0040 0000_H to 00FF $FFFF_H$	Access prohibited area	
0100 0000_H to 0100 $7FFF_H$	Code Flash (Extended user area)	32 KB
0100 8000_H to 1001 $FFFF_H$	Access prohibited area	
1002 0000_H to 1002 $1FFF_H$	FlexRay Interface (FLXA)	8 KB
1002 2000_H to FEBB $FFFF_H$	Access prohibited area	
FEBC 0000_H to FEBF $FFFF_H$	Local RAM (CPU1 area)	256 KB*[1]
FEC0 0000_H to FEDB $FFFF_H$	Access prohibited area	
FEDC 0000_H to FEDF $FFFF_H$	Local RAM (self area)	256 KB*[1]
FEE0 0000_H to FEEE $7FFF_H$	Access prohibited area	
FEEE 8000_H to FEEF $FFFF_H$	Global RAM A	96 KB*[2]
FEF0 0000_H to FEF0 $FFFF_H$	Retention RAM	64 KB
FEF1 0000_H to FEFE $7FFF_H$	Access prohibited area	
FEFE 8000_H to FEFF $FFFF_H$	Global RAM B	96 KB*[2]
FF00 0000_H to FF1F $FFFF_H$	Access prohibited area	
FF20 0000_H to FF21 $FFFF_H$	Data flash	128 KB*[3]
FF22 0000_H to FF9F $FFFF_H$	Access prohibited area	
FFA0 0000_H to FFFD $FFFF_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000_H to FFFE $DFFF_H$	Access prohibited area	
FFFE $E000_H$ to FFFE $FFFF_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000_H to FFFF $4FFF_H$	Access prohibited area	
FFFF 5000_H to FFFF $FFFF_H$	On-chip peripheral I/O area	44 KB

Note 1. 192 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 64 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Table 4B.4 Address Space (3-MB 176/233/272-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 002F FFFF$_H$	Code Flash	3 MB
0030 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area	
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	1 KB
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area	
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area (CS0)	16 MB[*4]
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area	
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area (CS1)	16 MB[*4]
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area	
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area (CS2)	16 MB[*4]
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area	
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area (CS3)	16 MB[*4]
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FEBC FFFF$_H$	Access prohibited area	
FEBD 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	192 KB[*1]
FEC0 0000$_H$ to FEDC FFFF$_H$	Access prohibited area	
FEDD 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	192 KB[*1]
FEE0 0000$_H$ to FEEE FFFF$_H$	Access prohibited area	
FEEF 0000$_H$ to FEEF FFFF$_H$	Global RAM A	64 KB[*2]
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFE FFFF$_H$	Access prohibited area	
FEFF 0000$_H$ to FEFF FFFF$_H$	Global RAM B	64 KB[*2]
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF21 FFFF$_H$	Data flash	128 KB[*3]
FF22 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. 160 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 48 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Note 4. 8 MB in 176/233-pin products: For details, see **Section 16, External Memory Access Controller (MEMC)**.

Table 4B.5 Address Space (3-MB 144-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 002F FFFF$_H$	Code Flash	3 MB
0030 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to 1003 FFFF$_H$	Access prohibited area	
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	4 KB
1004 1000$_H$ to 2FFF FFFF$_H$	Access prohibited area	
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	64 MB
3400 0000$_H$ to FEBC FFFF$_H$	Access prohibited area	
FEBD 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	192 KB*1
FEC0 0000$_H$ to FEDC FFFF$_H$	Access prohibited area	
FEDD 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	192 KB*1
FEE0 0000$_H$ to FEEE FFFF$_H$	Access prohibited area	
FEEF 0000$_H$ to FEEF FFFF$_H$	Global RAM A	64 KB*2
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFE FFFF$_H$	Access prohibited area	
FEFF 0000$_H$ to FEFF FFFF$_H$	Global RAM B	64 KB*2
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF21 FFFF$_H$	Data flash	128 KB*3
FF22 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. 160 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 48 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

Table 4B.6 Address Space (3-MB 100-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 002F FFFF$_H$	Code Flash	3 MB
0030 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area	
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	8 KB
1002 2000$_H$ to FEBC FFFF$_H$	Access prohibited area	
FEBD 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	192 KB*[1]
FEC0 0000$_H$ to FEDC FFFF$_H$	Access prohibited area	
FEDD 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	192 KB*[1]
FEE0 0000$_H$ to FEEE FFFF$_H$	Access prohibited area	
FEEF 0000$_H$ to FEEF FFFF$_H$	Global RAM A	64 KB*[2]
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	64 KB
FEF1 0000$_H$ to FEFE FFFF$_H$	Access prohibited area	
FEFF 0000$_H$ to FEFF FFFF$_H$	Global RAM B	64 KB*[2]
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF21 FFFF$_H$	Data flash	128 KB*[3]
FF22 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. 160 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 2. 48 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 45, RAM**.

Note 3. 96 KB in products of CPU frequency 160 MHz max.: For detail, see **Section 44, Flash Memory**.

4B.2 Address Space Viewed from Each Bus Master

Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product) shows address spaces viewed from each bus master.

4B.2.1 Space in which Instructions can be Fetched

Instructions of the CPU can be fetched from the Code flash, local RAM, global RAM, and retention RAM.

4B.2.2 Data Space Accessible by CPU

See **Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product)** for the spaces accessible from the CPU.

4B.2.3 Data Space Accessible by DMA

See **Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product)** for the spaces accessible from the DMA.

4B.2.4 Data Space Accessible by Flexray

See **Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product)** for the spaces accessible from the Flexray.

4B.2.5 Data Space Accessible by ETNB

See **Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product)** for the spaces accessible from the ETNB.

4B.2.6 Data Space Accessible by Each Bus Master

See **Table 4B.7, Address Space Viewed from Each Bus Master (4-MB Product)** for the spaces accessible from Each Bus Master.

Table 4B.7 Address Space Viewed from Each Bus Master (4-MB Product)

Address	Resource	From CPU	From DMA	From FlexRay	From ETNB
0000 0000$_H$ to 003F FFFF$_H$	Code Flash	✓	✓		
0040 0000$_H$ to 00FF FFFF$_H$	Access prohibited area				
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	✓	✓		
0100 8000$_H$ to 1001 FFFF$_H$	Access prohibited area				
1002 0000$_H$ to 1002 1FFF$_H$	FlexRay Interface (FLXA)	✓	✓		
1002 2000$_H$ to 1002 FFFF$_H$	Access prohibited area				
1003 0000$_H$ to 1003 03FF$_H$	External Memory Access Controller (MEMC)	✓	✓		
1003 0400$_H$ to 1003 FFFF$_H$	Access prohibited area				
1004 0000$_H$ to 1004 0FFF$_H$	Serial Flash Memory Interface (SFMA)	✓	✓		
1004 1000$_H$ to 1FFF FFFF$_H$	Access prohibited area				
2000 0000$_H$ to 20FF FFFF$_H$	External Memory Area (CS0)	✓	✓		
2100 0000$_H$ to 21FF FFFF$_H$	Access prohibited area				
2200 0000$_H$ to 22FF FFFF$_H$	External Memory Area (CS1)	✓	✓		
2300 0000$_H$ to 23FF FFFF$_H$	Access prohibited area				
2400 0000$_H$ to 24FF FFFF$_H$	External Memory Area (CS2)	✓	✓		
2500 0000$_H$ to 27FF FFFF$_H$	Access prohibited area				
2800 0000$_H$ to 28FF FFFF$_H$	External Memory Area (CS3)	✓	✓		
2900 0000$_H$ to 2FFF FFFF$_H$	Access prohibited area				
3000 0000$_H$ to 33FF FFFF$_H$	External Serial Flash Memory Area	✓	✓		
3400 0000$_H$ to FEBB FFFF$_H$	Access prohibited area				
FEBC 0000$_H$ to FEBF FFFF$_H$	Local RAM (CPU1 area)	✓	✓	✓	✓
FEC0 0000$_H$ to FEDB FFFF$_H$	Access prohibited area				
FEDC 0000$_H$ to FEDF FFFF$_H$	Local RAM (self area)	✓			
FEE0 0000$_H$ to FEEE 7FFF$_H$	Access prohibited area				
FEEE 8000$_H$ to FEEF FFFF$_H$	Global RAM A	✓	✓	✓	✓
FEF0 0000$_H$ to FEF0 FFFF$_H$	Retention RAM	✓	✓	✓	✓
FEF1 0000$_H$ to FEFE 7FFF$_H$	Access prohibited area				
FEFE 8000$_H$ to FEFF FFFF$_H$	Global RAM B	✓	✓	✓	✓
FF00 0000$_H$ to FF1F FFFF$_H$	Access prohibited area				
FF20 0000$_H$ to FF21 FFFF$_H$	Data flash	✓	✓		
FF22 0000$_H$ to FF9F FFFF$_H$	Access prohibited area				
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	✓	✓		
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area				
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	✓			
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area				
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	✓	✓		

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

4B.3 Peripheral I/O Address Map

Table 4B.8, Peripheral I/O Address Map shows peripheral I/O address map.

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FF00 0000$_H$ to FF1F FFFF$_H$	—	Access prohibited area
FF20 0000$_H$ to FF21 FFFF$_H$	4	Data Flash
FF22 0000$_H$ to FF9F FFFF$_H$	—	Access prohibited area
FFA0 0000$_H$ to FFA0 001F$_H$	1	FLMD
FFA0 0020$_H$ to FFA0 0FFF$_H$	—	Access prohibited area
FFA0 1000$_H$ to FFA0 103F$_H$	—	Access prohibited area
FFA0 1040$_H$ to FFA0 1FFF$_H$	—	Access prohibited area
FFA0 2000$_H$ to FFA0 201F$_H$	—	Access prohibited area
FFA0 2020$_H$ to FFA0 FFFF$_H$	—	Access prohibited area
FFA1 0000$_H$ to FFA1 1FFF$_H$	1	Flash controller
FFA1 2000$_H$ to FFA1 FFFF$_H$	—	Access prohibited area
FFA2 0000$_H$ to FFA2 FFFF$_H$	1	FACI command-issuing area
FFA3 0000$_H$ to FFBF FFFF$_H$	—	Access prohibited area
FFC0 0000$_H$ to FFC0 000F$_H$	1	FENMI (ECON_NMI)
FFC0 0010$_H$ to FFC0 00FF$_H$	—	Access prohibited area
FFC0 0100$_H$ to FFC0 010F$_H$	1	FEINT (ECON_FEINT)
FFC0 0110$_H$ to FFC0 0FFF$_H$	—	Access prohibited area
FFC0 1000$_H$ to FFC0 1003$_H$	1	SELB_INTC (SL_INTC)
FFC0 1004$_H$ to FFC0 1FFF$_H$	—	Access prohibited area
FFC0 2000$_H$ to FFC0 200F$_H$	1	SELB_DMAC (SL_DMAC)
FFC0 2010$_H$ to FFC0 2FFF$_H$	—	Access prohibited area
FFC0 3000$_H$ to FFC0 300F$_H$	1	GRZF
FFC0 3010$_H$ to FFC0 FFFF$_H$	—	Access prohibited area
FFC1 0000$_H$ to FFC1 4CCF$_H$	1	PORT
FFC1 4CD0$_H$ to FFC1 FFFF$_H$	—	Access prohibited area
FFC2 0000$_H$ to FFC2 04CF$_H$	1	PORT(JTAG)
FFC2 04D0$_H$ to FFC2 FFFF$_H$	—	Access prohibited area
FFC3 0000$_H$ to FFC3 000F$_H$	1	DNFA_TAUD0 (DNF)
FFC3 0010$_H$ to FFC3 001F$_H$	—	Access prohibited area
FFC3 0020$_H$ to FFC3 002F$_H$	1	DNFA_TAUB0 (DNF)
FFC3 0030$_H$ to FFC3 003F$_H$	—	Access prohibited area
FFC3 0040$_H$ to FFC3 004F$_H$	1	DNFA_TAUB1 (DNF)
FFC3 0050$_H$ to FFC3 005F$_H$	—	Access prohibited area
FFC3 0060$_H$ to FFC3 006F$_H$	1	DNFA_ENCA0 (DNF)
FFC3 0070$_H$ to FFC3 009F$_H$	—	Access prohibited area
FFC3 00A0$_H$ to FFC3 00AF$_H$	1	DNFA_ADCA0 (DNF)
FFC3 00B0$_H$ to FFC3 00BF$_H$	—	Access prohibited area
FFC3 00C0$_H$ to FFC3 00CF$_H$	1	DNFA_ADCA1 (DNF)
FFC3 00D0$_H$ to FFC3 00DF$_H$	—	Access prohibited area

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC3 00E0$_H$ to FFC3 00EF$_H$	1	DNFA_SENT (DNF)
FFC3 00F0$_H$ to FFC3 00FF$_H$	—	Access prohibited area
FFC3 0100$_H$ to FFC3 010F$_H$	—	Access prohibited area
FFC3 0110$_H$ to FFC3 3FFF$_H$	—	Access prohibited area
FFC3 4000$_H$ to FFC3 401F$_H$	1	FCLA_NMI (FCLA0)
FFC3 4020$_H$ to FFC3 403F$_H$	1	FCLA_INTPL (FCLA0)
FFC3 4040$_H$ to FFC3 405F$_H$	1	FCLA_INTPH (FCLA0)
FFC3 4060$_H$ to FFC3 407F$_H$	1	FCLA_ADCA0 (FCLA0)
FFC3 4080$_H$ to FFC3 409F$_H$	1	FCLA_ADCA1 (FCLA0)
FFC3 40A0$_H$ to FFC3 40BF$_H$	1	FCLA_INTPU (FCLA0)
FFC3 40C0$_H$ to FFC3 FFFF$_H$	—	Access prohibited area
FFC4 0000$_H$ to FFC4 004F$_H$	1	P-Bus guard (PBG10)
FFC4 0050$_H$ to FFC4 00FF$_H$	—	Access prohibited area
FFC4 0100$_H$ to FFC4 014F$_H$	1	P-Bus guard (PBG11)
FFC4 0150$_H$ to FFC4 03FF$_H$	—	Access prohibited area
FFC4 0400$_H$ to FFC4 044F$_H$	1	P-Bus guard (PBG12)
FFC4 0450$_H$ to FFC4 04FF$_H$	—	Access prohibited area
FFC4 0500$_H$ to FFC4 054F$_H$	1	P-Bus guard (PBG13)
FFC4 0550$_H$ to FFC4 8FFF$_H$	—	Access prohibited area
FFC4 9000$_H$ to FFC4 907F$_H$	2	Global RAM Guard Bank A (MGDGR)
FFC4 9080$_H$ to FFC4 91FF$_H$	—	Access prohibited area
FFC4 9200$_H$ to FFC4 927F$_H$	2	Global RAM Guard Bank B (MGDGR)
FFC4 9280$_H$ to FFC4 BFFF$_H$	—	Access prohibited area
FFC4 C000$_H$ to FFC4 C00F$_H$	2	P-Bus guard (PBGC0)
FFC4 C010$_H$ to FFC4 C10F$_H$	—	Access prohibited area
FFC4 C110$_H$ to FFC4 C13F$_H$	2	P-Bus guard (PBGC1)
FFC4 C140$_H$ to FFC4 C7FF$_H$	—	Access prohibited area
FFC4 C800$_H$ to FFC4 C80F$_H$	2	ERRSLV (PBGC0)
FFC4 C810$_H$ to FFC4 C8FF$_H$	—	Access prohibited area
FFC4 C900$_H$ to FFC4 C90F$_H$	2	ERRSLV (PBGC1)
FFC4 C910$_H$ to FFC5 97FF$_H$	—	Access prohibited area
FFC5 9800$_H$ to FFC5 981F$_H$	4	EEPRDCYCL (DCIB)
FFC5 9820$_H$ to FFC5 9BFF$_H$	—	Access prohibited area
FFC5 9C00$_H$ to FFC5 9C4F$_H$	4	P-Bus guard (PBG40)
FFC5 9C50$_H$ to FFC5 AFFF$_H$	—	Access prohibited area
FFC5 B000$_H$ to FFC5 B003$_H$	2	FBUFCCTL (FBUF_CTRL)
FFC5 B004$_H$ to FFC6 21FF$_H$	—	Access prohibited area
FFC6 2200$_H$ to FFC6 23FF$_H$	2	Code Flash ECC (CFECC_VCI)
FFC6 2400$_H$ to FFC6 25FF$_H$	2	Code Flash ECC (CFECC_CPU1)
FFC6 2600$_H$ to FFC6 29FF$_H$	—	Access prohibited area
FFC6 2A00$_H$ to FFC6 2A3F$_H$	4	Data Flash ECC (DFECC)

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC6 2A40$_H$ to FFC6 3FFF$_H$	—	Access prohibited area
FFC6 4000$_H$ to FFC6 403F$_H$	2	Global RAM ECC Bank A (GRECC)
FFC6 4040$_H$ to FFC6 41FF$_H$	—	Access prohibited area
FFC6 4200$_H$ to FFC6 423F$_H$	2	Global RAM ECC Bank B (GRECC)
FFC6 4240$_H$ to FFC6 4FFF$_H$	—	Access prohibited area
FFC6 5000$_H$ to FFC6 501F$_H$	2	Local RAM ECC TEST PE1 (LRTST)
FFC6 5020$_H$ to FFC6 53FF$_H$	—	Access prohibited area
FFC6 5400$_H$ to FFC6 547F$_H$	2	Local RAM ECC PE1 (LRECC)
FFC6 5480$_H$ to FFC7 00FF$_H$	—	Access prohibited area
FFC7 0100$_H$ to FFC7 011F$_H$	3	ECCCSIH0
FFC7 0120$_H$ to FFC7 01FF$_H$	—	Access prohibited area
FFC7 0200$_H$ to FFC7 021F$_H$	3	ECCCSIH1
FFC7 0220$_H$ to FFC7 02FF$_H$	—	Access prohibited area
FFC7 0300$_H$ to FFC7 031F$_H$	3	ECCCSIH2
FFC7 0320$_H$ to FFC7 03FF$_H$	—	Access prohibited area
FFC7 0400$_H$ to FFC7 041F$_H$	3	ECCCSIH3
FFC7 0420$_H$ to FFC7 12FF$_H$	—	Access prohibited area
FFC7 1300$_H$ to FFC7 131F$_H$	5	ECCCFD0MB
FFC7 1320$_H$ to FFC7 13FF$_H$	—	Access prohibited area
FFC7 1400$_H$ to FFC7 141F$_H$	5	ECCCFD0AFL0
FFC7 1420$_H$ to FFC7 14FF$_H$	—	Access prohibited area
FFC7 1500$_H$ to FFC7 151F$_H$	5	ECCCFD0AFL1
FFC7 1520$_H$ to FFC7 17FF$_H$	—	Access prohibited area
FFC7 1800$_H$ to FFC7 184F$_H$	5	P-Bus guard (PBG60)
FFC7 1850$_H$ to FFC7 30FF$_H$	—	Access prohibited area
FFC7 3100$_H$ to FFC7 311F$_H$	3	ECCFLXA0
FFC7 3120$_H$ to FFC7 31FF$_H$	—	Access prohibited area
FFC7 3200$_H$ to FFC7 321F$_H$	3	ECCFLXA0T0
FFC7 3220$_H$ to FFC7 32FF$_H$	—	Access prohibited area
FFC7 3300$_H$ to FFC7 331F$_H$	3	ECCFLXA0T1
FFC7 3320$_H$ to FFC7 40FF$_H$	—	Access prohibited area
FFC7 4100$_H$ to FFC7 411F$_H$	3	ECCETNB0TX
FFC7 4120$_H$ to FFC7 41FF$_H$	—	Access prohibited area
FFC7 4200$_H$ to FFC7 421F$_H$	3	ECCETNB0RX
FFC7 4220$_H$ to FFC7 7FFF$_H$	—	Access prohibited area
FFC7 8000$_H$ to FFC7 8003$_H$	1	SELB READ TEST (SL_READTEST)
FFC7 8004$_H$ to FFC9 FFFF$_H$	—	Access prohibited area
FFCA 0000$_H$ to FFCA 007F$_H$	1	RIIC0
FFCA 0080$_H$ to FFCA 00FF$_H$	1	RIIC1
FFCA 0100$_H$ to FFCC FFFF$_H$	—	Access prohibited area
FFCD 0000$_H$ to FFCD 01FF$_H$	1	PRDNAME/CHIPID (SCDS)
FFCD 0200$_H$ to FFCD FFFF$_H$	—	Access prohibited area
FFCE 0000$_H$ to FFCE 007F$_H$	1	RLN240
FFCE 0080$_H$ to FFCE 00FF$_H$	1	RLN241
FFCE 0100$_H$ to FFCE 017F$_H$	1	RLN242

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFCE 0180$_H$ to FFCE 1FFF$_H$	—	Access prohibited area
FFCE 2000$_H$ to FFCE 203F$_H$	1	RLN30
FFCE 2040$_H$ to FFCE 207F$_H$	1	RLN31
FFCE 2080$_H$ to FFCE 20BF$_H$	1	RLN32
FFCE 20C0$_H$ to FFCE 20FF$_H$	1	RLN33
FFCE 2100$_H$ to FFCE 213F$_H$	1	RLN34
FFCE 2140$_H$ to FFCE 217F$_H$	1	RLN35
FFCE 2180$_H$ to FFCE 21BF$_H$	1	RLN36
FFCE 21C0$_H$ to FFCE 21FF$_H$	1	RLN37
FFCE 2200$_H$ to FFCE FFFF$_H$	—	Access prohibited area
FFCF 0000$_H$ to FFCF 007F$_H$	1	RSENT0
FFCF 0080$_H$ to FFCF 00FF$_H$	—	Access prohibited area
FFCF 0100$_H$ to FFCF 017F$_H$	1	RSENT1
FFCF 0180$_H$ to FFCF FFFF$_H$	—	Access prohibited area
FFD0 0000$_H$ to FFD1 FFFF$_H$	5	RCFDC0
FFD2 0000$_H$ to FFD6 CFFF$_H$	—	Access prohibited area
FFD6 D000$_H$ to FFD6 D7FF$_H$	3	ADCA1
FFD6 D800$_H$ to FFD6 DFFF$_H$	—	Access prohibited area
FFD6 E000$_H$ to FFD6 E7FF$_H$	3	ETNB0
FFD6 E800$_H$ to FFD6 FFFF$_H$	—	Access prohibited area
FFD7 0000$_H$ to FFD7 003F$_H$	3	OSTM0
FFD7 0040$_H$ to FFD7 00FF$_H$	—	Access prohibited area
FFD7 0100$_H$ to FFD7 013F$_H$	3	OSTM1
FFD7 0140$_H$ to FFD7 01FF$_H$	—	Access prohibited area
FFD7 0200$_H$ to FFD7 023F$_H$	3	OSTM2
FFD7 0240$_H$ to FFD7 02FF$_H$	—	Access prohibited area
FFD7 0300$_H$ to FFD7 033F$_H$	3	OSTM3
FFD7 0340$_H$ to FFD7 03FF$_H$	—	Access prohibited area
FFD7 0400$_H$ to FFD7 043F$_H$	3	OSTM4
FFD7 0440$_H$ to FFD7 FFFF$_H$	—	Access prohibited area
FFD8 0000$_H$ to FFD8 001F$_H$	3	CSIH0 (CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU)
FFD8 0020$_H$ to FFD8 0FFF$_H$	—	Access prohibited area
FFD8 1000$_H$ to FFD8 107F$_H$	3	CSIH0 (CSIH0 registers other than above)
FFD8 1080$_H$ to FFD8 1FFF$_H$	—	Access prohibited area
FFD8 2000$_H$ to FFD8 201F$_H$	3	CSIH1 (CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU)
FFD8 2020$_H$ to FFD8 2FFF$_H$	—	Access prohibited area
FFD8 3000$_H$ to FFD8 307F$_H$	3	CSIH1 (CSIH1 registers other than above)
FFD8 3080$_H$ to FFD8 3FFF$_H$	—	Access prohibited area
FFD8 4000$_H$ to FFD8 401F$_H$	3	CSIH2 (CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU)
FFD8 4020$_H$ to FFD8 4FFF$_H$	—	Access prohibited area
FFD8 5000$_H$ to FFD8 507F$_H$	3	CSIH2 (CSIH2 registers other than above)
FFD8 5080$_H$ to FFD8 5FFF$_H$	—	Access prohibited area
FFD8 6000$_H$ to FFD8 601F$_H$	3	CSIH3 (CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU)
FFD8 6020$_H$ to FFD8 6FFF$_H$	—	Access prohibited area
FFD8 7000$_H$ to FFD8 707F$_H$	3	CSIH3 (CSIH3 registers other than above)

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFD8 7080$_H$ to FFD8 7FFF$_H$	—	Access prohibited area
FFD8 8000$_H$ to FFD8 801F$_H$	3	CSIG0 (CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU)
FFD8 8020$_H$ to FFD8 8FFF$_H$	—	Access prohibited area
FFD8 9000$_H$ to FFD8 901F$_H$	3	CSIG0 (CSIG0 registers other than above)
FFD8 9020$_H$ to FFD8 9FFF$_H$	—	Access prohibited area
FFD8 A000$_H$ to FFD8 A01F$_H$	3	CSIG1 (CSIG1CTL0-2, CSIG1STR0, CSIG1STCR0, CSIG1EMU)
FFD8 A020$_H$ to FFD8 AFFF$_H$	—	Access prohibited area
FFD8 B000$_H$ to FFD8 B01F$_H$	3	CSIG1 (CSIG1 registers other than above)
FFD8 B020$_H$ to FFD8 BFFF$_H$	—	Access prohibited area
FFD8 C000$_H$ to FFD8 C01F$_H$	3	CSIG2 (CSIG2CTL0-2, CSIG2STR0, CSIG2STCR0, CSIG2EMU)
FFD8 C020$_H$ to FFD8 CFFF$_H$	—	Access prohibited area
FFD8 D000$_H$ to FFD8 D01F$_H$	3	CSIG2 (CSIG2 registers other than above)
FFD8 D020$_H$ to FFD8 DFFF$_H$	—	Access prohibited area
FFD8 E000$_H$ to FFD8 E01F$_H$	3	CSIG3 (CSIG3CTL0-2, CSIG3STR0, CSIG3STCR0, CSIG3EMU)
FFD8 E020$_H$ to FFD8 EFFF$_H$	—	Access prohibited area
FFD8 F000$_H$ to FFD8 F01F$_H$	3	CSIG3 (CSIG3 registers other than above)
FFD8 F020$_H$ to FFDC FFFF$_H$	—	Access prohibited area
FFDD 0000$_H$ to FFDD 00FF$_H$	2	PIC0
FFDD 0100$_H$ to FFDD CFFF$_H$	—	Access prohibited area
FFDD D000$_H$ to FFDD D04F$_H$	2	P-Bus guard (PBG20)
FFDD D050$_H$ to FFDD D0FF$_H$	—	Access prohibited area
FFDD D100$_H$ to FFDD D14F$_H$	2	P-Bus guard (PBG21)
FFDD D150$_H$ to FFE1 FFFF$_H$	—	Access prohibited area
FFE2 0000$_H$ to FFE2 03FF$_H$	2	TAUD0
FFE2 0400$_H$ to FFE2 3FFF$_H$	—	Access prohibited area
FFE2 4000$_H$ to FFE2 4003$_H$	2	SELB_TAUD0 (SL_TAUD0)
FFE2 4004$_H$ to FFE2 FFFF$_H$	—	Access prohibited area
FFE3 0000$_H$ to FFE3 03FF$_H$	2	TAUB0
FFE3 0400$_H$ to FFE3 0FFF$_H$	—	Access prohibited area
FFE3 1000$_H$ to FFE3 13FF$_H$	2	TAUB1
FFE3 1400$_H$ to FFE3 1FFF$_H$	—	Access prohibited area
FFE3 2000$_H$ to FFE3 2003$_H$	2	SELB_TAUB0 (SL_TAUB0)
FFE3 2004$_H$ to FFE3 2FFF$_H$	—	Access prohibited area
FFE3 3000$_H$ to FFE3 3003$_H$	2	SELB_TAUB1 (SL_TAUB1)
FFE3 3004$_H$ to FFE4 FFFF$_H$	—	Access prohibited area
FFE5 0000$_H$ to FFE5 00FF$_H$	2	TAUJ0
FFE5 0100$_H$ to FFE5 01FF$_H$	2	TAUJ2
FFE5 0200$_H$ to FFE5 0FFF$_H$	—	Access prohibited area
FFE5 1000$_H$ to FFE5 10FF$_H$	2	TAUJ1
FFE5 1100$_H$ to FFE5 11FF$_H$	2	TAUJ3
FFE5 1200$_H$ to FFE5 3FFF$_H$	—	Access prohibited area
FFE5 4000$_H$ to FFE5 4003$_H$	2	SELB_TAUJ0 (SL_TAUJ0)
FFE5 4004$_H$ to FFE5 4007$_H$	2	SELB_TAUJ2 (SL_TAUJ2)
FFE5 4008$_H$ to FFE6 FFFF$_H$	—	Access prohibited area
FFE7 0000$_H$ to FFE7 03FF$_H$	2	PWSA0

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFE7 0400$_H$ to FFE7 0FFF$_H$	—	Access prohibited area
FFE7 1000$_H$ to FFE7 27FF$_H$	2	PWGAn
FFE7 2800$_H$ to FFE7 281F$_H$	2	PWBA0
FFE7 2820$_H$ to FFE7 2FFF$_H$	—	Access prohibited area
FFE7 3000$_H$ to FFE7 301F$_H$	2	SLPWG
FFE7 3020$_H$ to FFE7 30FF$_H$	—	Access prohibited area
FFE7 3100$_H$ to FFE7 315F$_H$	2	PWGA_INTF
FFE7 3160$_H$ to FFE7 7FFF$_H$	—	Access prohibited area
FFE7 8000$_H$ to FFE7 807F$_H$	2	RTCA0
FFE7 8080$_H$ to FFE7 FFFF$_H$	—	Access prohibited area
FFE8 0000$_H$ to FFE8 007F$_H$	2	ENCA0
FFE8 0080$_H$ to FFE8 FFFF$_H$	—	Access prohibited area
FFE9 0000$_H$ to FFE9 003F$_H$	2	TAPA0
FFE9 0040$_H$ to FFEC FFFF$_H$	—	Access prohibited area
FFED 0000$_H$ to FFED 000F$_H$	2	WDTA0
FFED 0010$_H$ to FFED 0FFF$_H$	—	Access prohibited area
FFED 1000$_H$ to FFED 100F$_H$	2	WDTA1
FFED 1010$_H$ to FFF1 FFFF$_H$	—	Access prohibited area
FFF2 0000$_H$ to FFF2 07FF$_H$	1	ADCA0
FFF2 0800$_H$ to FFF6 FFFF$_H$	—	Access prohibited area
FFF7 0000$_H$ to FFF7 003F$_H$	1	DCRA0
FFF7 0040$_H$ to FFF7 0FFF$_H$	—	Access prohibited area
FFF7 1000$_H$ to FFF7 103F$_H$	1	DCRA1
FFF7 1040$_H$ to FFF7 1FFF$_H$	—	Access prohibited area
FFF7 2000$_H$ to FFF7 203F$_H$	1	DCRA2
FFF7 2040$_H$ to FFF7 2FFF$_H$	—	Access prohibited area
FFF7 3000$_H$ to FFF7 303F$_H$	1	DCRA3
FFF7 3040$_H$ to FFF7 7FFF$_H$	—	Access prohibited area
FFF7 8000$_H$ to FFF7 8003$_H$	1	KR0
FFF7 8004$_H$ to FFF7 FFFF$_H$	—	Access prohibited area
FFF8 0000$_H$ to FFF8 000F$_H$	1	Write protected register (WPROTR)
FFF8 0010$_H$ to FFF8 00FF$_H$	—	Access prohibited area
FFF8 0100$_H$ to FFF8 011F$_H$	1	STBC0
FFF8 0120$_H$ to FFF8 03FF$_H$	—	Access prohibited area
FFF8 0400$_H$ to FFF8 040F$_H$	1	STBC_WUF0
FFF8 0410$_H$ to FFF8 041F$_H$	1	STBC_WUF1
FFF8 0420$_H$ to FFF8 051F$_H$	—	Access prohibited area
FFF8 0520$_H$ to FFF8 052F$_H$	1	STBC_WUF20
FFF8 0530$_H$ to FFF8 075F$_H$	—	Access prohibited area
FFF8 0760$_H$ to FFF8 0AFF$_H$	1	Reset controller / Supply voltage monitor (LVI,VLVI)
FFF8 0B00$_H$ to FFF8 0FFF$_H$	1	STBC_IOHOLD
FFF8 1000$_H$ to FFF8 2FFF$_H$	1	Clock controller (CLKCTL)
FFF8 3000$_H$ to FFF8 307F$_H$	1	LPS0
FFF8 3080$_H$ to FFF8 30FF$_H$	—	Access prohibited area
FFF8 3100$_H$ to FFF8 3207$_H$	1	CVM (SVM)

Table 4B.8 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFF8 3208$_H$ to FFF8 3603$_H$	—	Access prohibited area
FFF8 3604$_H$ to FFF8 7FFF$_H$	—	Access prohibited area
FFF8 8000$_H$ to FFF8 800F$_H$	1	Write protected register (WPROTR)
FFF8 8010$_H$ to FFF8 810F$_H$	—	Access prohibited area
FFF8 8110$_H$ to FFF8 811F$_H$	1	STBC_WUFISO
FFF8 8120$_H$ to FFF8 BFFF$_H$	1	Clock controller (CLKCTL)
FFF8 C000$_H$ to FFF8 CFFF$_H$	1	CLMA0, CLMA
FFF8 D000$_H$ to FFF8 DFFF$_H$	1	CLMA1
FFF8 E000$_H$ to FFF8 EFFF$_H$	1	CLMA2
FFF8 F000$_H$ to FFF8 FFFF$_H$	1	CLMA3
FFF9 0000$_H$ to FFF9 004F$_H$	1	P-Bus guard (PBG50)
FFF9 0050$_H$ to FFF9 3FFF$_H$	—	Access prohibited area
FFF9 4000$_H$ to FFF9 404F$_H$	3	P-Bus guard (PBG30)
FFF9 4050$_H$ to FFF9 40FF$_H$	—	Access prohibited area
FFF9 4100$_H$ to FFF9 414F$_H$	3	P-Bus guard (PBG31)
FFF9 4150$_H$ to FFF9 41FF$_H$	—	Access prohibited area
FFF9 4200$_H$ to FFF9 424F$_H$	3	P-Bus guard (PBG32)
FFF9 4250$_H$ to FFF9 BFFF$_H$	—	Access prohibited area
FFF9 C000$_H$ to FFF9 C04F$_H$	3	H-Bus guard (HBG00)
FFF9 C050$_H$ to FFF9 C0FF$_H$	—	Access prohibited area
FFF9 C100$_H$ to FFF9 C14F$_H$	3	H-Bus guard (HBG01)
FFF9 C150$_H$ to FFF9 C1FF$_H$	—	Access prohibited area
FFF9 C200$_H$ to FFF9 C24F$_H$	3	H-Bus guard (HBG02)
FFF9 C250$_H$ to FFFE DFFF$_H$	—	Access prohibited area
FFFE E000$_H$ to FFFE E03F$_H$	CPU local peripheral	IPG
FFFE E040$_H$ to FFFE E5FF$_H$		Access prohibited area
FFFE E600$_H$ to FFFE E6FF$_H$		PEG
FFFE E700$_H$ to FFFE E97F$_H$		Access prohibited area
FFFE E980$_H$ to FFFE E98F$_H$		SEG
FFFE E990$_H$ to FFFE E9FF$_H$		Access prohibited area
FFFE EA00$_H$ to FFFE EBFF$_H$		INTC1
FFFE EC00$_H$ to FFFF 4FFF$_H$		Access prohibited area
FFFF 5000$_H$ to FFFF 7FFF$_H$	—	Access prohibited area
FFFF 8000$_H$ to FFFF 8FFF$_H$	2	PDMA0
FFFF 9000$_H$ to FFFF AFFF$_H$	—	Access prohibited area
FFFF B000$_H$ to FFFF BFFF$_H$	2	INTC2
FFFF C000$_H$ to FFFF FFFF$_H$	—	Access prohibited area

Section 4C Address Space of RH850/F1KM-S1

4C.1 Address Space

Table 4C.1, Address Space (1-MB 48/64/80/100-Pin Product), **Table 4C.2, Address Space (768-KB 48/64/80/100-Pin Product)**, and **Table 4C.3, Address Space (512-KB 48/64/80/100-Pin Product)** show the address space of the RH850/F1KM.

CAUTION

Do not access an address with which no register is mapped in the on-chip I/O register space. In addition, do not access any access prohibited area specified in **Table 4C.1, Address Space (1-MB 48/64/80/100-Pin Product)**, **Table 4C.2, Address Space (768-KB 48/64/80/100-Pin Product)**, and **Table 4C.3, Address Space (512-KB 48/64/80/100-Pin Product)**. If such an address is accessed, operation is not guaranteed.

NOTE

The Local RAM is accessible through the following two address areas in the address space.
CPU1 area: Address area accessible from CPU and DMA.
Self area: Mirrored address area, accessible only from CPU to refer the CPU's self resource.

Table 4C.1 Address Space (1-MB 48/64/80/100-Pin Product)

Address	Address Space Type	Size
0000 0000_H to 000F $FFFF_H$	Code Flash	1 MB
0010 0000_H to 00FF $FFFF_H$	Access prohibited area	
0100 0000_H to 0100 $7FFF_H$	Code Flash (Extended user area)	32 KB
0100 8000_H to FEBD $FFFF_H$	Access prohibited area	
FEBE 0000_H to FEBF $7FFF_H$	Local RAM (CPU1 area)	96 KB
FEBF 8000_H to FEBF $FFFF_H$	Retention RAM (CPU1 area)	32 KB
FEC0 0000_H to FEDD $FFFF_H$	Access prohibited area	
FEDE 0000_H to FEDF $7FFF_H$	Local RAM (self area)	96 KB
FEDF 8000_H to FEDF $FFFF_H$	Retention RAM (self area)	32 KB
FEE0 0000_H to FF1F $FFFF_H$	Access prohibited area	
FF20 0000_H to FF20 $FFFF_H$	Data Flash	64 KB
FF21 0000_H to FF9F $FFFF_H$	Access prohibited area	
FFA0 0000_H to FFFD $FFFF_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000_H to FFFE $DFFF_H$	Access prohibited area	
FFFE $E000_H$ to FFFE $FFFF_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000_H to FFFF $4FFF_H$	Access prohibited area	
FFFF 5000_H to FFFF $FFFF_H$	On-chip peripheral I/O area	44 KB

Table 4C.2 Address Space (768-KB 48/64/80/100-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 000B FFFF$_H$	Code Flash	768 KB
000C 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to FEBE 7FFF$_H$	Access prohibited area	
FEBE 8000$_H$ to FEBF 7FFF$_H$	Local RAM (CPU1 area)	64 KB
FEBF 8000$_H$ to FEBF FFFF$_H$	Retention RAM (CPU1 area)	32 KB
FEC0 0000$_H$ to FEDE 7FFF$_H$	Access prohibited area	
FEDE 8000$_H$ to FEDF 7FFF$_H$	Local RAM (self area)	64 KB
FEDF 8000$_H$ to FEDF FFFF$_H$	Retention RAM (self area)	32 KB
FEE0 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF20 FFFF$_H$	Data Flash	64 KB
FF21 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Table 4C.3 Address Space (512-KB 48/64/80/100-Pin Product)

Address	Address Space Type	Size
0000 0000$_H$ to 0007 FFFF$_H$	Code Flash	512 KB
0008 0000$_H$ to 00FF FFFF$_H$	Access prohibited area	
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	32 KB
0100 8000$_H$ to FEBE FFFF$_H$	Access prohibited area	
FEBF 0000$_H$ to FEBF 7FFF$_H$	Local RAM (CPU1 area)	32 KB
FEBF 8000$_H$ to FEBF FFFF$_H$	Retention RAM (CPU1 area)	32 KB
FEC0 0000$_H$ to FEDE FFFF$_H$	Access prohibited area	
FEDF 0000$_H$ to FEDF 7FFF$_H$	Local RAM (self area)	32 KB
FEDF 8000$_H$ to FEDF FFFF$_H$	Retention RAM (self area)	32 KB
FEE0 0000$_H$ to FF1F FFFF$_H$	Access prohibited area	
FF20 0000$_H$ to FF20 FFFF$_H$	Data Flash	64 KB
FF21 0000$_H$ to FF9F FFFF$_H$	Access prohibited area	
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	6 MB - 128 KB
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area	
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	8 KB
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

4C.2 Address Space Viewed from Each Bus Master

Table 4C.4, Address Space Viewed from Each Bus Master (1-MB Product) shows address spaces viewed from each bus master.

4C.2.1 Space in which Instructions can be Fetched

Instructions of the CPU can be fetched from the Code flash, local RAM, and retention RAM.

4C.2.2 Data Space Accessible by CPU

See **Table 4C.4, Address Space Viewed from Each Bus Master (1-MB Product)** for the spaces accessible from the CPU.

4C.2.3 Data Space Accessible by Each Bus Master

See **Table 4C.4, Address Space Viewed from Each Bus Master (1-MB Product)** for the spaces accessible from Each Bus Master.

Table 4C.4 Address Space Viewed from Each Bus Master (1-MB Product)

Address	Resource	From CPU	From DMA
0000 0000$_H$ to 000F FFFF$_H$	Code Flash	✓	✓
0010 0000$_H$ to 00FF FFFF$_H$	Access prohibited area		
0100 0000$_H$ to 0100 7FFF$_H$	Code Flash (Extended user area)	✓	✓
0100 8000$_H$ to FEBD FFFF$_H$	Access prohibited area		
FEBE 0000$_H$ to FEBF 7FFF$_H$	Local RAM (CPU1 area)	✓	✓
FEBF 8000$_H$ to FEBF FFFF$_H$	Retention RAM (CPU1 area)	✓	✓
FEC0 0000$_H$ to FEDD FFFF$_H$	Access prohibited area		
FEDE 0000$_H$ to FEDF 7FFF$_H$	Local RAM (self area)	✓	
FEDF 8000$_H$ to FEDF FFFF$_H$	Retention RAM (self area)	✓	
FEE0 0000$_H$ to FF1F FFFF$_H$	Access prohibited area		
FF20 0000$_H$ to FF20 FFFF$_H$	Data Flash	✓	✓
FF21 0000$_H$ to FF9F FFFF$_H$	Access prohibited area		
FFA0 0000$_H$ to FFFD FFFF$_H$	On-chip peripheral I/O area	✓	✓
FFFE 0000$_H$ to FFFE DFFF$_H$	Access prohibited area		
FFFE E000$_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area (self area)	✓	
FFFF 0000$_H$ to FFFF 4FFF$_H$	Access prohibited area		
FFFF 5000$_H$ to FFFF FFFF$_H$	On-chip peripheral I/O area	✓	✓

Note: The following color coding is used in the map above.

Fetch and data access available

Data access available

Access prohibited

4C.3 Peripheral I/O Address Map

Table 4C.5, Peripheral I/O Address Map shows peripheral I/O address map.

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FF00 0000$_H$ to FF1F FFFF$_H$	—	Access prohibited area
FF20 0000$_H$ to FF20 FFFF$_H$	2	Data Flash
FF21 0000$_H$ to FF9F FFFF$_H$	—	Access prohibited area
FFA0 0000$_H$ to FFA0 001F$_H$	1	FLMD
FFA0 0020$_H$ to FFA0 0FFF$_H$	—	Access prohibited area
FFA0 1000$_H$ to FFA0 103F$_H$	—	Access prohibited area
FFA0 1040$_H$ to FFA0 1FFF$_H$	—	Access prohibited area
FFA0 2000$_H$ to FFA0 201F$_H$	—	Access prohibited area
FFA0 2020$_H$ to FFA0 FFFF$_H$	—	Access prohibited area
FFA1 0000$_H$ to FFA1 1FFF$_H$	1	Flash controller
FFA1 2000$_H$ to FFA1 FFFF$_H$	—	Access prohibited area
FFA2 0000$_H$ to FFA2 FFFF$_H$	1	FACI command-issuing area
FFA3 0000$_H$ to FFBF FFFF$_H$	—	Access prohibited area
FFC0 0000$_H$ to FFC0 000F$_H$	1	FENMI (ECON_NMI)
FFC0 0010$_H$ to FFC0 00FF$_H$	—	Access prohibited area
FFC0 0100$_H$ to FFC0 010F$_H$	1	FEINT (ECON_FEINT)
FFC0 0110$_H$ to FFC0 0FFF$_H$	—	Access prohibited area
FFC0 1000$_H$ to FFC0 1003$_H$	1	SELB_INTC (SL_INTC)
FFC0 1004$_H$ to FFC0 1FFF$_H$	—	Access prohibited area
FFC0 2000$_H$ to FFC0 2007$_H$	1	SELB_DMAC (SL_DMAC)
FFC0 2008$_H$ to FFC0 FFFF$_H$	—	Access prohibited area
FFC1 0000$_H$ to FFC1 4CCF$_H$	1	PORT
FFC1 4CD0$_H$ to FFC1 FFFF$_H$	—	Access prohibited area
FFC2 0000$_H$ to FFC2 04CF$_H$	1	PORT (JTAG)
FFC2 04D0$_H$ to FFC2 FFFF$_H$	—	Access prohibited area
FFC3 0000$_H$ to FFC3 000F$_H$	1	DNFA_TAUD0 (DNF)
FFC3 0010$_H$ to FFC3 001F$_H$	—	Access prohibited area
FFC3 0020$_H$ to FFC3 002F$_H$	1	DNFA_TAUB0 (DNF)
FFC3 0030$_H$ to FFC3 003F$_H$	—	Access prohibited area
FFC3 0040$_H$ to FFC3 004F$_H$	—	Access prohibited area
FFC3 0050$_H$ to FFC3 005F$_H$	—	Access prohibited area
FFC3 0060$_H$ to FFC3 006F$_H$	1	DNFA_ENCA0 (DNF)
FFC3 0070$_H$ to FFC3 009F$_H$	—	Access prohibited area
FFC3 00A0$_H$ to FFC3 00AF$_H$	1	DNFA_ADCA0 (DNF)
FFC3 00B0$_H$ to FFC3 00BF$_H$	—	Access prohibited area
FFC3 00C0$_H$ to FFC3 00CF$_H$	—	Access prohibited area
FFC3 00D0$_H$ to FFC3 00DF$_H$	—	Access prohibited area
FFC3 00E0$_H$ to FFC3 00EF$_H$	1	DNFA_SENT (DNF)
FFC3 00F0$_H$ to FFC3 00FF$_H$	—	Access prohibited area
FFC3 0100$_H$ to FFC3 010F$_H$	—	Access prohibited area
FFC3 0110$_H$ to FFC3 3FFF$_H$	—	Access prohibited area
FFC3 4000$_H$ to FFC3 401F$_H$	1	FCLA_NMI (FCLA0)

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC3 4020$_H$ to FFC3 403F$_H$	1	FCLA_INTPL (FCLA0)
FFC3 4040$_H$ to FFC3 405F$_H$	1	FCLA_INTPH (FCLA0)
FFC3 4060$_H$ to FFC3 407F$_H$	1	FCLA_ADCA0 (FCLA0)
FFC3 4080$_H$ to FFC3 FFFF$_H$	—	Access prohibited area
FFC4 0000$_H$ to FFC4 004F$_H$	1	P-Bus guard (PBG10)
FFC4 0050$_H$ to FFC4 00FF$_H$	—	Access prohibited area
FFC4 0100$_H$ to FFC4 014F$_H$	1	P-Bus guard (PBG11)
FFC4 0150$_H$ to FFC4 03FF$_H$	—	Access prohibited area
FFC4 0400$_H$ to FFC4 044F$_H$	1	P-Bus guard (PBG12)
FFC4 0450$_H$ to FFC4 04FF$_H$	—	Access prohibited area
FFC4 0500$_H$ to FFC4 054F$_H$	1	P-Bus guard (PBG13)
FFC4 0550$_H$ to FFC4 BFFF$_H$	—	Access prohibited area
FFC4 C000$_H$ to FFC4 C00F$_H$	2	P-Bus guard (PBGC0)
FFC4 C010$_H$ to FFC4 C0FF$_H$	—	Access prohibited area
FFC4 C100$_H$ to FFC4 C13F$_H$	2	P-Bus guard (PBGC1)
FFC4 C140$_H$ to FFC4 C7FF$_H$	—	Access prohibited area
FFC4 C800$_H$ to FFC4 C80F$_H$	2	ERRSLV (PBGC0)
FFC4 C810$_H$ to FFC4 C8FF$_H$	—	Access prohibited area
FFC4 C900$_H$ to FFC4 C90F$_H$	2	ERRSLV (PBGC1)
FFC4 C910$_H$ to FFC5 97FF$_H$	—	Access prohibited area
FFC5 9800$_H$ to FFC5 981F$_H$	2	EEPRDCYCL (DCIB)
FFC5 9820$_H$ to FFC5 AFFF$_H$	—	Access prohibited area
FFC5 B000$_H$ to FFC5 B003$_H$	2	FBUFCCTL (FBUF_CTRL)
FFC5 B004$_H$ to FFC6 21FF$_H$	—	Access prohibited area
FFC6 2200$_H$ to FFC6 23FF$_H$	2	Code Flash ECC (CFECC_VCI)
FFC6 2400$_H$ to FFC6 25FF$_H$	2	Code Flash ECC (CFECC_CPU1)
FFC6 2600$_H$ to FFC6 29FF$_H$	—	Access prohibited area
FFC6 2A00$_H$ to FFC6 2A3F$_H$	2	Data Flash ECC (DFECC)
FFC6 2A40$_H$ to FFC6 4FFF$_H$	—	Access prohibited area
FFC6 5000$_H$ to FFC6 501F$_H$	2	Local RAM ECC TEST PE1 (LRTST)
FFC6 5020$_H$ to FFC6 53FF$_H$	—	Access prohibited area
FFC6 5400$_H$ to FFC6 547F$_H$	2	Local RAM ECC PE1 (LRECC)
FFC6 5480$_H$ to FFC7 00FF$_H$	—	Access prohibited area
FFC7 0100$_H$ to FFC7 011F$_H$	3	ECCCSIH0
FFC7 0120$_H$ to FFC7 01FF$_H$	—	Access prohibited area
FFC7 0200$_H$ to FFC7 021F$_H$	3	ECCCSIH1
FFC7 0220$_H$ to FFC7 02FF$_H$	—	Access prohibited area
FFC7 0300$_H$ to FFC7 031F$_H$	3	ECCCSIH2
FFC7 0320$_H$ to FFC7 03FF$_H$	—	Access prohibited area
FFC7 0400$_H$ to FFC7 041F$_H$	3	ECCCSIH3
FFC7 0420$_H$ to FFC7 12FF$_H$	—	Access prohibited area
FFC7 1300$_H$ to FFC7 131F$_H$	5	ECCCFD0MB
FFC7 1320$_H$ to FFC7 13FF$_H$	—	Access prohibited area
FFC7 1400$_H$ to FFC7 141F$_H$	5	ECCCFD0AFL0
FFC7 1420$_H$ to FFC7 14FF$_H$	—	Access prohibited area

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFC7 1500$_H$ to FFC7 151F$_H$	5	ECCCFD0AFL1
FFC7 1520$_H$ to FFC7 7FFF$_H$	—	Access prohibited area
FFC7 8000$_H$ to FFC7 8003$_H$	1	SELB READ TEST (SL_READTEST)
FFC7 8004$_H$ to FFC9 FFFF$_H$	—	Access prohibited area
FFCA 0000$_H$ to FFCA 007F$_H$	1	RIIC0
FFCA 0080$_H$ to FFCA 00FF$_H$	1	RIIC1
FFCA 0100$_H$ to FFCC FFFF$_H$	—	Access prohibited area
FFCD 0000$_H$ to FFCD 01FF$_H$	1	PRDNAME/CHIPID (SCDS)
FFCD 0200$_H$ to FFCD FFFF$_H$	—	Access prohibited area
FFCE 0000$_H$ to FFCE 007F$_H$	1	RLN240
FFCE 0080$_H$ to FFCE 1FFF$_H$	—	Access prohibited area
FFCE 2000$_H$ to FFCE 203F$_H$	1	RLN30
FFCE 2040$_H$ to FFCE 207F$_H$	1	RLN31
FFCE 2080$_H$ to FFCE 20BF$_H$	1	RLN32
FFCE 20C0$_H$ to FFCE 20FF$_H$	1	RLN33
FFCE 2100$_H$ to FFCE FFFF$_H$	—	Access prohibited area
FFCF 0000$_H$ to FFCF 007F$_H$	1	RSENT0
FFCF 0080$_H$ to FFCF 00FF$_H$	—	Access prohibited area
FFCF 0100$_H$ to FFCF 017F$_H$	1	RSENT1
FFCF 0180$_H$ to FFCF FFFF$_H$	—	Access prohibited area
FFD0 0000$_H$ to FFD1 003F$_H$	3	RCFDC0
FFD1 0040$_H$ to FFD6 CFFF$_H$	—	Access prohibited area
FFD6 D000$_H$ to FFD6 D7FF$_H$	—	Access prohibited area
FFD6 D800$_H$ to FFD6 FFFF$_H$	—	Access prohibited area
FFD7 0000$_H$ to FFD7 003F$_H$	3	OSTM0
FFD7 0040$_H$ to FFD7 FFFF$_H$	—	Access prohibited area
FFD8 0000$_H$ to FFD8 001F$_H$	3	CSIH0 (CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU)
FFD8 0020$_H$ to FFD8 0FFF$_H$	—	Access prohibited area
FFD8 1000$_H$ to FFD8 107F$_H$	3	CSIH0 (CSIH0 registers other than above)
FFD8 1080$_H$ to FFD8 1FFF$_H$	—	Access prohibited area
FFD8 2000$_H$ to FFD8 201F$_H$	3	CSIH1 (CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU)
FFD8 2020$_H$ to FFD8 2FFF$_H$	—	Access prohibited area
FFD8 3000$_H$ to FFD8 307F$_H$	3	CSIH1 (CSIH1 registers other than above)
FFD8 3080$_H$ to FFD8 3FFF$_H$	—	Access prohibited area
FFD8 4000$_H$ to FFD8 401F$_H$	3	CSIH2 (CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU)
FFD8 4020$_H$ to FFD8 4FFF$_H$	—	Access prohibited area
FFD8 5000$_H$ to FFD8 507F$_H$	3	CSIH2 (CSIH2 registers other than above)
FFD8 5080$_H$ to FFD8 5FFF$_H$	—	Access prohibited area
FFD8 6000$_H$ to FFD8 601F$_H$	3	CSIH3 (CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU)
FFD8 6020$_H$ to FFD8 6FFF$_H$	—	Access prohibited area

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFD8 7000$_H$ to FFD8 707F$_H$	3	CSIH3 (CSIH3 registers other than above)
FFD8 7080$_H$ to FFD8 7FFF$_H$	—	Access prohibited area
FFD8 8000$_H$ to FFD8 801F$_H$	3	CSIG0 (CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU)
FFD8 8020$_H$ to FFD8 8FFF$_H$	—	Access prohibited area
FFD8 9000$_H$ to FFD8 901F$_H$	3	CSIG0 (CSIG0 registers other than above)
FFD8 9020$_H$ to FFDC FFFF$_H$	—	Access prohibited area
FFDD 0000$_H$ to FFDD 00FF$_H$	2	PIC0
FFDD 0100$_H$ to FFDD CFFF$_H$	—	Access prohibited area
FFDD D000$_H$ to FFDD D04F$_H$	2	P-Bus guard (PBG20)
FFDD D050$_H$ to FFDD D0FF$_H$	—	Access prohibited area
FFDD D100$_H$ to FFDD D14F$_H$	2	P-Bus guard (PBG21)
FFDD D150$_H$ to FFE1 FFFF$_H$	—	Access prohibited area
FFE2 0000$_H$ to FFE2 03FF$_H$	2	TAUD0
FFE2 0400$_H$ to FFE2 3FFF$_H$	—	Access prohibited area
FFE2 4000$_H$ to FFE2 4003$_H$	2	SELB_TAUD0 (SL_TAUD0)
FFE2 4004$_H$ to FFE2 FFFF$_H$	—	Access prohibited area
FFE3 0000$_H$ to FFE3 03FF$_H$	2	TAUB0
FFE3 0400$_H$ to FFE3 1FFF$_H$	—	Access prohibited area
FFE3 2000$_H$ to FFE3 2003$_H$	2	SELB_TAUB0 (SL_TAUB0)
FFE3 2004$_H$ to FFE4 FFFF$_H$	—	Access prohibited area
FFE5 0000$_H$ to FFE5 00FF$_H$	2	TAUJ0
FFE5 0100$_H$ to FFE5 01FF$_H$	2	TAUJ2
FFE5 0200$_H$ to FFE5 0FFF$_H$	—	Access prohibited area
FFE5 1000$_H$ to FFE5 10FF$_H$	2	TAUJ1
FFE5 1100$_H$ to FFE5 11FF$_H$	2	TAUJ3
FFE5 1200$_H$ to FFE5 3FFF$_H$	—	Access prohibited area
FFE5 4000$_H$ to FFE5 4003$_H$	2	SELB_TAUJ0 (SL_TAUJ0)
FFE5 4004$_H$ to FFE5 4007$_H$	2	SELB_TAUJ2 (SL_TAUJ2)
FFE5 4008$_H$ to FFE6 FFFF$_H$	—	Access prohibited area
FFE7 0000$_H$ to FFE7 02FF$_H$	2	PWSA0
FFE7 0300$_H$ to FFE7 0FFF$_H$	—	Access prohibited area
FFE7 1000$_H$ to FFE7 1BFF$_H$	2	PWGAn
FFE7 1C00$_H$ to FFE7 27FF$_H$	—	Access prohibited area
FFE7 2800$_H$ to FFE7 281F$_H$	2	PWBA0
FFE7 2820$_H$ to FFE7 2FFF$_H$	—	Access prohibited area
FFE7 3000$_H$ to FFE7 301F$_H$	2	SLPWG
FFE7 3020$_H$ to FFE7 30FF$_H$	—	Access prohibited area
FFE7 3100$_H$ to FFE7 312F$_H$	2	PWGA_INTF
FFE7 3130$_H$ to FFE7 7FFF$_H$	—	Access prohibited area
FFE7 8000$_H$ to FFE7 807F$_H$	2	RTCA0
FFE7 8080$_H$ to FFE7 FFFF$_H$	—	Access prohibited area
FFE8 0000$_H$ to FFE8 007F$_H$	2	ENCA0
FFE8 0080$_H$ to FFE8 FFFF$_H$	—	Access prohibited area
FFE9 0000$_H$ to FFE9 003F$_H$	2	TAPA0
FFE9 0040$_H$ to FFEC FFFF$_H$	—	Access prohibited area

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFED 0000$_H$ to FFED 000F$_H$	2	WDTA0
FFED 0010$_H$ to FFED 0FFF$_H$	—	Access prohibited area
FFED 1000$_H$ to FFED 100F$_H$	2	WDTA1
FFED 1010$_H$ to FFF1 FFFF$_H$	—	Access prohibited area
FFF2 0000$_H$ to FFF2 07FF$_H$	1	ADCA0
FFF2 0800$_H$ to FFF6 FFFF$_H$	—	Access prohibited area
FFF7 0000$_H$ to FFF7 003F$_H$	1	DCRA0
FFF7 0040$_H$ to FFF7 0FFF$_H$	—	Access prohibited area
FFF7 1000$_H$ to FFF7 103F$_H$	1	DCRA1
FFF7 1040$_H$ to FFF7 1FFF$_H$	—	Access prohibited area
FFF7 2000$_H$ to FFF7 203F$_H$	1	DCRA2
FFF7 2040$_H$ to FFF7 2FFF$_H$	—	Access prohibited area
FFF7 3000$_H$ to FFF7 303F$_H$	1	DCRA3
FFF7 3040$_H$ to FFF7 7FFF$_H$	—	Access prohibited area
FFF7 8000$_H$ to FFF7 8003$_H$	1	KR0
FFF7 8004$_H$ to FFF7 FFFF$_H$	—	Access prohibited area
FFF8 0000$_H$ to FFF8 000F$_H$	1	Write protected register (WPROTR)
FFF8 0010$_H$ to FFF8 00FF$_H$	—	Access prohibited area
FFF8 0100$_H$ to FFF8 011F$_H$	1	STBC0
FFF8 0120$_H$ to FFF8 03FF$_H$	—	Access prohibited area
FFF8 0400$_H$ to FFF8 040F$_H$	1	STBC_WUF0
FFF8 0410$_H$ to FFF8 041F$_H$	1	STBC_WUF1
FFF8 0420$_H$ to FFF8 051F$_H$	—	Access prohibited area
FFF8 0520$_H$ to FFF8 052F$_H$	1	STBC_WUF20
FFF8 0530$_H$ to FFF8 075F$_H$	—	Access prohibited area
FFF8 0760$_H$ to FFF8 0AFF$_H$	1	Reset controller / Supply voltage monitor (LVI,VLVI)
FFF8 0B00$_H$ to FFF8 0FFF$_H$	1	STBC_IOHOLD
FFF8 1000$_H$ to FFF8 2FFF$_H$	1	Clock controller (CLKCTL)
FFF8 3000$_H$ to FFF8 307F$_H$	1	LPS0
FFF8 3080$_H$ to FFF8 30FF$_H$	—	Access prohibited area
FFF8 3100$_H$ to FFF8 3207$_H$	1	CVM (SVM)
FFF8 3208$_H$ to FFF8 35FF$_H$	—	Access prohibited area
FFF8 3600$_H$ to FFF8 3603$_H$	1	Reset controller CYCRBASE (RESCTL)
FFF8 3604$_H$ to FFF8 7FFF$_H$	—	Access prohibited area
FFF8 8000$_H$ to FFF8 800F$_H$	1	Write protected register (WPROTR)
FFF8 8010$_H$ to FFF8 810F$_H$	—	Access prohibited area
FFF8 8110$_H$ to FFF8 811F$_H$	1	STBC_WUFISO
FFF8 8120$_H$ to FFF8 BFFF$_H$	1	Clock controller (CLKCTL)
FFF8 C000$_H$ to FFF8 CFFF$_H$	1	CLMA0, CLMA
FFF8 D000$_H$ to FFF8 DFFF$_H$	1	CLMA1
FFF8 E000$_H$ to FFF8 EFFF$_H$	1	CLMA3
FFF8 F000$_H$ to FFF8 FFFF$_H$	—	Access prohibited area
FFF9 0000$_H$ to FFF9 004F$_H$	1	P-Bus guard (PBG50)
FFF9 0050$_H$ to FFF9 3FFF$_H$	—	Access prohibited area
FFF9 4000$_H$ to FFF9 404F$_H$	3	P-Bus guard (PBG30)

Table 4C.5 Peripheral I/O Address Map

Address	Peripheral Group	Peripheral I/O
FFF9 4050_H to FFF9 $40FF_H$	—	Access prohibited area
FFF9 4100_H to FFF9 $414F_H$	3	P-Bus guard (PBG31)
FFF9 4150_H to FFF9 $41FF_H$	—	Access prohibited area
FFF9 4200_H to FFF9 $424F_H$	3	P-Bus guard (PBG32)
FFF9 4250_H to FFFE $DFFF_H$	—	Access prohibited area
FFFE $E000_H$ to FFFE $E03F_H$	CPU local peripheral	IPG
FFFE $E040_H$ to FFFE $E5FF_H$		Access prohibited area
FFFE $E600_H$ to FFFE $E6FF_H$		PEG
FFFE $E700_H$ to FFFE $E97F_H$		Access prohibited area
FFFE $E980_H$ to FFFE $E98F_H$		SEG
FFFE $E990_H$ to FFFE $E9FF_H$		Access prohibited area
FFFE $EA00_H$ to FFFE $EBFF_H$		INTC1
FFFE $EC00_H$ to FFFF $4FFF_H$		Access prohibited area
FFFF 5000_H to FFFF $7FFF_H$	—	Access prohibited area
FFFF 8000_H to FFFF $8FFF_H$	2	PDMA0
FFFF 9000_H to FFFF $AFFF_H$	—	Access prohibited area
FFFF $B000_H$ to FFFF $BFFF_H$	2	INTC2
FFFF $C000_H$ to FFFF $FFFF_H$	—	Access prohibited area

Section 5 Write-Protected Registers

This section contains a generic description of the write-protected registers.

The first part in this section describes the features specific to the write-protected registers, and the ensuing sections describe the various registers.

5.1 Overview

5.1.1 Functional Overview

The RH850/F1KH, RH850/F1KM products require a special procedure using write-protected registers to set important registers that affect the system, such as clock, reset, and port-related registers. The settings of protected registers are protected against illegal writing by programs by requiring a special procedure. For details about the protected registers, see **Section 5.1.5, Write-Protection Target Registers**. Write-protected registers are managed in units of protected registers called register protection clusters.

5.1.2 Writing Procedure to Write-Protected Registers

Write access to a write-protected register is enabled by using the following protection unlock sequence:

1. Write the fixed value 0000 $00A5_H$ to the protection command register.
2. Write the desired value to the protected register.
3. Write the bit-wise inversion of the desired value to the protected register.
4. Write the desired value to the protected register.
5. Verify that the desired value has been written to the protected register.

Verify successful write of the desired value to the protected register by verifying that the error monitor bit in the protection status register is “0”.

In case the write was not successful, indicated by the error monitor bit set to “1”, the entire sequence has to be restarted at step 1.

If another register (second register) is accessed between step 1 and step 4 of the above sequence for writing to a write-protected register (first register), the protection mechanism operates as follows:

- If the second register belongs to the same cluster, the write to the protected register fails (the error monitor bit is set to 1). The entire sequence has to be restarted at step 1.
- If the second register does not belong to the same cluster, the protection unlock sequence is not disrupted and the write to the first register completes successfully.

5.1.3 Interrupt during Write Protection Unlock

If an interrupt occurs during the protection unlock sequence, the protection mechanism operates as follows:

(1) If an interrupt request is accepted during the protection unlock sequence and write access to a register of the same cluster is performed.

The protection unlock sequence is disrupted, so the write operation to the protected register cannot be completed upon returning from the interrupt service routine. **Figure 5.1, Example of Interruption of Register Protection Unlock Sequence** shows an execution example.

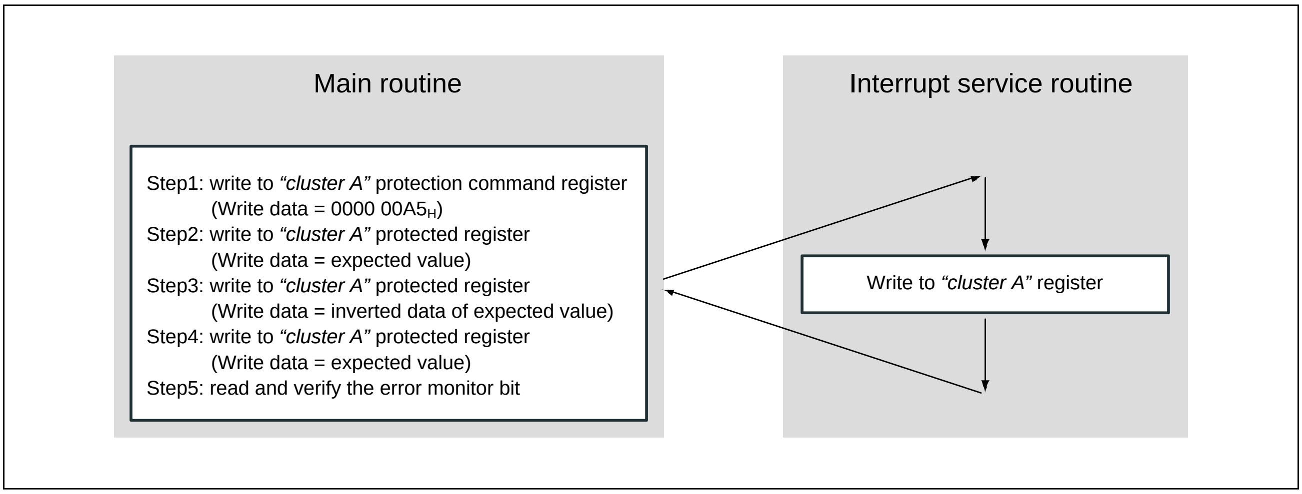


Figure 5.1 Example of Interruption of Register Protection Unlock Sequence

(2) If an interrupt request is accepted during the protection unlock sequence and write access to a register of a different cluster is performed.

The protection unlock sequence is not disrupted, so the write operation to the protected register is completed upon returning from the interrupt service routine. **Figure 5.2, Example of Successful Protection Unlock Sequence** shows an execution example.

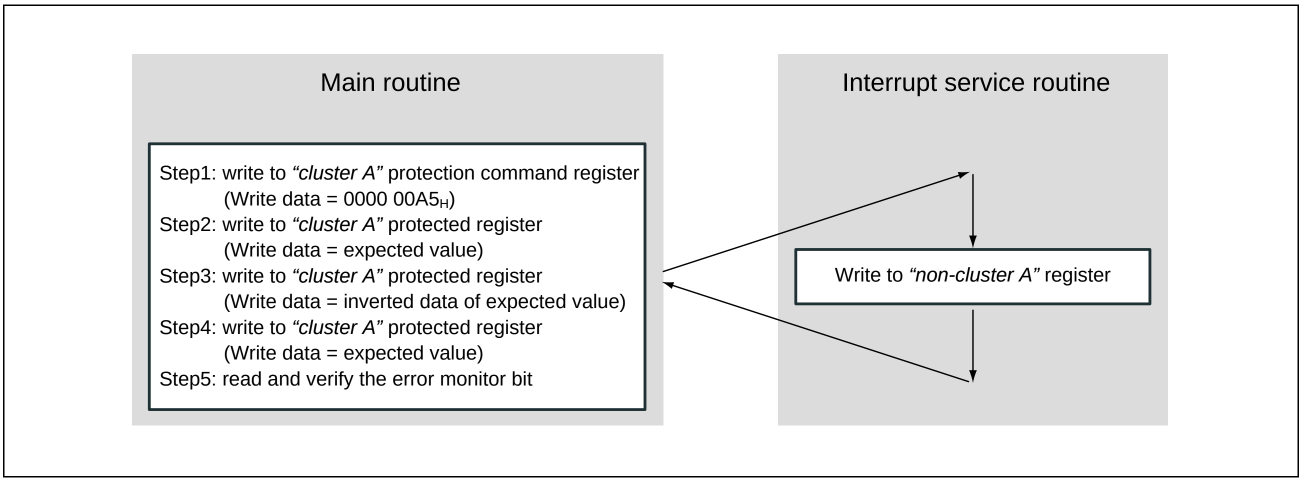


Figure 5.2 Example of Successful Protection Unlock Sequence

For more information on registers of RH850/F1KH, RH850/F1KM register protection clusters, see **Section 5.1.5, Write-Protection Target Registers**.

5.1.4 Emulation Break during Write Protection Unlock Sequence

If an emulation break occurs during the protection unlock sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed.

Even if any register of the same cluster is accessed during the break, the protection unlock sequence is not disrupted and the error monitor bit is not set to 1.

5.1.5 Write-Protection Target Registers

The registers that are protected through the write-protection control registers are listed below.

Table 5.1 Write-Protection Target Registers (RH850/F1KH-D8)

		Protection Control Register		
Protection Target	Protection Target Register	Command Register	Status Register	Protection Cluster
Clock controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	ROSCE			
	SOSCE			
	CKSC_AWDTAD_CTL			
	CKSC_ATAUJS_CTL			
	CKSC_ATAUJD_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AADCAS_CTL			
	CKSC_AADCAD_CTL			
	CKSC_AFOUTS_CTL			
	ROSCUT			
Stand-by function	STBC0PSC			
	STBC0STPT			
	IOHOLD			
Reset function	LVICNT			
	SWRESA			
Clock controller	PLL0E	PROTCMD1	PROTS1	Control protection cluster 1
	PLL1E			
	CKSC_CPUCLKS_CTL			
	CKSC_CPUCLKD_CTL			
	CKSC_IPERI1S_CTL			
	CKSC_IPERI2S_CTL			
	CKSC_ILINS_CTL			
	CKSC_IADCAS_CTL			
	CKSC_IADCAD_CTL			
	CKSC_ILIND_CTL			
	CKSC_ICANS_CTL			
	CKSC_ICANOSCD_CTL			
	CKSC_ICSIS_CTL			
	CKSC_IIICS_CTL			
	CKSC_PPLLCLKS_CTL			
	CKSC_PLL0IS_CTL			
	CKSC_PLL1IS_CTL			

Table 5.1 Write-Protection Target Registers (RH850/F1KH-D8)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock Monitors	CLMA0CTL0	CLMA0PCMD	CLMA0PS	Clock Monitor control protection cluster 0
	CLMA1CTL0	CLMA1PCMD	CLMA1PS	Clock Monitor control protection cluster 1
	CLMA2CTL0	CLMA2PCMD	CLMA2PS	Clock Monitor control protection cluster 2
	CLMA3CTL0	CLMA3PCMD	CLMA3PS	Clock Monitor control protection cluster 3
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster
Port*[1]	JPODC0	JPPCMD0	JPPROTS0	Port protection cluster 0
	PODC0	PPCMD0	PPROTS0	
	PODC1	PPCMD1	PPROTS1	
	PODC2	PPCMD2	PPROTS2	
	PODC3	PPCMD3	PPROTS3	
	PODC8	PPCMD8	PPROTS8	
	JPDSC0	JPPCMD0	JPPROTS0	
	PDSC0	PPCMD0	PPROTS0	
	PDSC1	PPCMD1	PPROTS1	
	PDSC2	PPCMD2	PPROTS2	
	PDSC3	PPCMD3	PPROTS3	
	PODC9	PPCMD9	PPROTS9	Port protection cluster 1
	PODC10	PPCMD10	PPROTS10	
	PODC11	PPCMD11	PPROTS11	
	PODC12	PPCMD12	PPROTS12	
	PODC13	PPCMD13	PPROTS13	
	PODC18	PPCMD18	PPROTS18	
	PODC19	PPCMD19	PPROTS19	
	PODC20	PPCMD20	PPROTS20	
	PODC21	PPCMD21	PPROTS21	
	PODC22	PPCMD22	PPROTS22	
	PODC23	PPCMD23	PPROTS23	
	PODC24	PPCMD24	PPROTS24	
	PDSC10	PPCMD10	PPROTS10	
	PDSC11	PPCMD11	PPROTS11	
	PDSC12	PPCMD12	PPROTS12	
	PDSC13	PPCMD13	PPROTS13	
	PDSC18	PPCMD18	PPROTS18	
	PDSC19	PPCMD19	PPROTS19	
	PDSC20	PPCMD20	PPROTS20	
	PDSC21	PPCMD21	PPROTS21	
	PDSC22	PPCMD22	PPROTS22	
	PDSC23	PPCMD23	PPROTS23	
	PDSC24	PPCMD24	PPROTS24	

Table 5.1 Write-Protection Target Registers (RH850/F1KH-D8)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Core Voltage Monitor	CVMF	PROTCMDCVM	PROTSCVM	Core Voltage Monitor protection cluster
	CVMDIAG			
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection cluster

Note 1. Each port group has its own protection command register and status register. For details, see **Section 5.1.5(1), Port Protection Clusters**.

Table 5.2 Write-Protection Target Registers (RH850/F1KM-S4)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	ROSCE			
	SOSCE			
	CKSC_AWDTAD_CTL			
	CKSC_ATAUJS_CTL			
	CKSC_ATAUJD_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AADCAS_CTL			
	CKSC_AADCAD_CTL			
	CKSC_AFOUTS_CTL			
	ROSCUT			
Stand-by function	STBC0PSC			
	STBC0STPT			
	IOHOLD			
Reset function	LVICNT			
	SWRESA			
Clock controller	PLL0E	PROTCMD1	PROTS1	Control protection cluster 1
	PLL1E			
	CKSC_CPUCLKS_CTL			
	CKSC_CPUCLKD_CTL			
	CKSC_IPERI1S_CTL			
	CKSC_IPERI2S_CTL			
	CKSC_ILINS_CTL			
	CKSC_IADCAS_CTL			
	CKSC_IADCAD_CTL			
	CKSC_ILIND_CTL			
	CKSC_ICANS_CTL			
	CKSC_ICANOSCD_CTL			
	CKSC_ICSIS_CTL			
	CKSC_IIICS_CTL			
	CKSC_PPLLCLKS_CTL			
	CKSC_PLL0IS_CTL			
	CKSC_PLL1IS_CTL			

Table 5.2 Write-Protection Target Registers (RH850/F1KM-S4)

		Protection Control Register		
Protection Target	Protection Target Register	Command Register	Status Register	Protection Cluster
Clock Monitors	CLMA0CTL0	CLMA0PCMD	CLMA0PS	Clock Monitor control protection cluster 0
	CLMA1CTL0	CLMA1PCMD	CLMA1PS	Clock Monitor control protection cluster 1
	CLMA2CTL0	CLMA2PCMD	CLMA2PS	Clock Monitor control protection cluster 2
	CLMA3CTL0	CLMA3PCMD	CLMA3PS	Clock Monitor control protection cluster 3
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster
Port*[1]	JPODC0	JPPCMD0	JPPROTS0	Port protection cluster 0
	PODC0	PPCMD0	PPROTS0	
	PODC1	PPCMD1	PPROTS1	
	PODC2	PPCMD2	PPROTS2	
	PODC3	PPCMD3	PPROTS3	
	PODC8	PPCMD8	PPROTS8	
	JPDSC0	JPPCMD0	JPPROTS0	
	PDSC0	PPCMD0	PPROTS0	
	PDSC1	PPCMD1	PPROTS1	
	PDSC2	PPCMD2	PPROTS2	
	PDSC3	PPCMD3	PPROTS3	
	PODC9	PPCMD9	PPROTS9	Port protection cluster 1
	PODC10	PPCMD10	PPROTS10	
	PODC11	PPCMD11	PPROTS11	
	PODC12	PPCMD12	PPROTS12	
	PODC13	PPCMD13	PPROTS13	
	PODC18	PPCMD18	PPROTS18	
	PODC19	PPCMD19	PPROTS19	
	PODC20	PPCMD20	PPROTS20	
	PODC21	PPCMD21	PPROTS21	
	PODC22	PPCMD22	PPROTS22	
	PDSC10	PPCMD10	PPROTS10	
	PDSC11	PPCMD11	PPROTS11	
	PDSC12	PPCMD12	PPROTS12	
	PDSC13	PPCMD13	PPROTS13	
	PDSC18	PPCMD18	PPROTS18	
	PDSC19	PPCMD19	PPROTS19	
	PDSC20	PPCMD20	PPROTS20	
Core Voltage Monitor	CVMF	PROTCMDCVM	PROTSCVM	Core Voltage Monitor protection cluster
	CVMDIAG			
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection cluster

Note 1. Each port group has its own protection command register and status register. For details, see **Section 5.1.5(1), Port Protection Clusters**.

Table 5.3 Write-Protection Target Registers (RH850/F1KM-S1)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	ROSCE			
	CKSC_AWDTAD_CTL			
	CKSC_ATAUJS_CTL			
	CKSC_ATAUJD_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AADCAS_CTL			
	CKSC_AADCAD_CTL			
	CKSC_AFOUTS_CTL			
	ROSCUT			
Stand-by function	STBC0PSC			
	STBC0STPT			
	IOHOLD			
Reset function	LVICNT			
	SWRESA			
	CYCRBASE			
Clock controller	PLL1E	PROTCMD1	PROTS1	Control protection cluster 1
	CKSC_CPUCLKS_CTL			
	CKSC_CPUCLKD_CTL			
	CKSC_IPERI1S_CTL			
	CKSC_IPERI2S_CTL			
	CKSC_ILINS_CTL			
	CKSC_ILIND_CTL			
	CKSC_ICANS_CTL			
	CKSC_ICANOSCD_CTL			
	CKSC_ICSIS_CTL			
	CKSC_IIICS_CTL			
	CKSC_PPLLCLKS_CTL			
	CKSC_PLL1IS_CTL			
Clock Monitors	CLMA0CTL0	CLMA0PCMD	CLMA0PS	Clock Monitor control protection cluster 0
	CLMA1CTL0	CLMA1PCMD	CLMA1PS	Clock Monitor control protection cluster 1
	CLMA3CTL0	CLMA3PCMD	CLMA3PS	Clock Monitor control protection cluster 3
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster

Table 5.3 Write-Protection Target Registers (RH850/F1KM-S1)

		Protection Control Register		
Protection Target	Protection Target Register	Command Register	Status Register	Protection Cluster
Port*[1]	JPODC0	JPPCMD0	JPPROTS0	Port protection cluster 0
	PODC0	PPCMD0	PPROTS0	
	PODC8	PPCMD8	PPROTS8	
	JPDSC0	JPPCMD0	JPPROTS0	
	PDSC0	PPCMD0	PPROTS0	
	PODC9	PPCMD9	PPROTS9	Port protection cluster 1
	PODC10	PPCMD10	PPROTS10	
	PODC11	PPCMD11	PPROTS11	
	PDSC10	PPCMD10	PPROTS10	
	PDSC11	PPCMD11	PPROTS11	
Core Voltage Monitor	CVMF	PROTCMDCVM	PROTSCVM	Core Voltage Monitor protection cluster
	CVMDIAG			
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection cluster

Note 1. Each port group has its own protection command register and status register. For details, see **Section 5.1.5(1), Port Protection Clusters**.

(1) Port Protection Clusters

The following port control registers have a write protection function:

- Port open drain control registers (PODCn, JPODC0)
- Port drive strength control registers (PDSCn, JPDSC0)

The write protected port registers are divided into two port protection clusters as shown in the following table:

Table 5.4 Port Protection Clusters (RH850/F1KH-D8)

Port Protection Cluster	Port Group
0	JP0, P0, P1, P2, P3, P8
1	P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24

Table 5.5 Port Protection Clusters (RH850/F1KM-S4)

Port Protection Cluster	Port Group
0	JP0, P0, P1, P2, P3, P8
1	P9, P10, P11, P12, P13, P18, P19, P20, P21, P22

Table 5.6 Port Protection Clusters (RH850/F1KM-S1)

Port Protection Cluster	Port Group
0	JP0, P0, P8
1	P9, P10, P11

NOTE

Each port group n has its own port protection command register PPCMDn and port protection status register PPROTSn.

However, any port protection command registers of the same port protection cluster can be used in the protection unlock sequence. For instance, PPCMD0 can be used to unlock PODC8.

5.2 Registers

5.2.1 List of Registers

The following table lists the write-protection control registers.

Table 5.7 List of Registers (RH850/F1KH-D8)

Module Name	Register Name	Symbol	Address
Control protection cluster			
WPROTR	Protection command register 0	PROTCMD0	FFF8 0000_H
	Protection command register 1	PROTCMD1	FFF8 8000_H
	Protection status register 0	PROTS0	FFF8 0004_H
	Protection status register 1	PROTS1	FFF8 8004_H
Clock monitor control and test protection cluster			
CLMA0	CLMA protection command register 0	CLMA0PCMD	FFF8 $C010_H$
CLMA1	CLMA protection command register 1	CLMA1PCMD	FFF8 $D010_H$
CLMA2	CLMA protection command register 2	CLMA2PCMD	FFF8 $E010_H$
CLMA3	CLMA protection command register 3	CLMA3PCMD	FFF8 $F010_H$
CLMA0	CLMA protection status register 0	CLMA0PS	FFF8 $C014_H$
CLMA1	CLMA protection status register 1	CLMA1PS	FFF8 $D014_H$
CLMA2	CLMA protection status register 2	CLMA2PS	FFF8 $E014_H$
CLMA3	CLMA protection status register 3	CLMA3PS	FFF8 $F014_H$
CLMA	Clock monitor test protection command register	PROTCMDCLMA	FFF8 $C200_H$
	Clock monitor test protection status register	PROTSCLMA	FFF8 $C204_H$
Port protection cluster 0			
JTAG	Port protection command registers	JPPCMD0	FFC2 $04C0_H$
PORT		PPCMD0	FFC1 $4C00_H$
		PPCMD1	FFC1 $4C04_H$
		PPCMD2	FFC1 $4C08_H$
		PPCMD3	FFC1 $4C0C_H$
		PPCMD8	FFC1 $4C20_H$
JTAG	Port protection status registers	JPPROTS0	FFC2 $04B0_H$
PORT		PPROTS0	FFC1 $4B00_H$
		PPROTS1	FFC1 $4B04_H$
		PPROTS2	FFC1 $4B08_H$
		PPROTS3	FFC1 $4B0C_H$
		PPROTS8	FFC1 $4B20_H$

Table 5.7 List of Registers (RH850/F1KH-D8)

Module Name	Register Name	Symbol	Address
Port protection cluster 1			
PORT	Port protection command registers	PPCMD9	FFC1 4C24$_H$
		PPCMD10	FFC1 4C28$_H$
		PPCMD11	FFC1 4C2C$_H$
		PPCMD12	FFC1 4C30$_H$
		PPCMD13	FFC1 4C34$_H$
		PPCMD18	FFC1 4C48$_H$
		PPCMD19	FFC1 4C4C$_H$
		PPCMD20	FFC1 4C50$_H$
		PPCMD21	FFC1 4C54$_H$
		PPCMD22	FFC1 4C58$_H$
		PPCMD23	FFC1 4C5C$_H$
		PPCMD24	FFC1 4C60$_H$
	Port protection status registers	PPROTS9	FFC1 4B24$_H$
		PPROTS10	FFC1 4B28$_H$
		PPROTS11	FFC1 4B2C$_H$
		PPROTS12	FFC1 4B30$_H$
		PPROTS13	FFC1 4B34$_H$
		PPROTS18	FFC1 4B48$_H$
		PPROTS19	FFC1 4B4C$_H$
		PPROTS20	FFC1 4B50$_H$
		PPROTS21	FFC1 4B54$_H$
		PPROTS22	FFC1 4B58$_H$
		PPROTS23	FFC1 4B5C$_H$
		PPROTS24	FFC1 4B60$_H$
Core voltage monitor protection cluster			
SVM	Core voltage monitor protection command register	PROTCMDCVM	FFF8 3200$_H$
	Core voltage monitor protection status register	PROTSCVM	FFF8 3204$_H$
Self-programming protection cluster			
FLMD	FLMD protection command register	FLMDPCMD	FFA0 0004$_H$
	FLMD protection error status register	FLMDPS	FFA0 0008$_H$

Table 5.8 List of Registers (RH850/F1KM-S4)

Module Name	Register Name	Symbol	Address
Control protection cluster			
WPROTR	Protection command register 0	PROTCMD0	FFF8 0000$_H$
	Protection command register 1	PROTCMD1	FFF8 8000$_H$
	Protection status register 0	PROTS0	FFF8 0004$_H$
	Protection status register 1	PROTS1	FFF8 8004$_H$
Clock monitor control and test protection cluster			
CLMA0	CLMA protection command register 0	CLMA0PCMD	FFF8 C010$_H$
CLMA1	CLMA protection command register 1	CLMA1PCMD	FFF8 D010$_H$
CLMA2	CLMA protection command register 2	CLMA2PCMD	FFF8 E010$_H$
CLMA3	CLMA protection command register 3	CLMA3PCMD	FFF8 F010$_H$
CLMA0	CLMA protection status register 0	CLMA0PS	FFF8 C014$_H$
CLMA1	CLMA protection status register 1	CLMA1PS	FFF8 D014$_H$
CLMA2	CLMA protection status register 2	CLMA2PS	FFF8 E014$_H$
CLMA3	CLMA protection status register 3	CLMA3PS	FFF8 F014$_H$
CLMA	Clock monitor test protection command register	PROTCMDCLMA	FFF8 C200$_H$
	Clock monitor test protection status register	PROTSCLMA	FFF8 C204$_H$
Port protection cluster 0			
JTAG	Port protection command registers	JPPCMD0	FFC2 04C0$_H$
PORT		PPCMD0	FFC1 4C00$_H$
		PPCMD1	FFC1 4C04$_H$
		PPCMD2	FFC1 4C08$_H$
		PPCMD3	FFC1 4C0C$_H$
		PPCMD8	FFC1 4C20$_H$
JTAG	Port protection status registers	JPPROTS0	FFC2 04B0$_H$
PORT		PPROTS0	FFC1 4B00$_H$
		PPROTS1	FFC1 4B04$_H$
		PPROTS2	FFC1 4B08$_H$
		PPROTS3	FFC1 4B0C$_H$
		PPROTS8	FFC1 4B20$_H$

Table 5.8 List of Registers (RH850/F1KM-S4)

Module Name	Register Name	Symbol	Address
Port protection cluster 1			
PORT	Port protection command registers	PPCMD9	FFC1 4C24$_H$
		PPCMD10	FFC1 4C28$_H$
		PPCMD11	FFC1 4C2C$_H$
		PPCMD12	FFC1 4C30$_H$
		PPCMD13	FFC1 4C34$_H$
		PPCMD18	FFC1 4C48$_H$
		PPCMD19	FFC1 4C4C$_H$
		PPCMD20	FFC1 4C50$_H$
		PPCMD21	FFC1 4C54$_H$
		PPCMD22	FFC1 4C58$_H$
	Port protection status registers	PPROTS9	FFC1 4B24$_H$
		PPROTS10	FFC1 4B28$_H$
		PPROTS11	FFC1 4B2C$_H$
		PPROTS12	FFC1 4B30$_H$
		PPROTS13	FFC1 4B34$_H$
		PPROTS18	FFC1 4B48$_H$
		PPROTS19	FFC1 4B4C$_H$
		PPROTS20	FFC1 4B50$_H$
		PPROTS21	FFC1 4B54$_H$
		PPROTS22	FFC1 4B58$_H$
Core voltage monitor protection cluster			
SVM	Core voltage monitor protection command register	PROTCMDCVM	FFF8 3200$_H$
	Core voltage monitor protection status register	PROTSCVM	FFF8 3204$_H$
Self-programming protection cluster			
FLMD	FLMD protection command register	FLMDPCMD	FFA0 0004$_H$
	FLMD protection error status register	FLMDPS	FFA0 0008$_H$

Table 5.9 List of Registers (RH850/F1KM-S1)

Module Name	Register Name	Symbol	Address
Control protection cluster			
WPROTR	Protection command register 0	PROTCMD0	FFF8 0000_H
	Protection command register 1	PROTCMD1	FFF8 8000_H
	Protection status register 0	PROTS0	FFF8 0004_H
	Protection status register 1	PROTS1	FFF8 8004_H
Clock monitor control and test protection cluster			
CLMA0	CLMA protection command register 0	CLMA0PCMD	FFF8 $C010_H$
CLMA1	CLMA protection command register 1	CLMA1PCMD	FFF8 $D010_H$
CLMA3	CLMA protection command register 3	CLMA3PCMD	FFF8 $E010_H$
CLMA0	CLMA protection status register 0	CLMA0PS	FFF8 $C014_H$
CLMA1	CLMA protection status register 1	CLMA1PS	FFF8 $D014_H$
CLMA3	CLMA protection status register 3	CLMA3PS	FFF8 $E014_H$
CLMA	Clock monitor test protection command register	PROTCMDCLMA	FFF8 $C200_H$
	Clock monitor test protection status register	PROTSCLMA	FFF8 $C204_H$
Port protection cluster 0			
JTAG	Port protection command registers	JPPCMD0	FFC2 $04C0_H$
PORT		PPCMD0	FFC1 $4C00_H$
		PPCMD8	FFC1 $4C20_H$
JTAG	Port protection status registers	JPPROTS0	FFC2 $04B0_H$
PORT		PPROTS0	FFC1 $4B00_H$
		PPROTS8	FFC1 $4B20_H$
Port protection cluster 1			
PORT	Port protection command registers	PPCMD9	FFC1 $4C24_H$
		PPCMD10	FFC1 $4C28_H$
		PPCMD11	FFC1 $4C2C_H$
	Port protection status registers	PPROTS9	FFC1 $4B24_H$
		PPROTS10	FFC1 $4B28_H$
		PPROTS11	FFC1 $4B2C_H$
Core voltage monitor protection cluster			
SVM	Core voltage monitor protection command register	PROTCMDCVM	FFF8 3200_H
	Core voltage monitor protection status register	PROTSCVM	FFF8 3204_H
Self-programming protection cluster			
FLMD	FLMD protection command register	FLMDPCMD	FFA0 0004_H
	FLMD protection error status register	FLMDPS	FFA0 0008_H

5.2.2 Details of Control Protection Cluster Registers

5.2.2.1 PROTCMDn — Protection Command Register

This register is used to initiate the write protection unlock sequence for write-protected registers.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: XXXX XX00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCMDn[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.10 PROTCMDn Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PCMDn[7:0]	Protection command register bits to enable writing to protected registers of control protection cluster

5.2.2.2 PROTSn — Protection Status Register

This register indicates the status of the protection unlock sequence performed by PROTCMDn.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROTSnERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.11 PROTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PROTSnERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.3 Details of Clock Monitor Control and Test Protection Cluster Registers

5.2.3.1 CLMAnPCMD — CLMAn Protection Command Register

This register is a protection command register for the CLMAnCTL0 register.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a write-only register that can be written in 8-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CLMAnREG[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Table 5.12 CLMAnPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMAnREG[7:0]	Protection command register bits to enable writing to the CLMAnCTL0 register

5.2.3.2 CLMAnPS — CLMAn Protection Status Register

This register is used to verify whether the write-protected register (CLMAnCTL0) has been successfully written or not.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnPRERR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 5.13 CLMAnPS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	CLMAnPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.3.3 PROTCMDCLMA — Clock Monitor Test Protection Command Register

This register is a protection command register for the CLMATEST register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: XXXX XX00$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.14 PROTCMDCLMA Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write “0”.
0	CLMATREG[7:0]	Protection command register bits to enable writing to the CLMATEST register

5.2.3.4 PROTSCLMA — Clock Monitor Test Protection Status Register

This register is used to verify whether the write-protected register (CLMATEST) has been successfully written or not.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMAT PRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.15 PROTSCLMA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CLMATPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.4 Details of Core Voltage Monitor Protection Cluster Registers

5.2.4.1 PROTCMDCVM — Core Voltage Monitor Protection Command Register

This register is a protection command register for the CVMF and CVMDIAG registers.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: XXXX XX00$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CVMFREG[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.16 PROTCMDCVM Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	CVMFREG[7:0]	Protection command register bits to enable writing to the CVMF and CVMDIAG registers

5.2.4.2 PROTSCVM — Core Voltage Monitor Protection Status Register

This register is used to verify whether the write-protected register (CVMF, CVMDIAG) has been successfully written or not.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVMFPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.17 PROTSCVM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CVMFPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.5 Details of Port Protection Cluster Registers

5.2.5.1 PPCMDn — Port Protection Command Register

PPCMDn is a protection command register for port group n.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: XXXX XX00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PPCMDn[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.18 PPCMDn Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PPCMDn[7:0]	Protection command register bits that enable writing to port protection cluster registers

NOTE

The protection command register for port group JP0 is JPPCMD0. Its bits are JPPCMD0[7:0].

5.2.5.2 PPROTSn — Port Protection Status Register

PPROTSn is a protection status register for write-protected registers of port group n. It indicates the status of the protection sequence operated by PPCMDn.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers (RH850/F1KH-D8)**, **Table 5.2, Write-Protection Target Registers (RH850/F1KM-S4)** and **Table 5.3, Write-Protection Target Registers (RH850/F1KM-S1)**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPROTSnPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.19 PPROTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PPROTSnPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

NOTE

The protection status register for port group JP0 is JPPROTS0. Its bit is JPPROTS0PRERR.

5.2.6 Details of Self-Programming Protection Cluster Registers

5.2.6.1 FLMDPCMD — FLMD Protection Command Register

FLMDPCMD is a protection command register for the FLMDCNT register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.

Value after reset: XXXX XX00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FLMDPC[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.20 FLMDPCMD Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	FLMDPC[7:0]	Protection command register bits that enable writing to FLMDCNT register

5.2.6.2 FLMDPS — FLMD Protection Error Status Register

This register is used to verify whether the write-protected register (FLMDCNT) has been successfully written or not.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	See **Table 5.7, List of Registers (RH850/F1KH-D8)**, **Table 5.8, List of Registers (RH850/F1KM-S4)** and **Table 5.9, List of Registers (RH850/F1KM-S1)**.
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.21 FLMDPS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	FLMDPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

Section 6 Operating Mode

This section describes the operating mode and mode selection of the RH850/F1KH, RH850/F1KM.

The RH850/F1KH, RH850/F1KM has the operating mode shown below.

- Normal operating mode

  This mode is for execution of the user program. The on-chip debug functions also use this mode. If FLMD0 is pulled up high during operation in this mode, writing to the code flash memory through self-programming is enabled.

  RH850/F1KH-D8:
  CPU2 (PE2) start-up control at reset release depends on the option byte setting. For details of the option byte, see **Section 44.9.2, OPBT0 — Option Byte 0**.

- Serial programming mode

  The dedicated flash memory programmer enables erasing/writing to flash memory.

- Boundary scan mode

  This mode allows boundary scan tests compliant with IEEE Standard 1149.1.

- User boot mode

  This mode is the same as normal operating mode except that the reset vector address is fixed to “0100 0000_H”, and transition to standby modes is not supported.

  RH850/F1KH-D8:
  CPU2 (PE2) is disabled regardless of the option byte setting in this mode.

When the external reset or power-on clear reset is generated, the state of the FLMD0, FLMD1, MODE0, MODE1, and MODE2 pins are used to determine the operating mode after reset is released. The operating mode is fixed by the release of these reset factors. **Table 6.1, Selection of Operating Mode** lists the relationship between the pin settings and the operating mode.

Table 6.1 Selection of Operating Mode

Pins					
FLMD0	FLMD1 (P10_8)	MODE0 (P10_1)	MODE1 (P10_2)	MODE2 (P10_6)	Operating Mode
0	x	x	x	x	Normal operating mode
1	0	x	x	x	Serial programming mode
1	1	0	1	x	Boundary scan mode
1	1	1	1	1	User boot mode
Other than the above					Setting prohibited

CAUTION

To change operating mode, restart from power-on clear reset. (Remove the power supply once and apply it again.) In the case of only by the external reset, some functions are not initialized after the mode transitions. For details of functions not initialized by the external reset, see **Section 9A.1.1, Reset Sources** and **9BC.1.1, Reset Sources**.

Section 7A Exception/Interrupts of RH850/F1KH-D8

7A.1 Features of RH850/F1KH Exception/Interrupts

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions.

The details on exceptions are described in *the RH850G3KH User's Manual: Software*.

Table 7A.1 List of Exception Sources

Name	Symbol	Source	Priority	Saved to
Reset	RESET	Reset input	High	—
FE level non-maskable interrupt*[1]	FENMI	FENMI input		FE
System error exception	SYSERR	SYSERR input		FE
FE level maskable interrupt*[1]	FEINT	FEINT input		FE
Floating-point arithmetic exception (imprecise)	FPI	Execution of FPU instruction		EI
EI level maskable interrupt*[1]	EIINT	Interrupt controller		EI
Memory protection exception (execution right)	MIP	Memory protection violation		FE
System error exception	SYSERR	Error input at instruction fetch		FE
Reserved instruction exception	RIE	Execution of reserved instruction		FE
Coprocessor unusable exception	UCPOP	Execution of coprocessor instruction/access right violation		FE
Privileged instruction exception	PIE	Execution of privileged instruction		FE
Misaligned exception	MAE	Generation of misaligned access		FE
Memory protection exception (access right)	MDP	Memory protection violation		FE
Floating-point arithmetic exception (precise)	FPP	Execution of FPU instruction		EI
System call	SYSCALL	Execution of SYSCALL instruction		EI
FE level trap	FETRAP	Execution of FETRAP instruction		FE
EI level trap 0	TRAP0	Execution of TRAP instruction		EI
EI level trap 1	TRAP1	Execution of TRAP instruction	Low	EI

Note 1. These interrupt exceptions are described in this section.

(1) Interrupts

The following three exceptions in **Table 7A.1, List of Exception Sources** are called interrupts, and are described in this section.

- FE level non-maskable interrupt (FENMI)
  An FENMI interrupt is acknowledged even if another FE level interrupt - FEINT - has been generated.
  - An FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
  - Return from an FENMI interrupt is not possible and neither is recovery.
- FE level maskable interrupt (FEINT)
  - FEINT can be acknowledged if the CPU system register PSW.NP = 0. It is masked if PSW.NP = 1.
  - Return from an FEINT interrupt is possible and so is recovery.
- EI level maskable interrupt (EIINT)
  An EIINT interrupt can be acknowledged if an FE level interrupt - FENMI or FEINT - has not been generated.
  - EIINT can be acknowledged if the CPU system register PSW.NP = 0.
    It is masked if PSW.NP = 1, EIINT with a higher priority is being processed, or PSW.ID = 1.
  - Return from an EIINT interrupt is possible and so is recovery.
  - Interrupt masking can be specified for each interrupt channel.
  - 16 interrupt priority levels can be specified for each interrupt channel
  - In this section, the EIINT that corresponds to interrupt channel n is indicated by “INTn”, whereas the EIINT that corresponds to interrupt source xxx is indicated by “INTxxx”.

For the PSW register, see **Table 3A.10, PSW Register Contents** and *the RH850G3KH User’s Manual: Software*.

NOTE

Return:	Indicates whether or not the program can resume from where it was interrupted.
Recovery:	Indicates whether or not the processor status (status of processor resources including general-purpose registers and system registers) can be restored to the status they were in when the program was interrupted.

These interrupt sources are described in **Section 7A.2, Interrupt Sources**.

(2) Overview of interrupts

- Simultaneous distribution of interrupt sources to multiple cores
  - A single interrupt source can be distributed to multiple CPU cores.
    (Distribution target cores: CPU1/CPU2)
  - Target interrupt sources: Non-maskable interrupt 1 source,
    FE level interrupt 1 source,
    EI level interrupts 29 sources
- Priority levels for interrupt
  16 priority levels of maskable interrupts by request can be set by interrupt control register.
- Detecting methods of external interrupts (TNMI/INTPm)
  A method of detecting external interrupts (TNMI and INTPm) can be selected from five types:
  rising edge, falling edge, both edges, low level, and high level.
- 2 types of interrupt handler address setting Direct branching method or table referencing method is selectable by register setting.
- Inter-processor interrupts
  High-speed inter-processor interrupts are enabled.

7A.2 Interrupt Sources

7A.2.1 Interrupt Sources

7A.2.1.1 FE Level Non-Maskable Interrupts

(1) Priority

See **Table 7A.1, List of Exception Sources**.

(2) Return PC

Return or recovery from an FE non-maskable interrupt is not possible.

(3) Status Register

See **Section 7A.4.5, FNC — FE Level NMI Status Register**.

(4) Return Instruction

None

Table 7A.2 FE Level Non-Maskable Interrupt Requests

Interrupt			Interrupt Request					
	Control Register							
Symbol	Name	Address	Name	Source	Unit	Priority	Exception Source Code	Handler Address 00000...
FENMI	FNC	FFFE EA78_H	TNMI	NMI pin	Port	*1	$0E0_H$	$0E0_H$
			WDTA0NMI	WDTA0 FENMI interrupt	WDTA0			
			WDTA1NMI	WDTA1 FENMI interrupt	WDTA1			
			WDTA2NMI	WDTA2 FENMI interrupt	WDTA2			

Note 1. See **Table 7A.1, List of Exception Sources**.

The source of the FENMI interrupt can be evaluated by a dedicated flag register. See **Section 7A.2.2, FE Level Non-Maskable Interrupt Sources** for details.

7A.2.1.2 FE Level Maskable Interrupts

(1) Priority

See **Table 7A.1, List of Exception Sources**.

(2) Return PC

The return PC returned from an interrupt handling routine by the FERET instruction is the PC from when the program was suspended (current PC).

(3) Status Register

See **Section 7A.4.6, FIC — FE Level Maskable Interrupt Status Register**.

(4) Return Instruction

FERET

Table 7A.3 FE Level Maskable Interrupt Requests (RH850/F1KH-D8)

Interrupt			Interrupt Request		Unit	176 Pins	233 Pins	272 Pins	324 Pins	Priority	Exception Source Code	Handler Address 00000...
	Control Register											
Symbol	Name	Address	Name	Source								
FEINT	FIC	FFFE EA7A$_H$	INTLVIL	LVI voltage detection (falling)	LVI	✓	✓	—	✓	*[1]	0F0$_H$	0F0$_H$
			INTECCEEP0	Data flash ECC 1-bit error or 2-bit error interrupt	Data flash	✓	✓	—	✓			
			INTECCFLRAM	FLXA0 ECC 1-bit error or 2-bit error interrupt	FLXA0	✓	✓	—	✓			
			INTECCCNFDRAM0	RCFDC0 ECC 1-bit error or 2-bit error interrupt	RCFDC0	✓	✓	—	✓			
			INTECCCSIH0	CSIH0 ECC 1-bit error or 2-bit error interrupt	CSIH0	✓	✓	—	✓			
			INTECCCSIH1	CSIH1 ECC 1-bit error or 2-bit error interrupt	CSIH1	✓	✓	—	✓			
			INTECCCSIH2	CSIH2 ECC 1-bit error or 2-bit error interrupt	CSIH2	✓	✓	—	✓			
			INTECCCSIH3	CSIH3 ECC 1-bit error or 2-bit error interrupt	CSIH3	✓	✓	—	✓			
			INTECCCSIH4	CSIH4 ECC 1-bit error or 2-bit error interrupt	CSIH4	✓	✓	—	✓			
			INTECCFLI0	Code Flash ECC 1-bit error or 2-bit error interrupt	Code flash	✓	✓	—	✓			
			INTECCRAM	RAM ECC 1-bit error or 2-bit error interrupt	RAM	✓	✓	—	✓			
			INTOSTM0_FE*[2]	OSTM0 interrupt	OSTM0	✓	✓	—	✓			
			INTLVIH	LVI voltage detection (rising)	LVI	✓	✓	—	✓			
			INTGUARD	PE Guard (PEG) error or Global RAM Guard (GRG) error or Peripheral Guard (PBG/HBG/PBGC)	PEG, GRG, PBG/HBG/PBGC	✓	✓	—	✓			
			INTOSTM1_FE	OSTM1 interrupt	OSTM1	✓	✓	—	✓			
			INTOSTM2_FE	OSTM2 interrupt	OSTM2	✓	✓	—	✓			
			INTOSTM3_FE	OSTM3 interrupt	OSTM3	✓	✓	—	✓			
			INTOSTM4_FE	OSTM4 interrupt	OSTM4	✓	✓	—	✓			
			INTECCCNFDRAM1	RCFDC1 ECC 1-bit error or 2-bit error interrupt	RCFDC1	—	—	—	✓			
			INTECCETH0	ETNB0 ECC 1-bit error or 2-bit error interrupt	ETNB0	✓	✓	—	✓			
			INTOSTM5_FE*[3]	OSTM5 interrupt	OSTM5	✓	✓	—	✓			
			INTOSTM6_FE	OSTM6 interrupt	OSTM6	✓	✓	—	✓			
			INTOSTM7_FE	OSTM7 interrupt	OSTM7	✓	✓	—	✓			
			INTOSTM8_FE	OSTM8 interrupt	OSTM8	✓	✓	—	✓			
			INTOSTM9_FE	OSTM9 interrupt	OSTM9	✓	✓	—	✓			
			INTECCETH1	ETNB1 ECC 1-bit error or 2-bit error interrupt	ETNB1	—	—	—	✓			
			INTECCMMCA0RAM	MMCA0 ECC 1-bit error or 2-bit error interrupt	MMCA0	—	—	—	✓			
			INTDMAERR	DMA transfer error interrupt	PDMAn	✓	✓	—	✓			

Note 1. See **Table 7A.1, List of Exception Sources**.

Note 2. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is prohibited. When INTOSTM0 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM0.

Note 3. INTOSTM5 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is prohibited. When INTOSTM5 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM5

* Timing monitor (TSU)
This prevents the illicit use of CPU time by non-trusted programs, manages properties, and controls the intervals over which interrupts are disabled.

7A.2.1.3 EI Level Maskable Interrupts

(1) Interrupt Naming Rules

The composition of the interrupt request signal names, their assigned interrupt control registers and the bits in these registers follow special rules.

In the following the name of the specific interrupt request is represented by <*name*>.

For details of the names used in **IC**<*name*>, see **Table 7A.4, EI Level Maskable Interrupt Sources**.

- Interrupt request name: **INT**<*name*>
  The prefix "**INT**" is appended to <*name*>.
- Interrupt request control register: **IC**<*name*>
  The prefix "**IC**" is appended to <*name*>.
  The 16-bit register **IC**<*name*> can also be accessed in byte units:
  - Low byte (bits [7:0]) of the **IC** <*name*> register: **IC** <*name*>L
    The suffix "L" is appended to the register name **IC**<*name*>.
  - High byte (bits [15:8]) of the **IC** <*name*> register: **IC** <*name*>H
    The suffix "H" is appended to the register name **IC**<*name*>.
- Interrupt control register bit names: **CT**<*name*>, **RF**<*name*>, **MK**<*name*>, **TB**<*name*>, **P3**<*name*>, **P2**<*name*>, **P1**<*name*>, **P0**<*name*>
  The bit prefix "**CT**", "**RF**", "**MK**", "**TB**", "**P3**", "**P2**", "**P1**", or "**P0**" is appended to the interrupt <*name*>.

Example

The interrupt request from channel 2 of TAUD0 channel (<*name*> = *TAUD0I2*) is named

INTTAUD0I2

The related interrupt control registers are

ICTAUD0I2, **IC**TAUD0I2L, **IC**TAUD0I2H

The bits in this register are

CTTAUD0I2,RFTAUD0I2, MKTAUD0I2, TBTAUD0I2, P3TAUD0I2, P2TAUD0I2,
P1TAUD0I2, P0TAUD0I2

(2) Priority

See **Table 7A.1, List of Exception Sources**.

(3) Return PC

The return PC returned from an interrupt handling routine by the EIRET instruction is the PC from when the program was suspended (current PC).

(4) Control Register

EI level maskable interrupt control register

See **Section 7A.4.2, ICxxx — EI Level Interrupt Control Registers**.

(5) Return Instruction

EIRET instruction

(6) Configuration

EI-level maskable interrupts are controlled by the two controllers, INTC1 and INTC2. The interrupts are supported on a total of 382 channels with a cascade connection of INTC1 and INTC2.

- INTC1
  CPU1 and CPU2 have their own interrupt controllers.
  Each CPU accesses the INTC1 register that corresponds to respective CPUs.
  INTC1 controls high-speed interrupt and has the following functions:
  - Priority setting
  - Interrupt mask setting
- INTC2
  INTC2 is a common interrupt controller that CPU1 and CPU2 share.
  INTC2 controls low-speed interrupts and has the following functions.
  - Priority setting
  - Interrupt mask setting
  - Binding setting

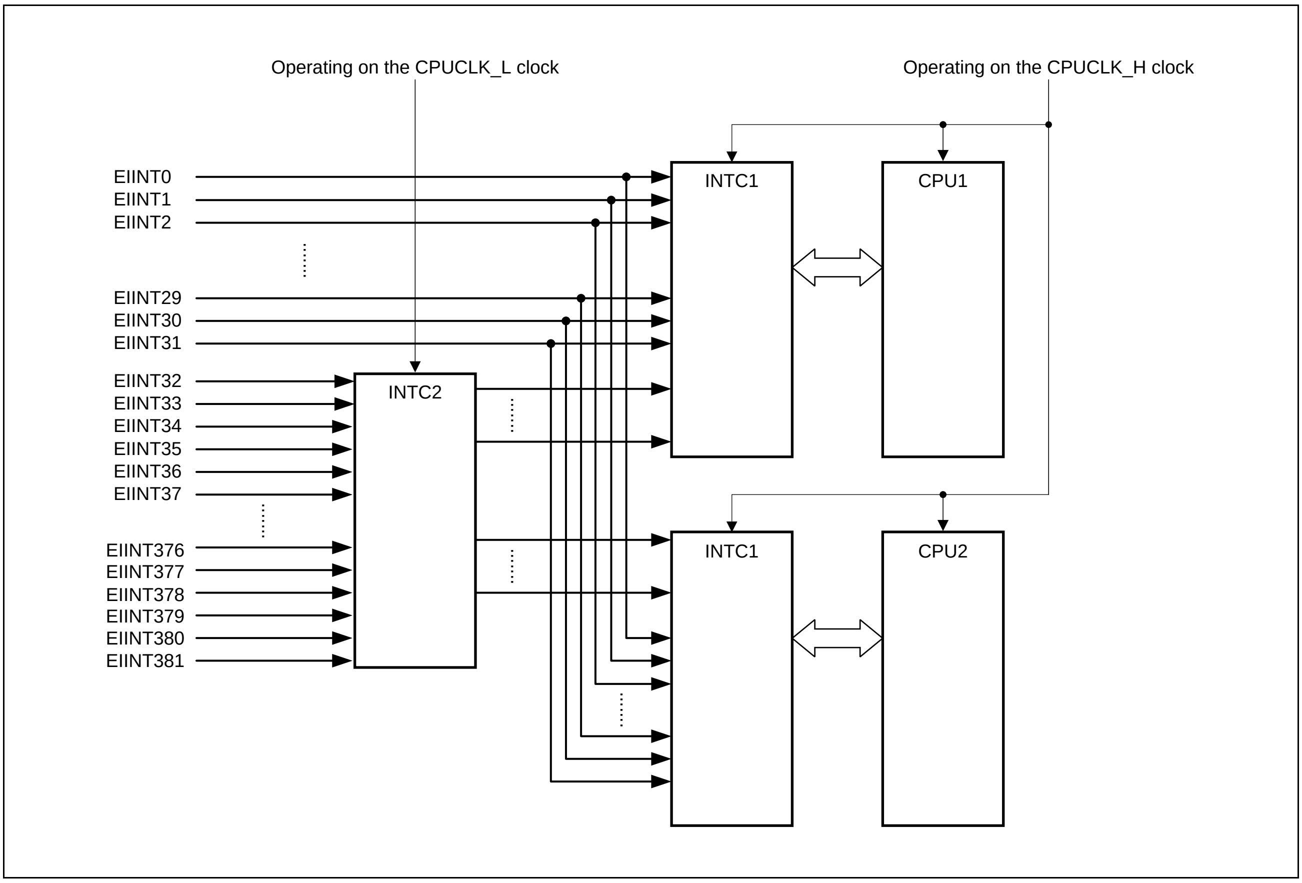


Figure 7A.1 Configuration Diagram of EI Level Maskable Interrupt

CAUTION

As CPUCLK_L is the operating clock for INTC2, the EIINT32 to EIINT381 interrupts, which are connected to INTC2, are delayed more than the interrupts directly connected to INTC1.

See **Table 7A.4, EI Level Maskable Interrupt Sources** lists EI level maskable interrupts.

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register					Detection Type[*2]						Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
0	ICIPIR0	FFFE EA00$_H$	INTIPIR0	Internal processor interrupt 0	IPIRSS	Edge	1000$_H$	✓	✓	—	✓	[*3]	[*4]	+000$_H$
1	ICIPIR1	FFFE EA02$_H$	INTIPIR1	Internal processor interrupt 1	IPIRSS	Edge	1001$_H$	✓	✓	—	✓			+004$_H$
2	ICIPIR2	FFFE EA04$_H$	INTIPIR2	Internal processor interrupt 2	IPIRSS	Edge	1002$_H$	✓	✓	—	✓			+008$_H$
3	ICIPIR3	FFFE EA06$_H$	INTIPIR3	Internal processor interrupt 3	IPIRSS	Edge	1003$_H$	✓	✓	—	✓			+00C$_H$
4	ICBN0	FFFE EA08$_H$	INTBN0	Broadcasting detection interrupt 0	INTC2	Edge	1004$_H$	✓	✓	—	✓			+010$_H$
5	ICBN1	FFFE EA0A$_H$	INTBN1	Broadcasting detection interrupt 1	INTC2	Edge	1005$_H$	✓	✓	—	✓			+014$_H$
6	Reserved	FFFE EA0C$_H$					1006$_H$	—	—	—	—			+018$_H$
7	Reserved	FFFE EA0E$_H$					1007$_H$	—	—	—	—			+01C$_H$
8	ICTAUD0I0	FFFE EA10$_H$	INTTAUD0I0[*6]	Interrupt for CH0 of TAUD0	TAUD0	Edge	1008$_H$	✓	✓	—	✓			+020$_H$
	ICCSIH2IC_1		INTCSIH2IC_1[*6]	CSIH2 communication status interrupt	CSIH2	Edge		✓	✓	—	✓			
9	ICTAUD0I2	FFFE EA12$_H$	INTTAUD0I2[*6]	Interrupt for CH2 of TAUD0	TAUD0	Edge	1009$_H$	✓	✓	—	✓			+024$_H$
	ICCSIH3IC_1		INTCSIH3IC_1[*6]	CSIH3 communication status interrupt	CSIH3	Edge		✓	✓	—	✓			
10	ICTAUD0I4	FFFE EA14$_H$	INTTAUD0I4[*6]	Interrupt for CH4 of TAUD0	TAUD0	Edge	100A$_H$	✓	✓	—	✓			+028$_H$
	ICCSIH2IR_1		INTCSIH2IR_1[*6]	CSIH2 receive status interrupt	CSIH2	Edge		✓	✓	—	✓			
11	ICTAUD0I6	FFFE EA16$_H$	INTTAUD0I6[*6]	Interrupt for CH6 of TAUD0	TAUD0	Edge	100B$_H$	✓	✓	—	✓			+02C$_H$
	ICCSIH2IRE_1		INTCSIH2IRE_1[*6]	CSIH2 communication error interrupt	CSIH2	Edge		✓	✓	—	✓			
12	ICTAUD0I8	FFFE EA18$_H$	INTTAUD0I8[*6]	Interrupt for CH8 of TAUD0	TAUD0	Edge	100C$_H$	✓	✓	—	✓			+030$_H$
	ICCSIH2IJC_1		INTCSIH2IJC_1[*6]	CSIH2 job completion interrupt	CSIH2	Edge		✓	✓	—	✓			
13	ICTAUD0I10	FFFE EA1A$_H$	INTTAUD0I10[*6]	Interrupt for CH10 of TAUD0	TAUD0	Edge	100D$_H$	✓	✓	—	✓			+034$_H$
	ICCSIH3IR_1		INTCSIH3IR_1[*6]	CSIH3 receive status interrupt	CSIH3	Edge		✓	✓	—	✓			
14	ICTAUD0I12	FFFE EA1C$_H$	INTTAUD0I12[*6]	Interrupt for CH12 of TAUD0	TAUD0	Edge	100E$_H$	✓	✓	—	✓			+038$_H$
	ICCSIH3IRE_1		INTCSIH3IRE_1[*6]	CSIH3 communication error interrupt	CSIH3	Edge		✓	✓	—	✓			
15	ICTAUD0I14	FFFE EA1E$_H$	INTTAUD0I14[*6]	Interrupt for CH14 of TAUD0	TAUD0	Edge	100F$_H$	✓	✓	—	✓			+03C$_H$
	ICCSIH3IJC_1		INTCSIH3IJC_1[*6]	CSIH3 job completion interrupt	CSIH3	Edge		✓	✓	—	✓			
16	ICTAPA0IPEK0	FFFE EA20$_H$	INTTAPA0IPEK0[*6]	TAPA0 peak interrupt 0	TAPA0	Edge	1010$_H$	✓	✓	—	✓			+040$_H$
	ICCSIH1IC_1		INTCSIH1IC_1[*6]	CSIH1 communication status interrupt	CSIH1	Edge		✓	✓	—	✓			
17	ICTAPA0IVLY0	FFFE EA22$_H$	INTTAPA0IVLY0[*6]	TAPA0 valley interrupt 0	TAPA0	Edge	1011$_H$	✓	✓	—	✓			+044$_H$
	ICCSIH1IR_1		INTCSIH1IR_1[*6]	CSIH1 receive status interrupt	CSIH1	Edge		✓	✓	—	✓			
18	ICADCA0I0	FFFE EA24$_H$	INTADCA0I0	ADCA0 scan group 1 (SG1) end interrupt	ADCA0	Edge	1012$_H$	✓	✓	—	✓			+048$_H$
19	ICADCA0I1	FFFE EA26$_H$	INTADCA0I1	ADCA0 scan group 2 (SG2) end interrupt	ADCA0	Edge	1013$_H$	✓	✓	—	✓			+04C$_H$
20	ICADCA0I2	FFFE EA28$_H$	INTADCA0I2[*6]	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge	1014$_H$	✓	✓	—	✓			+050$_H$
	ICCSIH0IJC_1		INTCSIH0IJC_1[*6]	CSIH0 job completion interrupt	CSIH0	Edge		✓	✓	—	✓			
21	ICDCUTDI	FFFE EA2A$_H$	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	1015$_H$	✓	✓	—	✓			+054$_H$
22	ICRCANGERR0	FFFE EA2C$_H$	INTRCANGERR0	CAN global error interrupt	RCFDC0	Level	1016$_H$	✓	✓	—	✓			+058$_H$
23	ICRCANGRECC0	FFFE EA2E$_H$	INTRCANGRECC0	CAN receive FIFO interrupt	RCFDC0	Level	1017$_H$	✓	✓	—	✓			+05C$_H$
24	ICRCAN0ERR	FFFE EA30$_H$	INTRCAN0ERR	CAN0 error interrupt	RCFDC0	Level	1018$_H$	✓	✓	—	✓			+060$_H$
25	ICRCAN0REC	FFFE EA32$_H$	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1019$_H$	✓	✓	—	✓			+064$_H$
26	ICRCAN0TRX	FFFE EA34$_H$	INTRCAN0TRX	CAN0 transmit interrupt	RCFDC0	Level	101A$_H$	✓	✓	—	✓			+068$_H$
27	ICCSIG0IC	FFFE EA36$_H$	INTCSIG0IC[*6]	CSIG0 communication status interrupt	CSIG0	Edge	101B$_H$	✓	✓	—	✓			+06C$_H$
	ICCSIH1IRE_1		INTCSIH1IRE_1[*6]	CSIH1 communication error interrupt	CSIH1	Edge		✓	✓	—	✓			
28	ICCSIG0IR	FFFE EA38$_H$	INTCSIG0IR[*6]	CSIG0 receive status interrupt	CSIG0	Edge	101C$_H$	✓	✓	—	✓			+070$_H$
	ICCSIH1IJC_1		INTCSIH1IJC_1[*6]	CSIH1 job completion interrupt	CSIH1	Edge		✓	✓	—	✓			
29	ICCSIH0IC	FFFE EA3A$_H$	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	101D$_H$	✓	✓	—	✓			+074$_H$
30	ICCSIH0IR	FFFE EA3C$_H$	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	101E$_H$	✓	✓	—	✓			+078$_H$
31	ICCSIH0IRE	FFFE EA3E$_H$	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	101F$_H$	✓	✓	—	✓			+07C$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register											Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2						RINT = 0	RINT = 1	
32	ICCSIH0IJC	FFFF B040$_H$	INTCSIH0IJC*6	CSIH0 job completion interrupt	CSIH0	Edge	1020$_H$	✓	✓	—	✓	*3	*4	+080$_H$
	ICADCA0I2_2		INTADCA0I2_2*6	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge		✓	✓	—	✓			
33	ICRLIN30	FFFF B042$_H$	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1021$_H$	✓	✓	—	✓			+084$_H$
34	ICRLIN30UR0	FFFF B044$_H$	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	1022$_H$	✓	✓	—	✓			+088$_H$
35	ICRLIN30UR1	FFFF B046$_H$	INTRLIN30UR1	RLIN30 receive complete interrupt	RLIN30	Edge	1023$_H$	✓	✓	—	✓			+08C$_H$
36	ICRLIN30UR2	FFFF B048$_H$	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	1024$_H$	✓	✓	—	✓			+090$_H$
37	ICP0	FFFF B04A$_H$	INTP0	External interrupt	Port	Edge	1025$_H$	✓	✓	—	✓			+094$_H$
38	ICP1	FFFF B04C$_H$	INTP1	External interrupt	Port	Edge	1026$_H$	✓	✓	—	✓			+098$_H$
39	ICP2	FFFF B04E$_H$	INTP2	External interrupt	Port	Edge	1027$_H$	✓	✓	—	✓			+09C$_H$
40	ICWDTA0	FFFF B050$_H$	INTWDTA0	WDTA0 75% interrupt	WDTA0	Edge	1028$_H$	✓	✓	—	✓			+0A0$_H$
41	ICWDTA1	FFFF B052$_H$	INTWDTA1	WDTA1 75% interrupt	WDTA1	Edge	1029$_H$	✓	✓	—	✓			+0A4$_H$
42	ICWDTA2	FFFF B054$_H$	INTWDTA2	WDTA2 75% interrupt	WDTA2	Edge	102A$_H$	✓	✓	—	✓			+0A8$_H$
43	ICP3	FFFF B056$_H$	INTP3	External interrupt	Port	Edge	102B$_H$	✓	✓	—	✓			+0AC$_H$
44	ICP4	FFFF B058$_H$	INTP4	External interrupt	Port	Edge	102C$_H$	✓	✓	—	✓			+0B0$_H$
45	ICP5	FFFF B05A$_H$	INTP5	External interrupt	Port	Edge	102D$_H$	✓	✓	—	✓			+0B4$_H$
46	ICP10	FFFF B05C$_H$	INTP10	External interrupt	Port	Edge	102E$_H$	✓	✓	—	✓			+0B8$_H$
47	ICP11	FFFF B05E$_H$	INTP11	External interrupt	Port	Edge	102F$_H$	✓	✓	—	✓			+0BC$_H$
48	ICTAUD0I1	FFFF B060$_H$	INTTAUD0I1	Interrupt for TAUD0 channel 1	TAUD0	Edge	1030$_H$	✓	✓	—	✓			+0C0$_H$
49	ICTAUD0I3	FFFF B062$_H$	INTTAUD0I3	Interrupt for TAUD0 channel 3	TAUD0	Edge	1031$_H$	✓	✓	—	✓			+0C4$_H$
50	ICTAUD0I5	FFFF B064$_H$	INTTAUD0I5	Interrupt for TAUD0 channel 5	TAUD0	Edge	1032$_H$	✓	✓	—	✓			+0C8$_H$
51	ICTAUD0I7	FFFF B066$_H$	INTTAUD0I7	Interrupt for TAUD0 channel 7	TAUD0	Edge	1033$_H$	✓	✓	—	✓			+0CC$_H$
52	ICTAUD0I9	FFFF B068$_H$	INTTAUD0I9	Interrupt for TAUD0 channel 9	TAUD0	Edge	1034$_H$	✓	✓	—	✓			+0D0$_H$
53	ICTAUD0I11	FFFF B06A$_H$	INTTAUD0I11	Interrupt for TAUD0 channel 11	TAUD0	Edge	1035$_H$	✓	✓	—	✓			+0D4$_H$
54	ICTAUD0I13	FFFF B06C$_H$	INTTAUD0I13	Interrupt for TAUD0 channel 13	TAUD0	Edge	1036$_H$	✓	✓	—	✓			+0D8$_H$
55	ICTAUD0I15	FFFF B06E$_H$	INTTAUD0I15	Interrupt for TAUD0 channel 15	TAUD0	Edge	1037$_H$	✓	✓	—	✓			+0DC$_H$
56	ICADCA0ERR	FFFF B070$_H$	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	1038$_H$	✓	✓	—	✓			+0E0$_H$
57	ICCSIG0IRE	FFFF B072$_H$	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1039$_H$	✓	✓	—	✓			+0E4$_H$
58	ICRLIN20	FFFF B074$_H$	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	103A$_H$	✓	✓	—	✓			+0E8$_H$
59	ICRLIN21	FFFF B076$_H$	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	103B$_H$	✓	✓	—	✓			+0EC$_H$
60	ICDMA0	FFFF B078$_H$	INTDMA0	DMA00 transfer completion	PDMA0	Edge	103C$_H$	✓	✓	—	✓			+0F0$_H$
61	ICDMA1	FFFF B07A$_H$	INTDMA1	DMA01 transfer completion	PDMA0	Edge	103D$_H$	✓	✓	—	✓			+0F4$_H$
62	ICDMA2	FFFF B07C$_H$	INTDMA2	DMA02 transfer completion	PDMA0	Edge	103E$_H$	✓	✓	—	✓			+0F8$_H$
63	ICDMA3	FFFF B07E$_H$	INTDMA3	DMA03 transfer completion	PDMA0	Edge	103F$_H$	✓	✓	—	✓			+0FC$_H$
64	ICDMA4	FFFF B080$_H$	INTDMA4	DMA04 transfer completion	PDMA0	Edge	1040$_H$	✓	✓	—	✓			+100$_H$
65	ICDMA5	FFFF B082$_H$	INTDMA5	DMA05 transfer completion	PDMA0	Edge	1041$_H$	✓	✓	—	✓			+104$_H$
66	ICDMA6	FFFF B084$_H$	INTDMA6	DMA06 transfer completion	PDMA0	Edge	1042$_H$	✓	✓	—	✓			+108$_H$
67	ICDMA7	FFFF B086$_H$	INTDMA7	DMA07 transfer completion	PDMA0	Edge	1043$_H$	✓	✓	—	✓			+10C$_H$
68	ICDMA8	FFFF B088$_H$	INTDMA8	DMA10 transfer completion	PDMA0	Edge	1044$_H$	✓	✓	—	✓			+110$_H$
69	ICDMA9	FFFF B08A$_H$	INTDMA9	DMA11 transfer completion	PDMA0	Edge	1045$_H$	✓	✓	—	✓			+114$_H$
70	ICDMA10	FFFF B08C$_H$	INTDMA10	DMA12 transfer completion	PDMA0	Edge	1046$_H$	✓	✓	—	✓			+118$_H$
71	ICDMA11	FFFF B08E$_H$	INTDMA11	DMA13 transfer completion	PDMA0	Edge	1047$_H$	✓	✓	—	✓			+11C$_H$
72	ICDMA12	FFFF B090$_H$	INTDMA12	DMA14 transfer completion	PDMA0	Edge	1048$_H$	✓	✓	—	✓			+120$_H$
73	ICDMA13	FFFF B092$_H$	INTDMA13	DMA15 transfer completion	PDMA0	Edge	1049$_H$	✓	✓	—	✓			+124$_H$
74	ICDMA14	FFFF B094$_H$	INTDMA14	DMA16 transfer completion	PDMA0	Edge	104A$_H$	✓	✓	—	✓			+128$_H$
75	ICDMA15	FFFF B096$_H$	INTDMA15	DMA17 transfer completion	PDMA0	Edge	104B$_H$	✓	✓	—	✓			+12C$_H$
76	ICRIIC0TI	FFFF B098$_H$	INTRIIC0TI	RIIC0 transmit data empty interrupt	RIIC0	Edge	104C$_H$	✓	✓	—	✓			+130$_H$
77	ICRIIC0EE	FFFF B09A$_H$	INTRIIC0EE	RIIC0 receive error/event interrupt	RIIC0	Level	104D$_H$	✓	✓	—	✓			+134$_H$
78	ICRIIC0RI	FFFF B09C$_H$	INTRIIC0RI	RIIC0 receive complete interrupt	RIIC0	Edge	104E$_H$	✓	✓	—	✓			+138$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register											Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2						RINT = 0	RINT = 1	
79	ICRIIC0TEI	FFFF B09E$_H$	INTRIIC0TEI	RIIC0 transmit complete interrupt	RIIC0	Level	104F$_H$	✓	✓	—	✓	*3	*4	+13C$_H$
80	ICTAUJ0I0	FFFF B0A0$_H$	INTTAUJ0I0	Interrupt for TAUJ0 channel 0	TAUJ0	Edge	1050$_H$	✓	✓	—	✓			+140$_H$
81	ICTAUJ0I1	FFFF B0A2$_H$	INTTAUJ0I1	Interrupt for TAUJ0 channel 1	TAUJ0	Edge	1051$_H$	✓	✓	—	✓			+144$_H$
82	ICTAUJ0I2	FFFF B0A4$_H$	INTTAUJ0I2	Interrupt for TAUJ0 channel 2	TAUJ0	Edge	1052$_H$	✓	✓	—	✓			+148$_H$
83	ICTAUJ0I3	FFFF B0A6$_H$	INTTAUJ0I3	Interrupt for TAUJ0 channel 3	TAUJ0	Edge	1053$_H$	✓	✓	—	✓			+14C$_H$
84	ICOSTM0	FFFF B0A8$_H$	INTOSTM0*9	OSTM0 interrupt	OSTM0	Edge	1054$_H$	✓	✓	—	✓			+150$_H$
85	ICENCA0IOV	FFFF B0AA$_H$	INTENCA0IOV	ENCA0 overflow interrupt	ENCA0	Edge	1055$_H$	✓	✓	—	✓			+154$_H$
86	ICENCA0IUD	FFFF B0AC$_H$	INTENCA0IUD	ENCA0 underflow interrupt	ENCA0	Edge	1056$_H$	✓	✓	—	✓			+158$_H$
87	ICENCA0I0	FFFF B0AE$_H$	INTENCA0I0	ENCA0 match/capture interrupt 0	ENCA0	Edge	1057$_H$	✓	✓	—	✓			+15C$_H$
88	ICENCA0I1	FFFF B0B0$_H$	INTENCA0I1	ENCA0 match/capture interrupt 1	ENCA0	Edge	1058$_H$	✓	✓	—	✓			+160$_H$
89	ICENCA0IEC	FFFF B0B2$_H$	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	Edge	1059$_H$	✓	✓	—	✓			+164$_H$
90	ICKR0	FFFF B0B4$_H$	INTKR0	KR0 key interrupt	KR0	Edge	105A$_H$	✓	✓	—	✓			+168$_H$
91	ICQFULL	FFFF B0B6$_H$	INTQFULL	PWSA0 queue full interrupt	PWSA0	Edge	105B$_H$	✓	✓	—	✓			+16C$_H$
92	ICPWGAG00	FFFF B0B8$_H$	INTPWGAG00*7	PWGA interrupt group 00 (PWGA0 to PWGA31)	PWGA0-31	Level	105C$_H$	✓	✓	—	✓			+170$_H$
93	ICPWGAG01	FFFF B0BA$_H$	INTPWGAG01*7	PWGA interrupt group 01 (PWGA32 to PWGA63)	PWGA32-63	Level	105D$_H$	✓	✓	—	✓			+174$_H$
94	ICPWGAG02	FFFF B0BC$_H$	INTPWGAG02*7	PWGA interrupt group 02 (PWGA64 to PWGA95)	PWGA64-95	Level	105E$_H$	✓	✓	—	✓			+178$_H$
95	ICPWGAG10	FFFF B0BE$_H$	INTPWGAG10*7	PWGA interrupt group 10 (PWGA0 to PWGA31)	PWGA0-31	Level	105F$_H$	✓	✓	—	✓			+17C$_H$
96	ICPWGAG11	FFFF B0C0$_H$	INTPWGAG11*7	PWGA interrupt group 11 (PWGA32 to PWGA63)	PWGA32-63	Level	1060$_H$	✓	✓	—	✓			+180$_H$
97	ICPWGAG12	FFFF B0C2$_H$	INTPWGAG12*7	PWGA interrupt group 12 (PWGA64 to PWGA95)	PWGA64-95	Level	1061$_H$	✓	✓	—	✓			+184$_H$
98	ICCSIG4IC	FFFF B0C4$_H$	INTCSIG4IC	CSIG4 communication status interrupt	CSIG4	Edge	1062$_H$	✓	✓	—	✓			+188$_H$
99	ICCSIG4IR	FFFF B0C6$_H$	INTCSIG4IR	CSIG4 receive status interrupt	CSIG4	Edge	1063$_H$	✓	✓	—	✓			+18C$_H$
100	ICCSIG4IRE	FFFF B0C8$_H$	INTCSIG4IRE	CSIG4 communication error interrupt	CSIG4	Edge	1064$_H$	✓	✓	—	✓			+190$_H$
101	ICDMA32	FFFF B0CA$_H$	INTDMA32	DMA00 transfer completion	PDMA1	Edge	1065$_H$	✓	✓	—	✓			+194$_H$
102	ICDMA33	FFFF B0CC$_H$	INTDMA33	DMA01 transfer completion	PDMA1	Edge	1066$_H$	✓	✓	—	✓			+198$_H$
103	ICDMA34	FFFF B0CE$_H$	INTDMA34	DMA02 transfer completion	PDMA1	Edge	1067$_H$	✓	✓	—	✓			+19C$_H$
104	Reserved	FFFF B0D0$_H$					1068$_H$	—	—	—	—			+1A0$_H$
105	Reserved	FFFF B0D2$_H$					1069$_H$	—	—	—	—			+1A4$_H$
106	ICDMA35	FFFF B0D4$_H$	INTDMA35	DMA03 transfer completion	PDMA1	Edge	106A$_H$	✓	✓	—	✓			+1A8$_H$
107	Reserved	FFFF B0D6$_H$					106B$_H$	—	—	—	—			+1AC$_H$
108	Reserved	FFFF B0D8$_H$					106C$_H$	—	—	—	—			+1B0$_H$
109	Reserved	FFFF B0DA$_H$					106D$_H$	—	—	—	—			+1B4$_H$
110	ICFLERR	FFFF B0DC$_H$	INTFLERR*11	Flash sequencer end error interrupt	FACI	Level	106E$_H$	✓	✓	—	✓			+1B8$_H$
111	ICFLENDNM	FFFF B0DE$_H$	INTFLENDNM*11	Flash sequencer end interrupt	FACI	Edge	106F$_H$	✓	✓	—	✓			+1BC$_H$
112	ICCWEND	FFFF B0E0$_H$	INTCWEND	LPS0 port polling end interrupt	LPS0	Edge	1070$_H$	✓	✓	—	✓			+1C0$_H$
113	ICRCAN1ERR	FFFF B0E2$_H$	INTRCAN1ERR	CAN1 error interrupt	RCFDC0	Level	1071$_H$	✓	✓	—	✓			+1C4$_H$
114	ICRCAN1REC	FFFF B0E4$_H$	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1072$_H$	✓	✓	—	✓			+1C8$_H$
115	ICRCAN1TRX	FFFF B0E6$_H$	INTRCAN1TRX	CAN1 transmit interrupt	RCFDC0	Level	1073$_H$	✓	✓	—	✓			+1CC$_H$
116	ICCSIH1IC	FFFF B0E8$_H$	INTCSIH1IC*6	CSIH1 communication status interrupt	CSIH1	Edge	1074$_H$	✓	✓	—	✓			+1D0$_H$
	ICTAPA0IPEK0_2		INTTAPA0IPEK0_2*6	TAPA0 peak interrupt 0	TAPA0	Edge		✓	✓	—	✓			
117	ICCSIH1IR	FFFF B0EA$_H$	INTCSIH1IR*6	CSIH1 receive status interrupt	CSIH1	Edge	1075$_H$	✓	✓	—	✓			+1D4$_H$
	ICTAPA0IVLY0_2		INTTAPA0IVLY0_2*6	TAPA0 valley interrupt 0	TAPA0	Edge		✓	✓	—	✓			
118	ICCSIH1IRE	FFFF B0EC$_H$	INTCSIH1IRE*6	CSIH1 communication error interrupt	CSIH1	Edge	1076$_H$	✓	✓	—	✓			+1D8$_H$
	ICCSIG0IC_2		INTCSIG0IC_2*6	CSIG0 communication status interrupt	CSIG0	Edge		✓	✓	—	✓			

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register											Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2						RINT = 0	RINT = 1	
119	ICCSIH1IJC	FFFF B0EE$_H$	INTCSIH1IJC*6	CSIH1 job completion interrupt	CSIH1	Edge	1077$_H$	✓	✓	—	✓	*3	*4	+1DC$_H$
	ICCSIG0IR_2		INTCSIG0IR_2*6	CSIG0 receive status interrupt	CSIG0	Edge		✓	✓	—	✓			
120	ICRLIN31	FFFF B0F0$_H$	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1078$_H$	✓	✓	—	✓			+1E0$_H$
121	ICRLIN31UR0	FFFF B0F2$_H$	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1079$_H$	✓	✓	—	✓			+1E4$_H$
122	ICRLIN31UR1	FFFF B0F4$_H$	INTRLIN31UR1	RLIN31 receive complete interrupt	RLIN31	Edge	107A$_H$	✓	✓	—	✓			+1E8$_H$
123	ICRLIN31UR2	FFFF B0F6$_H$	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	107B$_H$	✓	✓	—	✓			+1EC$_H$
124	ICCSIH4IC	FFFF B0F8$_H$	INTCSIH4IC	CSIH4 communication status interrupt	CSIH4	Edge	107C$_H$	✓	✓	—	✓			+1F0$_H$
125	ICCSIH4IR	FFFF B0FA$_H$	INTCSIH4IR	CSIH4 receive status interrupt	CSIH4	Edge	107D$_H$	✓	✓	—	✓			+1F4$_H$
126	ICCSIH4IRE	FFFF B0FC$_H$	INTCSIH4IRE	CSIH4 communication error interrupt	CSIH4	Edge	107E$_H$	✓	✓	—	✓			+1F8$_H$
127	ICCSIH4IJC	FFFF B0FE$_H$	INTCSIH4IJC	CSIH4 job completion interrupt	CSIH4	Edge	107F$_H$	✓	✓	—	✓			+1FC$_H$
128	ICP6	FFFF B100$_H$	INTP6	External interrupt	Port	Edge	1080$_H$	✓	✓	—	✓			+200$_H$
129	ICP7	FFFF B102$_H$	INTP7	External interrupt	Port	Edge	1081$_H$	✓	✓	—	✓			+204$_H$
130	ICP8	FFFF B104$_H$	INTP8	External interrupt	Port	Edge	1082$_H$	✓	✓	—	✓			+208$_H$
131	ICP12	FFFF B106$_H$	INTP12	External interrupt	Port	Edge	1083$_H$	✓	✓	—	✓			+20C$_H$
132	ICCSIH2IC	FFFF B108$_H$	INTCSIH2IC*6	CSIH2 communication status interrupt	CSIH2	Edge	1084$_H$	✓	✓	—	✓			+210$_H$
	ICTAUD0I0_2		INTTAUD0I0_2*6	Interrupt for TAUD0 channel 0	TAUD0	Edge		✓	✓	—	✓			
133	ICCSIH2IR	FFFF B10A$_H$	INTCSIH2IR*6	CSIH2 receive status interrupt	CSIH2	Edge	1085$_H$	✓	✓	—	✓			+214$_H$
	ICTAUD0I4_2		INTTAUD0I4_2*6	Interrupt for TAUD0 channel 4	TAUD0	Edge		✓	✓	—	✓			
134	ICCSIH2IRE	FFFF B10C$_H$	INTCSIH2IRE*6	CSIH2 communication error interrupt	CSIH2	Edge	1086$_H$	✓	✓	—	✓			+218$_H$
	ICTAUD0I6_2		INTTAUD0I6_2*6	Interrupt for TAUD0 channel 6	TAUD0	Edge		✓	✓	—	✓			
135	ICCSIH2IJC	FFFF B10E$_H$	INTCSIH2IJC*6	CSIH2 job completion interrupt	CSIH2	Edge	1087$_H$	✓	✓	—	✓			+21C$_H$
	ICTAUD0I8_2		INTTAUD0I8_2*6	Interrupt for TAUD0 channel 8	TAUD0	Edge		✓	✓	—	✓			
136	Reserved	FFFF B110$_H$					1088$_H$	—	—	—	—			+220$_H$
137	Reserved	FFFF B112$_H$					1089$_H$	—	—	—	—			+224$_H$
138	Reserved	FFFF B114$_H$					108A$_H$	—	—	—	—			+228$_H$
139	Reserved	FFFF B116$_H$					108B$_H$	—	—	—	—			+22C$_H$
140	Reserved	FFFF B118$_H$					108C$_H$	—	—	—	—			+230$_H$
141	Reserved	FFFF B11A$_H$					108D$_H$	—	—	—	—			+234$_H$
142	ICTAUB0I0	FFFF B11C$_H$	INTTAUB0I0	Interrupt for TAUB0 channel 0	TAUB0	Edge	108E$_H$	✓	✓	—	✓			+238$_H$
143	ICTAUB0I1	FFFF B11E$_H$	INTTAUB0I1	Interrupt for TAUB0 channel 1	TAUB0	Edge	108F$_H$	✓	✓	—	✓			+23C$_H$
144	ICTAUB0I2	FFFF B120$_H$	INTTAUB0I2	Interrupt for TAUB0 channel 2	TAUB0	Edge	1090$_H$	✓	✓	—	✓			+240$_H$
145	ICTAUB0I3	FFFF B122$_H$	INTTAUB0I3	Interrupt for TAUB0 channel 3	TAUB0	Edge	1091$_H$	✓	✓	—	✓			+244$_H$
146	ICTAUB0I4	FFFF B124$_H$	INTTAUB0I4	Interrupt for TAUB0 channel 4	TAUB0	Edge	1092$_H$	✓	✓	—	✓			+248$_H$
147	ICTAUB0I5	FFFF B126$_H$	INTTAUB0I5	Interrupt for TAUB0 channel 5	TAUB0	Edge	1093$_H$	✓	✓	—	✓			+24C$_H$
148	ICTAUB0I6	FFFF B128$_H$	INTTAUB0I6	Interrupt for TAUB0 channel 6	TAUB0	Edge	1094$_H$	✓	✓	—	✓			+250$_H$
149	ICTAUB0I7	FFFF B12A$_H$	INTTAUB0I7	Interrupt for TAUB0 channel 7	TAUB0	Edge	1095$_H$	✓	✓	—	✓			+254$_H$
150	ICTAUB0I8	FFFF B12C$_H$	INTTAUB0I8	Interrupt for TAUB0 channel 8	TAUB0	Edge	1096$_H$	✓	✓	—	✓			+258$_H$
151	ICTAUB0I9	FFFF B12E$_H$	INTTAUB0I9	Interrupt for TAUB0 channel 9	TAUB0	Edge	1097$_H$	✓	✓	—	✓			+25C$_H$
152	ICTAUB0I10	FFFF B130$_H$	INTTAUB0I10	Interrupt for TAUB0 channel 10	TAUB0	Edge	1098$_H$	✓	✓	—	✓			+260$_H$
153	ICTAUB0I11	FFFF B132$_H$	INTTAUB0I11	Interrupt for TAUB0 channel 11	TAUB0	Edge	1099$_H$	✓	✓	—	✓			+264$_H$
154	ICTAUB0I12	FFFF B134$_H$	INTTAUB0I12	Interrupt for TAUB0 channel 12	TAUB0	Edge	109A$_H$	✓	✓	—	✓			+268$_H$
155	ICTAUB0I13	FFFF B136$_H$	INTTAUB0I13	Interrupt for TAUB0 channel 13	TAUB0	Edge	109B$_H$	✓	✓	—	✓			+26C$_H$
156	ICTAUB0I14	FFFF B138$_H$	INTTAUB0I14	Interrupt for TAUB0 channel 14	TAUB0	Edge	109C$_H$	✓	✓	—	✓			+270$_H$
157	ICTAUB0I15	FFFF B13A$_H$	INTTAUB0I15	Interrupt for TAUB0 channel 15	TAUB0	Edge	109D$_H$	✓	✓	—	✓			+274$_H$
158	ICCSIH3IC	FFFF B13C$_H$	INTCSIH3IC*6	CSIH3 communication status interrupt	CSIH3	Edge	109E$_H$	✓	✓	—	✓			+278$_H$
	ICTAUD0I2_2		INTTAUD0I2_2*6	Interrupt for TAUD0 channel 2	TAUD0	Edge		✓	✓	—	✓			
159	ICCSIH3IR	FFFF B13E$_H$	INTCSIH3IR*6	CSIH3 receive status interrupt	CSIH3	Edge	109F$_H$	✓	✓	—	✓			+27C$_H$
	ICTAUD0I10_2		INTTAUD0I10_2*6	Interrupt for TAUD0 channel 10	TAUD0	Edge		✓	✓	—	✓			

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2						Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
160	ICCSIH3IRE	FFFF B140$_H$	INTCSIH3IRE*6	CSIH3 communication error interrupt	CSIH3	Edge	10A0$_H$	✓	✓	—	✓	*3	*4	+280$_H$
	ICTAUD0I12_2		INTTAUD0I12_2*6	Interrupt for TAUD0 channel 12	TAUD0	Edge		✓	✓	—	✓			
161	ICCSIH3IJC	FFFF B142$_H$	INTCSIH3IJC*6	CSIH3 job completion interrupt	CSIH3	Edge	10A1$_H$	✓	✓	—	✓			+284$_H$
	ICTAUD0I14_2		INTTAUD0I14_2*6	Interrupt for TAUD0 channel 14	TAUD0	Edge		✓	✓	—	✓			
162	ICRLIN22	FFFF B144$_H$	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	10A2$_H$	✓	✓	—	✓			+288$_H$
163	ICRLIN23	FFFF B146$_H$	INTRLIN23	RLIN23 interrupt	RLIN240	Edge	10A3$_H$	✓	✓	—	✓			+28C$_H$
164	ICRLIN32	FFFF B148$_H$	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	10A4$_H$	✓	✓	—	✓			+290$_H$
165	ICRLIN32UR0	FFFF B14A$_H$	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	10A5$_H$	✓	✓	—	✓			+294$_H$
166	ICRLIN32UR1	FFFF B14C$_H$	INTRLIN32UR1	RLIN32 receive complete interrupt	RLIN32	Edge	10A6$_H$	✓	✓	—	✓			+298$_H$
167	ICRLIN32UR2	FFFF B14E$_H$	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	10A7$_H$	✓	✓	—	✓			+29C$_H$
168	ICTAUJ1I0	FFFF B150$_H$	INTTAUJ1I0	Interrupt for TAUJ1 channel 0	TAUJ1	Edge	10A8$_H$	✓	✓	—	✓			+2A0$_H$
169	ICTAUJ1I1	FFFF B152$_H$	INTTAUJ1I1	Interrupt for TAUJ1 channel 1	TAUJ1	Edge	10A9$_H$	✓	✓	—	✓			+2A4$_H$
170	ICTAUJ1I2	FFFF B154$_H$	INTTAUJ1I2	Interrupt for TAUJ1 channel 2	TAUJ1	Edge	10AA$_H$	✓	✓	—	✓			+2A8$_H$
171	ICTAUJ1I3	FFFF B156$_H$	INTTAUJ1I3	Interrupt for TAUJ1 channel 3	TAUJ1	Edge	10AB$_H$	✓	✓	—	✓			+2AC$_H$
172	Reserved	FFFF B158$_H$					10AC$_H$	—	—	—	—			+2B0$_H$
173	ICFLXA0FDA	FFFF B15A$_H$	INTFLXA0FDA	FIFO transfer interrupt	FLXA0	Level	10AD$_H$	✓	✓	—	✓			+2B4$_H$
174	ICFLXA0FW	FFFF B15C$_H$	INTFLXA0FW	FIFO transfer warning interrupt	FLXA0	Level	10AE$_H$	✓	✓	—	✓			+2B8$_H$
175	ICFLXA0IQE	FFFF B15E$_H$	INTFLXA0IQE	Input queue empty interrupt	FLXA0	Level	10AF$_H$	✓	✓	—	✓			+2BC$_H$
176	ICFLXA0IQF	FFFF B160$_H$	INTFLXA0IQF	Input queue full interrupt	FLXA0	Level	10B0$_H$	✓	✓	—	✓			+2C0$_H$
177	ICFLXA0OT	FFFF B162$_H$	INTFLXA0OT	Output transfer end interrupt	FLXA0	Level	10B1$_H$	✓	✓	—	✓			+2C4$_H$
178	ICFLXA0OW	FFFF B164$_H$	INTFLXA0OW	Output transfer warning interrupt	FLXA0	Level	10B2$_H$	✓	✓	—	✓			+2C8$_H$
179	ICFLXA0LINE0	FFFF B166$_H$	INTFLXA0LINE0	FlexRay0 interrupt	FLXA0	Level	10B3$_H$	✓	✓	—	✓			+2CC$_H$
180	ICFLXA0LINE1	FFFF B168$_H$	INTFLXA0LINE1	FlexRay1 interrupt	FLXA0	Level	10B4$_H$	✓	✓	—	✓			+2D0$_H$
181	ICFLXA0TIM0	FFFF B16A$_H$	INTFLXA0TIM0	Timer 0 interrupt	FLXA0	Level	10B5$_H$	✓	✓	—	✓			+2D4$_H$
182	ICFLXA0TIM1	FFFF B16C$_H$	INTFLXA0TIM1	Timer 1 interrupt	FLXA0	Level	10B6$_H$	✓	✓	—	✓			+2D8$_H$
183	ICFLXA0TIM2	FFFF B16E$_H$	INTFLXA0TIM2	Timer 2 interrupt	FLXA0	Level	10B7$_H$	✓	✓	—	✓			+2DC$_H$
184	ICRLIN212	FFFF B170$_H$	INTRLIN212	RLIN212 interrupt	RLIN243	Edge	10B8$_H$	—	—	—	✓			+2E0$_H$
185	ICRLIN213	FFFF B172$_H$	INTRLIN213	RLIN213 interrupt	RLIN243	Edge	10B9$_H$	—	—	—	✓			+2E4$_H$
186	ICRLIN214	FFFF B174$_H$	INTRLIN214	RLIN214 interrupt	RLIN243	Edge	10BA$_H$	—	—	—	✓			+2E8$_H$
187	ICRLIN215	FFFF B176$_H$	INTRLIN215	RLIN215 interrupt	RLIN243	Edge	10BB$_H$	—	—	—	✓			+2EC$_H$
188	ICDMA36	FFFF B178$_H$	INTDMA36	DMA04 transfer completion	PDMA1	Edge	10BC$_H$	✓	✓	—	✓			+2F0$_H$
189	ICDMA37	FFFF B17A$_H$	INTDMA37	DMA05 transfer completion	PDMA1	Edge	10BD$_H$	✓	✓	—	✓			+2F4$_H$
190	ICDMA38	FFFF B17C$_H$	INTDMA38	DMA06 transfer completion	PDMA1	Edge	10BE$_H$	✓	✓	—	✓			+2F8$_H$
191	ICDMA39	FFFF B17E$_H$	INTDMA39	DMA07 transfer completion	PDMA1	Edge	10BF$_H$	✓	✓	—	✓			+2FC$_H$
192	ICDMA40	FFFF B180$_H$	INTDMA40	DMA10 transfer completion	PDMA1	Edge	10C0$_H$	✓	✓	—	✓			+300$_H$
193	ICDMA41	FFFF B182$_H$	INTDMA41	DMA11 transfer completion	PDMA1	Edge	10C1$_H$	✓	✓	—	✓			+304$_H$
194	ICDMA42	FFFF B184$_H$	INTDMA42	DMA12 transfer completion	PDMA1	Edge	10C2$_H$	✓	✓	—	✓			+308$_H$
195	ICDMA43	FFFF B186$_H$	INTDMA43	DMA13 transfer completion	PDMA1	Edge	10C3$_H$	✓	✓	—	✓			+30C$_H$
196	ICDMA44	FFFF B188$_H$	INTDMA44	DMA14 transfer completion	PDMA1	Edge	10C4$_H$	✓	✓	—	✓			+310$_H$
197	ICDMA45	FFFF B18A$_H$	INTDMA45	DMA15 transfer completion	PDMA1	Edge	10C5$_H$	✓	✓	—	✓			+314$_H$
198	ICDMA46	FFFF B18C$_H$	INTDMA46	DMA16 transfer completion	PDMA1	Edge	10C6$_H$	✓	✓	—	✓			+318$_H$
199	ICDMA47	FFFF B18E$_H$	INTDMA47	DMA17 transfer completion	PDMA1	Edge	10C7$_H$	✓	✓	—	✓			+31C$_H$
200	ICDMA48	FFFF B190$_H$	INTDMA48	DMA20 transfer completion	PDMA1	Edge	10C8$_H$	✓	✓	—	✓			+320$_H$
201	ICDMA49	FFFF B192$_H$	INTDMA49	DMA21 transfer completion	PDMA1	Edge	10C9$_H$	✓	✓	—	✓			+324$_H$
202	ICDMA50	FFFF B194$_H$	INTDMA50	DMA22 transfer completion	PDMA1	Edge	10CA$_H$	✓	✓	—	✓			+328$_H$
203	ICDMA51	FFFF B196$_H$	INTDMA51	DMA23 transfer completion	PDMA1	Edge	10CB$_H$	✓	✓	—	✓			+32C$_H$
204	ICDMA52	FFFF B198$_H$	INTDMA52	DMA24 transfer completion	PDMA1	Edge	10CC$_H$	✓	✓	—	✓			+330$_H$
205	ICP9	FFFF B19A$_H$	INTP9	External interrupt	Port	Edge	10CD$_H$	✓	✓	—	✓			+334$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2						Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
206	ICP13	FFFF B19C$_H$	INTP13	External interrupt	Port	Edge	10CE$_H$	✓	✓	—	✓	*3	*4	+338$_H$
207	ICP14	FFFF B19E$_H$	INTP14	External interrupt	Port	Edge	10CF$_H$	✓	✓	—	✓			+33C$_H$
208	ICP15	FFFF B1A0$_H$	INTP15	External interrupt	Port	Edge	10D0$_H$	✓	✓	—	✓			+340$_H$
209	ICRTCA01S	FFFF B1A2$_H$	INTRTCA01S	RTCA0 1-second interval interrupt	RTCA0	Edge	10D1$_H$	✓	✓	—	✓			+344$_H$
210	ICRTCA0AL	FFFF B1A4$_H$	INTRTCA0AL	RTCA0 alarm interrupt	RTCA0	Edge	10D2$_H$	✓	✓	—	✓			+348$_H$
211	ICRTCA0R	FFFF B1A6$_H$	INTRTCA0R	RTCA0 fixed interval interrupt	RTCA0	Edge	10D3$_H$	✓	✓	—	✓			+34C$_H$
212	ICADCA1ERR	FFFF B1A8$_H$	INTADCA1ERR	ADCA1 error interrupt	ADCA1	Edge	10D4$_H$	✓	✓	—	✓			+350$_H$
213	ICADCA1I0	FFFF B1AA$_H$	INTADCA1I0	ADCA1 scan group 1 (SG1) end interrupt	ADCA1	Edge	10D5$_H$	✓	✓	—	✓			+354$_H$
214	ICADCA1I1	FFFF B1AC$_H$	INTADCA1I1	ADCA1 scan group 2 (SG2) end interrupt	ADCA1	Edge	10D6$_H$	✓	✓	—	✓			+358$_H$
215	ICADCA1I2	FFFF B1AE$_H$	INTADCA1I2	ADCA1 scan group 3 (SG3) end interrupt	ADCA1	Edge	10D7$_H$	✓	✓	—	✓			+35C$_H$
216	ICDMA53	FFFF B1B0$_H$	INTDMA53	DMA25 transfer completion	PDMA1	Edge	10D8$_H$	✓	✓	—	✓			+360$_H$
217	ICRCAN2ERR	FFFF B1B2$_H$	INTRCAN2ERR	CAN2 error interrupt	RCFDC0	Level	10D9$_H$	✓	✓	—	✓			+364$_H$
218	ICRCAN2REC	FFFF B1B4$_H$	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DA$_H$	✓	✓	—	✓			+368$_H$
219	ICRCAN2TRX	FFFF B1B6$_H$	INTRCAN2TRX	CAN2 transmit interrupt	RCFDC0	Level	10DB$_H$	✓	✓	—	✓			+36C$_H$
220	ICRCAN3ERR	FFFF B1B8$_H$	INTRCAN3ERR	CAN3 error interrupt	RCFDC0	Level	10DC$_H$	✓	✓	—	✓			+370$_H$
221	ICRCAN3REC	FFFF B1BA$_H$	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DD$_H$	✓	✓	—	✓			+374$_H$
222	ICRCAN3TRX	FFFF B1BC$_H$	INTRCAN3TRX	CAN3 transmit interrupt	RCFDC0	Level	10DE$_H$	✓	✓	—	✓			+378$_H$
223	ICCSIG1IC	FFFF B1BE$_H$	INTCSIG1IC	CSIG1 communication status interrupt	CSIG1	Edge	10DF$_H$	✓	✓	—	✓			+37C$_H$
224	ICCSIG1IR	FFFF B1C0$_H$	INTCSIG1IR	CSIG1 receive status interrupt	CSIG1	Edge	10E0$_H$	✓	✓	—	✓			+380$_H$
225	ICCSIG1IRE	FFFF B1C2$_H$	INTCSIG1IRE	CSIG1 communication error interrupt	CSIG1	Edge	10E1$_H$	✓	✓	—	✓			+384$_H$
226	ICRLIN24	FFFF B1C4$_H$	INTRLIN24	RLIN24 interrupt	RLIN241	Edge	10E2$_H$	✓	✓	—	✓			+388$_H$
227	ICRLIN25	FFFF B1C6$_H$	INTRLIN25	RLIN25 interrupt	RLIN241	Edge	10E3$_H$	✓	✓	—	✓			+38C$_H$
228	ICRLIN33	FFFF B1C8$_H$	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10E4$_H$	✓	✓	—	✓			+390$_H$
229	ICRLIN33UR0	FFFF B1CA$_H$	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10E5$_H$	✓	✓	—	✓			+394$_H$
230	ICRLIN33UR1	FFFF B1CC$_H$	INTRLIN33UR1	RLIN33 receive complete interrupt	RLIN33	Edge	10E6$_H$	✓	✓	—	✓			+398$_H$
231	ICRLIN33UR2	FFFF B1CE$_H$	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10E7$_H$	✓	✓	—	✓			+39C$_H$
232	ICRLIN34	FFFF B1D0$_H$	INTRLIN34	RLIN34 interrupt	RLIN34	Edge	10E8$_H$	✓	✓	—	✓			+3A0$_H$
233	ICRLIN34UR0	FFFF B1D2$_H$	INTRLIN34UR0	RLIN34 transmit interrupt	RLIN34	Edge	10E9$_H$	✓	✓	—	✓			+3A4$_H$
234	ICRLIN34UR1	FFFF B1D4$_H$	INTRLIN34UR1	RLIN34 receive complete interrupt	RLIN34	Edge	10EA$_H$	✓	✓	—	✓			+3A8$_H$
235	ICRLIN34UR2	FFFF B1D6$_H$	INTRLIN34UR2	RLIN34 status interrupt	RLIN34	Edge	10EB$_H$	✓	✓	—	✓			+3AC$_H$
236	ICRLIN35	FFFF B1D8$_H$	INTRLIN35	RLIN35 interrupt	RLIN35	Edge	10EC$_H$	✓	✓	—	✓			+3B0$_H$
237	ICRLIN35UR0	FFFF B1DA$_H$	INTRLIN35UR0	RLIN35 transmit interrupt	RLIN35	Edge	10ED$_H$	✓	✓	—	✓			+3B4$_H$
238	ICRLIN35UR1	FFFF B1DC$_H$	INTRLIN35UR1	RLIN35 receive complete interrupt	RLIN35	Edge	10EE$_H$	✓	✓	—	✓			+3B8$_H$
239	ICRLIN35UR2	FFFF B1DE$_H$	INTRLIN35UR2	RLIN35 status interrupt	RLIN35	Edge	10EF$_H$	✓	✓	—	✓			+3BC$_H$
240	ICRIIC1TI	FFFF B1E0$_H$	INTRIIC1TI	RIIC1 transmit data empty interrupt	RIIC1	Edge	10F0$_H$	✓	✓	—	✓			+3C0$_H$
241	ICRIIC1EE	FFFF B1E2$_H$	INTRIIC1EE	RIIC1 receive error/event interrupt	RIIC1	Level	10F1$_H$	✓	✓	—	✓			+3C4$_H$
242	ICRIIC1RI	FFFF B1E4$_H$	INTRIIC1RI	RIIC1 receive complete interrupt	RIIC1	Edge	10F2$_H$	✓	✓	—	✓			+3C8$_H$
243	ICRIIC1TEI	FFFF B1E6$_H$	INTRIIC1TEI	RIIC1 transmit complete interrupt	RIIC1	Level	10F3$_H$	✓	✓	—	✓			+3CC$_H$
244	ICRCAN8ERR	FFFF B1E8$_H$	INTRCAN8ERR	CAN8 error interrupt	RCFDC1	Level	10F4$_H$	—	—	—	✓			+3D0$_H$
245	ICRCAN8REC	FFFF B1EA$_H$	INTRCAN8REC	CAN8 transmit/receive FIFO receive complete interrupt	RCFDC1	Level	10F5$_H$	—	—	—	✓			+3D4$_H$
246	ICRCAN8TRX	FFFF B1EC$_H$	INTRCAN8TRX	CAN8 transmit interrupt	RCFDC1	Level	10F6$_H$	—	—	—	✓			+3D8$_H$
247	ICRCAN9ERR	FFFF B1EE$_H$	INTRCAN9ERR	CAN9 error interrupt	RCFDC1	Level	10F7$_H$	—	—	—	✓			+3DC$_H$
248	ICRCAN9REC	FFFF B1F0$_H$	INTRCAN9REC	CAN9 transmit/receive FIFO receive complete interrupt	RCFDC1	Level	10F8$_H$	—	—	—	✓			+3E0$_H$
249	ICRCAN9TRX	FFFF B1F2$_H$	INTRCAN9TRX	CAN9 transmit interrupt	RCFDC1	Level	10F9$_H$	—	—	—	✓			+3E4$_H$
250	ICRCAN10ERR	FFFF B1F4$_H$	INTRCAN10ERR	CAN10 error interrupt	RCFDC1	Level	10FA$_H$	—	—	—	✓			+3E8$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2						Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
251	ICRCAN10REC	FFFF B1F6$_H$	INTRCAN10REC	CAN10 transmit/receive FIFO receive complete interrupt	RCFDC1	Level	10FB$_H$	—	—	—	✓	*3	*4	+3EC$_H$
252	ICRCAN10TRX	FFFF B1F8$_H$	INTRCAN10TRX	CAN10 transmit interrupt	RCFDC1	Level	10FC$_H$	—	—	—	✓			+3F0$_H$
253	ICRCAN11ERR	FFFF B1FA$_H$	INTRCAN11ERR	CAN11 error interrupt	RCFDC1	Level	10FD$_H$	—	—	—	✓			+3F4$_H$
254	ICRCAN11REC	FFFF B1FC$_H$	INTRCAN11REC	CAN11 transmit/receive FIFO receive complete interrupt	RCFDC1	Level	10FE$_H$	—	—	—	✓			+3F8$_H$
255	ICRCAN11TRX	FFFF B1FE$_H$	INTRCAN11TRX	CAN11 transmit interrupt	RCFDC1	Level	10FF$_H$	—	—	—	✓			+3FC$_H$
256	ICTAUB1I0	FFFF B200$_H$	INTTAUB1I0	Interrupt for TAUB1 channel 0	TAUB1	Edge	1100$_H$	✓	✓	—	✓			+400$_H$
257	ICTAUB1I1	FFFF B202$_H$	INTTAUB1I1	Interrupt for TAUB1 channel 1	TAUB1	Edge	1101$_H$	✓	✓	—	✓			+404$_H$
258	ICTAUB1I2	FFFF B204$_H$	INTTAUB1I2	Interrupt for TAUB1 channel 2	TAUB1	Edge	1102$_H$	✓	✓	—	✓			+408$_H$
259	ICTAUB1I3	FFFF B206$_H$	INTTAUB1I3	Interrupt for TAUB1 channel 3	TAUB1	Edge	1103$_H$	✓	✓	—	✓			+40C$_H$
260	ICTAUB1I4	FFFF B208$_H$	INTTAUB1I4	Interrupt for TAUB1 channel 4	TAUB1	Edge	1104$_H$	✓	✓	—	✓			+410$_H$
261	ICTAUB1I5	FFFF B20A$_H$	INTTAUB1I5	Interrupt for TAUB1 channel 5	TAUB1	Edge	1105$_H$	✓	✓	—	✓			+414$_H$
262	ICTAUB1I6	FFFF B20C$_H$	INTTAUB1I6	Interrupt for TAUB1 channel 6	TAUB1	Edge	1106$_H$	✓	✓	—	✓			+418$_H$
263	ICTAUB1I7	FFFF B20E$_H$	INTTAUB1I7	Interrupt for TAUB1 channel 7	TAUB1	Edge	1107$_H$	✓	✓	—	✓			+41C$_H$
264	ICTAUB1I8	FFFF B210$_H$	INTTAUB1I8	Interrupt for TAUB1 channel 8	TAUB1	Edge	1108$_H$	✓	✓	—	✓			+420$_H$
265	ICTAUB1I9	FFFF B212$_H$	INTTAUB1I9	Interrupt for TAUB1 channel 9	TAUB1	Edge	1109$_H$	✓	✓	—	✓			+424$_H$
266	ICTAUB1I10	FFFF B214$_H$	INTTAUB1I10	Interrupt for TAUB1 channel 10	TAUB1	Edge	110A$_H$	✓	✓	—	✓			+428$_H$
267	ICTAUB1I11	FFFF B216$_H$	INTTAUB1I11	Interrupt for TAUB1 channel 11	TAUB1	Edge	110B$_H$	✓	✓	—	✓			+42C$_H$
268	ICTAUB1I12	FFFF B218$_H$	INTTAUB1I12	Interrupt for TAUB1 channel 12	TAUB1	Edge	110C$_H$	✓	✓	—	✓			+430$_H$
269	ICTAUB1I13	FFFF B21A$_H$	INTTAUB1I13	Interrupt for TAUB1 channel 13	TAUB1	Edge	110D$_H$	✓	✓	—	✓			+434$_H$
270	ICTAUB1I14	FFFF B21C$_H$	INTTAUB1I14	Interrupt for TAUB1 channel 14	TAUB1	Edge	110E$_H$	✓	✓	—	✓			+438$_H$
271	ICTAUB1I15	FFFF B21E$_H$	INTTAUB1I15	Interrupt for TAUB1 channel 15	TAUB1	Edge	110F$_H$	✓	✓	—	✓			+43C$_H$
272	ICRCAN4ERR	FFFF B220$_H$	INTRCAN4ERR	CAN4 error interrupt	RCFDC0	Level	1110$_H$	✓	✓	—	✓			+440$_H$
273	ICRCAN4REC	FFFF B222$_H$	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1111$_H$	✓	✓	—	✓			+444$_H$
274	ICRCAN4TRX	FFFF B224$_H$	INTRCAN4TRX	CAN4 transmit interrupt	RCFDC0	Level	1112$_H$	✓	✓	—	✓			+448$_H$
275	ICRLIN26	FFFF B226$_H$	INTRLIN26	RLIN26 interrupt	RLIN241	Edge	1113$_H$	✓	✓	—	✓			+44C$_H$
276	ICRLIN27	FFFF B228$_H$	INTRLIN27	RLIN27 interrupt	RLIN241	Edge	1114$_H$	✓	✓	—	✓			+450$_H$
277	ICTAUJ2I0	FFFF B22A$_H$	INTTAUJ2I0	Interrupt for TAUJ2 channel 0	TAUJ2	Edge	1115$_H$	✓	✓	—	✓			+454$_H$
278	ICTAUJ2I1	FFFF B22C$_H$	INTTAUJ2I1	Interrupt for TAUJ2 channel 1	TAUJ2	Edge	1116$_H$	✓	✓	—	✓			+458$_H$
279	ICTAUJ2I2	FFFF B22E$_H$	INTTAUJ2I2	Interrupt for TAUJ2 channel 2	TAUJ2	Edge	1117$_H$	✓	✓	—	✓			+45C$_H$
280	ICTAUJ2I3	FFFF B230$_H$	INTTAUJ2I3	Interrupt for TAUJ2 channel 3	TAUJ2	Edge	1118$_H$	✓	✓	—	✓			+460$_H$
281	ICTAUJ3I0	FFFF B232$_H$	INTTAUJ3I0	Interrupt for TAUJ3 channel 0	TAUJ3	Edge	1119$_H$	✓	✓	—	✓			+464$_H$
282	ICTAUJ3I1	FFFF B234$_H$	INTTAUJ3I1	Interrupt for TAUJ3 channel 1	TAUJ3	Edge	111A$_H$	✓	✓	—	✓			+468$_H$
283	ICTAUJ3I2	FFFF B236$_H$	INTTAUJ3I2	Interrupt for TAUJ3 channel 2	TAUJ3	Edge	111B$_H$	✓	✓	—	✓			+46C$_H$
284	ICTAUJ3I3	FFFF B238$_H$	INTTAUJ3I3	Interrupt for TAUJ3 channel 3	TAUJ3	Edge	111C$_H$	✓	✓	—	✓			+470$_H$
285	ICRLIN28	FFFF B23A$_H$	INTRLIN28	RLIN28 interrupt	RLIN242	Edge	111D$_H$	✓	✓	—	✓			+474$_H$
286	ICRLIN29	FFFF B23C$_H$	INTRLIN29	RLIN29 interrupt	RLIN242	Edge	111E$_H$	✓	✓	—	✓			+478$_H$
287	ICRCAN5ERR	FFFF B23E$_H$	INTRCAN5ERR	CAN5 error interrupt	RCFDC0	Level	111F$_H$	✓	✓	—	✓			+47C$_H$
288	ICRCAN5REC	FFFF B240$_H$	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1120$_H$	✓	✓	—	✓			+480$_H$
289	ICRCAN5TRX	FFFF B242$_H$	INTRCAN5TRX	CAN5 transmit interrupt	RCFDC0	Level	1121$_H$	✓	✓	—	✓			+484$_H$
290	ICDMA54	FFFF B244$_H$	INTDMA54	DMA26 transfer completion	PDMA1	Edge	1122$_H$	✓	✓	—	✓			+488$_H$
291	ICDMA55	FFFF B246$_H$	INTDMA55	DMA27 transfer completion	PDMA1	Edge	1123$_H$	✓	✓	—	✓			+48C$_H$
292	ICDMA56	FFFF B248$_H$	INTDMA56	DMA30 transfer completion	PDMA1	Edge	1124$_H$	✓	✓	—	✓			+490$_H$
293	ICDMA57	FFFF B24A$_H$	INTDMA57	DMA31 transfer completion	PDMA1	Edge	1125$_H$	✓	✓	—	✓			+494$_H$
294	ICDMA58	FFFF B24C$_H$	INTDMA58	DMA32 transfer completion	PDMA1	Edge	1126$_H$	✓	✓	—	✓			+498$_H$
295	ICDMA59	FFFF B24E$_H$	INTDMA59	DMA33 transfer completion	PDMA1	Edge	1127$_H$	✓	✓	—	✓			+49C$_H$
296	ICDMA60	FFFF B250$_H$	INTDMA60	DMA34 transfer completion	PDMA1	Edge	1128$_H$	✓	✓	—	✓			+4A0$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register Name	Control Register Address	Name	Source	Unit	Detection Type*2						Direct Jumping to an Address RINT = 0	Direct Jumping to an Address RINT = 1	Reference to a Table*5
297	ICDMA61	FFFF B252$_H$	INTDMA61	DMA35 transfer completion	PDMA1	Edge	1129$_H$	✓	✓	—	✓	*3	*4	+4A4$_H$
298	ICDMA16	FFFF B254$_H$	INTDMA16	DMA20 transfer completion	PDMA0	Edge	112A$_H$	✓	✓	—	✓			+4A8$_H$
299	ICDMA17	FFFF B256$_H$	INTDMA17	DMA21 transfer completion	PDMA0	Edge	112B$_H$	✓	✓	—	✓			+4AC$_H$
300	ICDMA18	FFFF B258$_H$	INTDMA18	DMA22 transfer completion	PDMA0	Edge	112C$_H$	✓	✓	—	✓			+4B0$_H$
301	ICDMA19	FFFF B25A$_H$	INTDMA19	DMA23 transfer completion	PDMA0	Edge	112D$_H$	✓	✓	—	✓			+4B4$_H$
302	ICDMA20	FFFF B25C$_H$	INTDMA20	DMA24 transfer completion	PDMA0	Edge	112E$_H$	✓	✓	—	✓			+4B8$_H$
303	ICDMA21	FFFF B25E$_H$	INTDMA21	DMA25 transfer completion	PDMA0	Edge	112F$_H$	✓	✓	—	✓			+4BC$_H$
304	ICDMA22	FFFF B260$_H$	INTDMA22	DMA26 transfer completion	PDMA0	Edge	1130$_H$	✓	✓	—	✓			+4C0$_H$
305	ICDMA23	FFFF B262$_H$	INTDMA23	DMA27 transfer completion	PDMA0	Edge	1131$_H$	✓	✓	—	✓			+4C4$_H$
306	ICDMA24	FFFF B264$_H$	INTDMA24	DMA30 transfer completion	PDMA0	Edge	1132$_H$	✓	✓	—	✓			+4C8$_H$
307	ICDMA25	FFFF B266$_H$	INTDMA25	DMA31 transfer completion	PDMA0	Edge	1133$_H$	✓	✓	—	✓			+4CC$_H$
308	ICDMA26	FFFF B268$_H$	INTDMA26	DMA32 transfer completion	PDMA0	Edge	1134$_H$	✓	✓	—	✓			+4D0$_H$
309	ICDMA27	FFFF B26A$_H$	INTDMA27	DMA33 transfer completion	PDMA0	Edge	1135$_H$	✓	✓	—	✓			+4D4$_H$
310	ICDMA28	FFFF B26C$_H$	INTDMA28	DMA34 transfer completion	PDMA0	Edge	1136$_H$	✓	✓	—	✓			+4D8$_H$
311	ICDMA29	FFFF B26E$_H$	INTDMA29	DMA35 transfer completion	PDMA0	Edge	1137$_H$	✓	✓	—	✓			+4DC$_H$
312	ICDMA30	FFFF B270$_H$	INTDMA30	DMA36 transfer completion	PDMA0	Edge	1138$_H$	✓	✓	—	✓			+4E0$_H$
313	ICDMA31	FFFF B272$_H$	INTDMA31	DMA37 transfer completion	PDMA0	Edge	1139$_H$	✓	✓	—	✓			+4E4$_H$
314	ICOSTM5	FFFF B274$_H$	INTOSTM5*10	OSTM5 interrupt	OSTM5	Edge	113A$_H$	✓	✓	—	✓			+4E8$_H$
315	ICETNB0DATA	FFFF B276$_H$	INTETNB0DATA	Data related interrupt	ETNB0	Level	113B$_H$	✓	✓	—	✓			+4EC$_H$
316	ICETNB0ERR	FFFF B278$_H$	INTETNB0ERR	Error related interrupt	ETNB0	Level	113C$_H$	✓	✓	—	✓			+4F0$_H$
317	ICETNB0MNG	FFFF B27A$_H$	INTETNB0MNG	Management related interrupt	ETNB0	Level	113D$_H$	✓	✓	—	✓			+4F4$_H$
318	ICETNB0MAC	FFFF B27C$_H$	INTETNB0MAC	MAC interrupt	ETNB0	Level	113E$_H$	✓	✓	—	✓			+4F8$_H$
319	ICRCANGERR1	FFFF B27E$_H$	INTRCANGERR1	CAN global error interrupt	RCFDC1	Level	113F$_H$	—	—	—	✓			+4FC$_H$
320	ICRCANGRECC1	FFFF B280$_H$	INTRCANGRECC1	CAN receive FIFO interrupt	RCFDC1	Level	1140$_H$	—	—	—	✓			+500$_H$
321	ICRCAN6ERR	FFFF B282$_H$	INTRCAN6ERR	CAN6 error interrupt	RCFDC0	Level	1141$_H$	✓	✓	—	✓			+504$_H$
322	ICRCAN6REC	FFFF B284$_H$	INTRCAN6REC	CAN6 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1142$_H$	✓	✓	—	✓			+508$_H$
323	ICRCAN6TRX	FFFF B286$_H$	INTRCAN6TRX	CAN6 transmit interrupt	RCFDC0	Level	1143$_H$	✓	✓	—	✓			+50C$_H$
324	ICRLIN210	FFFF B288$_H$	INTRLIN210	RLIN210 interrupt	RLIN242	Edge	1144$_H$	—	✓	—	✓			+510$_H$
325	ICRLIN211	FFFF B28A$_H$	INTRLIN211	RLIN211 interrupt	RLIN242	Edge	1145$_H$	—	✓	—	✓			+514$_H$
326	ICCSIG2IC	FFFF B28C$_H$	INTCSIG2IC	CSIG2 communication status interrupt	CSIG2	Edge	1146$_H$	✓	✓	—	✓			+518$_H$
327	ICCSIG2IR	FFFF B28E$_H$	INTCSIG2IR	CSIG2 receive status interrupt	CSIG2	Edge	1147$_H$	✓	✓	—	✓			+51C$_H$
328	ICCSIG2IRE	FFFF B290$_H$	INTCSIG2IRE	CSIG2 communication error interrupt	CSIG2	Edge	1148$_H$	✓	✓	—	✓			+520$_H$
329	ICCSIG3IC	FFFF B292$_H$	INTCSIG3IC	CSIG3 communication status interrupt	CSIG3	Edge	1149$_H$	✓	✓	—	✓			+524$_H$
330	ICCSIG3IR	FFFF B294$_H$	INTCSIG3IR	CSIG3 receive status interrupt	CSIG3	Edge	114A$_H$	✓	✓	—	✓			+528$_H$
331	ICCSIG3IRE	FFFF B296$_H$	INTCSIG3IRE	CSIG3 communication error interrupt	CSIG3	Edge	114B$_H$	✓	✓	—	✓			+52C$_H$
332	ICRCAN7ERR	FFFF B298$_H$	INTRCAN7ERR	CAN7 error interrupt	RCFDC0	Level	114C$_H$	✓	✓	—	✓			+530$_H$
333	ICRCAN7REC	FFFF B29A$_H$	INTRCAN7REC	CAN7 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	114D$_H$	✓	✓	—	✓			+534$_H$
334	ICRCAN7TRX	FFFF B29C$_H$	INTRCAN7TRX	CAN7 transmit interrupt	RCFDC0	Level	114E$_H$	✓	✓	—	✓			+538$_H$
335	Reserved	FFFF B29E$_H$					114F$_H$	—	—	—	—			+53C$_H$
336	Reserved	FFFF B2A0$_H$					1150$_H$	—	—	—	—			+540$_H$
337	Reserved	FFFF B2A2$_H$					1151$_H$	—	—	—	—			+544$_H$
338	Reserved	FFFF B2A4$_H$					1152$_H$	—	—	—	—			+548$_H$
339	Reserved	FFFF B2A6$_H$					1153$_H$	—	—	—	—			+54C$_H$
340	Reserved	FFFF B2A8$_H$					1154$_H$	—	—	—	—			+550$_H$
341	Reserved	FFFF B2AA$_H$					1155$_H$	—	—	—	—			+554$_H$
342	Reserved	FFFF B2AC$_H$					1156$_H$	—	—	—	—			+558$_H$
343	Reserved	FFFF B2AE$_H$					1157$_H$	—	—	—	—			+55C$_H$

Table 7A.4 EI Level Maskable Interrupt Sources

Interrupt			Interrupt Request				Exception Source Code	176 Pins	233 Pins	272 Pins	324 Pins	Handler Address (Offset)^{*8}		
Channel No.^{*1}	Control Register					Detection Type^{*2}						Direct Jumping to an Address		Reference to a Table^{*5}
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
344	Reserved	FFFF B2B0$_H$					1158$_H$	—	—	—	—	*3	*4	+560$_H$
345	Reserved	FFFF B2B2$_H$					1159$_H$	—	—	—	—			+564$_H$
346	Reserved	FFFF B2B4$_H$					115A$_H$	—	—	—	—			+568$_H$
347	ICSENT0SI	FFFF B2B6$_H$	INTSENT0SI	Status interrupt for RSENT0	RSENT0	Level	115B$_H$	✓	✓	—	✓			+56C$_H$
348	ICSENT0RI	FFFF B2B8$_H$	INTSENT0RI	Receive interrupt for RSENT0	RSENT0	Edge	115C$_H$	✓	✓	—	✓			+570$_H$
349	ICSENT1SI	FFFF B2BA$_H$	INTSENT1SI	Status interrupt for RSENT1	RSENT1	Level	115D$_H$	✓	✓	—	✓			+574$_H$
350	ICSENT1RI	FFFF B2BC$_H$	INTSENT1RI	Receive interrupt for RSENT1	RSENT1	Edge	115E$_H$	✓	✓	—	✓			+578$_H$
351	Reserved	FFFF B2BE$_H$					115F$_H$	—	—	—	—			+57C$_H$
352	Reserved	FFFF B2C0$_H$					1160$_H$	—	—	—	—			+580$_H$
353	Reserved	FFFF B2C2$_H$					1161$_H$	—	—	—	—			+584$_H$
354	ICMMCA0	FFFF B2C4$_H$	INTMMCA0	interrupt for MMCA0	MMCA0	Level	1162$_H$	—	—	—	✓			+588$_H$
355	ICDMA62	FFFF B2C6$_H$	INTDMA62	DMA36 transfer completion	PDMA1	Edge	1163$_H$	✓	✓	—	✓			+58C$_H$
356	ICDPE	FFFF B2C8$_H$	INTDPE	LPS0 digital port error interrupt	LPS0	Level	1164$_H$	✓	✓	—	✓			+590$_H$
357	ICAPE	FFFF B2CA$_H$	INTAPE	LPS0 analog port error interrupt	LPS0	Level	1165$_H$	✓	✓	—	✓			+594$_H$
358	Reserved	FFFF B2CC$_H$					1166$_H$	—	—	—	—			+598$_H$
359	Reserved	FFFF B2CE$_H$					1167$_H$	—	—	—	—			+59C$_H$
360	ICRLIN36	FFFF B2D0$_H$	INTRLIN36	RLIN36 interrupt	RLIN36	Edge	1168$_H$	✓	✓	—	✓			+5A0$_H$
361	ICRLIN36UR0	FFFF B2D2$_H$	INTRLIN36UR0	RLIN36 transmit interrupt	RLIN36	Edge	1169$_H$	✓	✓	—	✓			+5A4$_H$
362	ICRLIN36UR1	FFFF B2D4$_H$	INTRLIN36UR1	RLIN36 receive complete interrupt	RLIN36	Edge	116A$_H$	✓	✓	—	✓			+5A8$_H$
363	ICRLIN36UR2	FFFF B2D6$_H$	INTRLIN36UR2	RLIN36 status interrupt	RLIN36	Edge	116B$_H$	✓	✓	—	✓			+5AC$_H$
364	ICRLIN37	FFFF B2D8$_H$	INTRLIN37	RLIN37 interrupt	RLIN37	Edge	116C$_H$	✓	✓	—	✓			+5B0$_H$
365	ICRLIN37UR0	FFFF B2DA$_H$	INTRLIN37UR0	RLIN37 transmit interrupt	RLIN37	Edge	116D$_H$	✓	✓	—	✓			+5B4$_H$
366	ICRLIN37UR1	FFFF B2DC$_H$	INTRLIN37UR1	RLIN37 receive complete interrupt	RLIN37	Edge	116E$_H$	✓	✓	—	✓			+5B8$_H$
367	ICRLIN37UR2	FFFF B2DE$_H$	INTRLIN37UR2	RLIN37 status interrupt	RLIN37	Edge	116F$_H$	✓	✓	—	✓			+5BC$_H$
368	ICP16	FFFF B2E0$_H$	INTP16	External interrupt	Port	Edge	1170$_H$	✓	✓	—	✓			+5C0$_H$
369	ICP17	FFFF B2E2$_H$	INTP17	External interrupt	Port	Edge	1171$_H$	✓	✓	—	✓			+5C4$_H$
370	ICP18	FFFF B2E4$_H$	INTP18	External interrupt	Port	Edge	1172$_H$	✓	✓	—	✓			+5C8$_H$
371	ICP19	FFFF B2E6$_H$	INTP19	External interrupt	Port	Edge	1173$_H$	✓	✓	—	✓			+5CC$_H$
372	ICP20	FFFF B2E8$_H$	INTP20	External interrupt	Port	Edge	1174$_H$	✓	✓	—	✓			+5D0$_H$
373	ICP21	FFFF B2EA$_H$	INTP21	External interrupt	Port	Edge	1175$_H$	✓	✓	—	✓			+5D4$_H$
374	ICP22	FFFF B2EC$_H$	INTP22	External interrupt	Port	Edge	1176$_H$	✓	✓	—	✓			+5D8$_H$
375	ICP23	FFFF B2EE$_H$	INTP23	External interrupt	Port	Edge	1177$_H$	✓	✓	—	✓			+5DC$_H$
376	ICGRZF	FFFF B2F0$_H$	INTGRZF	interrupt for GRZF	GRZF	Edge	1178$_H$	✓	✓	—	✓			+5E0$_H$
377	ICETNB1DATA	FFFF B2F2$_H$	INTETNB1DATA	Data related interrupt	ETNB1	Level	1179$_H$	—	—	—	✓			+5E4$_H$
378	ICETNB1ERR	FFFF B2F4$_H$	INTETNB1ERR	Error related interrupt	ETNB1	Level	117A$_H$	—	—	—	✓			+5E8$_H$
379	ICETNB1MNG	FFFF B2F6$_H$	INTETNB1MNG	Management related interrupt	ETNB1	Level	117B$_H$	—	—	—	✓			+5EC$_H$
380	ICETNB1MAC	FFFF B2F8$_H$	INTETNB1MAC	MAC interrupt	ETNB1	Level	117C$_H$	—	—	—	✓			+5F0$_H$
381	ICDMA63	FFFF B2FA$_H$	INTDMA63	DMA37 transfer completion	PDMA1	Edge	117D$_H$	✓	✓	—	✓			+5F4$_H$

Note 1. Each interrupt is connected to INTC1 channel 0 to 31 and INTC2 channel 32 to 381.

Note 2. This indicates whether an interrupt source is detected at the level or edge. This also affects the value after reset of an EI level interrupt control register. For details, see **Section 7A.4.2, ICxxx — EI Level Interrupt Control Registers**. For detection at level, an interrupt source is cleared by accessing to the register that retains an interrupt source. The procedure shown in **Section 3A.5.1, Synchronization of Store Instruction Completion and Subsequent Instruction Execution** are required to reflect the result of the register update to the subsequent instruction.

Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100$_H$ to 1F0$_H$ according to the priority (0 to 15).

Note 4. Irrespective of the priority, offset addresses are uniformly +100$_H$.

Note 5. The table reference method uses a table for reading an exception handler address for each interrupt channel, and it extracts handler address by referencing that table. Table reference position is determined by the following formula.
Exception handler address read position = INTBP register + channel number × 4 bytes

Note 6. The same interrupt source is assigned to different interrupt channels. For details, see **Section 7A.5.2, SELB_INTC1 — INTC1 Interrupt Select Register**.

Note 7. 32 interrupt sources are assigned to the same interrupt channel. For details, see **Section 37, PWM Output/Diagnostic (PWM-Diag)**.

Note 8. For details, see **Section 7A.10, Exception Handler Address**.

Note 9. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it in both ways at the same time is not possible.
It is used as FEINT when OSTM0 functions as the TSU (timing supervision unit). It is used as EIINT when OSTM0 functions as anything other than the TSU function.

Note 10. INTOSTM5 can operate as an EIINT or FEINT interrupt, but using it in both ways at the same time is not possible.
It is used as FEINT when OSTM5 functions as the TSU (timing supervision unit). It is used as EIINT when OSTM5 functions as anything other than the TSU function.

Note 11. For details on the interrupt source, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

7A.2.2 FE Level Non-Maskable Interrupt Sources

7A.2.2.1 List of Registers

The following table lists the FE Level Non-Maskable Interrupt Sources registers.

Table 7A.5 List of Registers

Module Name	Register Name	Symbol	Address
ECON_NMI	FENMI factor register	WDTNMIF	FFC0 0000$_H$
	WDTNMI factor clear register	WDTNMIFC	FFC0 0008$_H$

7A.2.2.2 WDTNMIF — FENMI Factor Register

This register contains information about which source has generated the FE level non-maskable interrupt (FENMI). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	WDTA2 NMIF	WDTA1 NMIF	WDTA0 NMIF	TNMIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7A.6 WDTNMIF Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	WDTA2NMIF	WDTA2NMI flag 0: No WDTA2NMI occurred 1: WDTA2NMI has occurred
2	WDTA1NMIF	WDTA1NMI flag 0: No WDTA1NMI occurred 1: WDTA1NMI has occurred
1	WDTA0NMIF	WDTA0NMI flag 0: No WDTA0NMI occurred 1: WDTA0NMI has occurred
0	TNMIF	Input signal flag from the NMI pin 0: No TNMI occurred 1: TNMI has occurred

7A.2.2.3 WDTNMIFC — WDTNMI Factor Clear Register

This register clears the FE level non-maskable interrupt flags of the WDTNMIF register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0008$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	WDTA2 NMIFC	WDTA1 NMIFC	WDTA0 NMIFC	TNMIF C
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 7A.7 WDTNMIFC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	WDTA2NMIFC	WDTA2NMIF flag clear 0: — 1: Clear
2	WDTA1NMIFC	WDTA1NMIF flag clear 0: — 1: Clear
1	WDTA0NMIFC	WDTA0NMIF flag clear 0: — 1: Clear
0	TNMIFC	TNMIF flag clear 0: — 1: Clear

7A.2.3 FE Level Maskable Interrupt Sources

7A.2.3.1 List of Registers

The following table lists the FE Level Maskable Interrupt Sources registers.

Table 7A.8 List of Registers

Module Name	Register Name	Symbol	Address
ECON_FEINT	FEINT factor register	FEINTF	FFC0 0100$_H$
	FEINT factor mask register	FEINTFMSK	FFC0 0104$_H$
	FEINT factor clear register	FEINTFC	FFC0 0108$_H$

7A.2.3.2 FEINTF — FEINT Factor Register

This register contains information about which source has generated the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0100$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIF	ECCMMCRAMA0FEIF	ECCETH1FEIF	OSTM9FEIF	OSTM8FEIF	OSTM7FEIF	OSTM6FEIF	OSTM5FEIF	ECCETH0FEIF	ECCCNFDRAM1FEIF	OSTM4FEIF	OSTM3FEIF	OSTM2FEIF	OSTM1FEIF	GUARDFEIF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIF	OSTM0FEIF	ECCRAMFEIF	ECCFLI0FEIF	ECCCSIH4FEIF	ECCCSIH3FEIF	ECCCSIH2FEIF	ECCCSIH1FEIF	ECCCSIH0FEIF	ECCCNFDRAM0FEIF	ECCFLRAMFEIF	ECCEEP0FEIF	—	—	—	LVILFEIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7A.9 FEINTF Register Contents

Bit Position	Bit Name	Function
31	DMAFEIF	INTDMAERR interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
30	ECCMMCA0RAMFEIF*1	INTECCMMCA0RAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
29	ECCETH1FEIF*1	INTECCETH1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
28	OSTM9FEIF	INTOSTM9_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
27	OSTM8FEIF	INTOSTM8_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Table 7A.9 FEINTF Register Contents

Bit Position	Bit Name	Function
26	OSTM7FEIF	INTOSTM7_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
25	OSTM6FEIF	INTOSTM6_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
24	OSTM5FEIF	INTOSTM5_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
23	ECCETH0 FEIF	INTECCETH0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
22	ECCCNFDRAM1FEIF*1	INTECCCNFDRAM1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
21	OSTM4FEIF	INTOSTM4_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
20	OSTM3FEIF	INTOSTM3_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
19	OSTM2FEIF	INTOSTM2_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
18	OSTM1FEIF	INTOSTM1_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
17	GUARDFEIF	INTGUARD interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
16	Reserved	When read, the value after reset is returned.
15	LVIHFEIF	INTLVIH interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
14	OSTM0 FEIF	INTOSTM0_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
13	ECCRAM FEIF	INTECCRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
12	ECCFLI0 FEIF	INTECCFLI0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
11	ECCCSIH4 FEIF	INTECCCSIH4 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
10	ECCCSIH3 FEIF	INTECCCSIH3 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Table 7A.9 FEINTF Register Contents

Bit Position	Bit Name	Function
9	ECCCSIH2 FEIF	INTECCCSIH2 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
8	ECCCSIH1 FEIF	INTECCCSIH1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
7	ECCCSIH0 FEIF	INTECCCSIH0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
6	ECCCNFDRAM0 FEIF	INTECCCNFDRAM0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
5	ECCFLRAM FEIF*1	INTECCFLRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
4	ECCEEP0 FEIF	INTECCEEP0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
3 to 1	Reserved	When read, the value after reset is returned.
0	LVILFEIF	INTLVIL interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Note 1. For the supported products, see **Table 7A.3, FE Level Maskable Interrupt Requests (RH850/F1KH-D8)**.

7A.2.3.3 FEINTFMSK — FEINT Factor Mask Register

This register masks the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFC0 0104_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIFMSK	ECCMMCA0RAMFEIFMSK	ECCETH1FEIFMSK	OSTM9FEIFMSK	OSTM8FEIFMSK	OSTM7FEIFMSK	OSTM6FEIFMSK	OSTM5FEIFMSK	ECCETH0FEIFMSK	ECCCNFDRAM1FEIFMSK	OSTM4FEIFMSK	OSTM3FEIFMSK	OSTM2FEIFMSK	OSTM1FEIFMSK	GUARDFEIFMSK	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIFMSK	OSTM0FEIFMSK	ECCRAMFEIFMSK	ECCFLI0FEIFMSK	ECCCSIH4FEIFMSK	ECCCSIH3FEIFMSK	ECCCSIH2FEIFMSK	ECCCSIH1FEIFMSK	ECCCSIH0FEIFMSK	ECCCNFDRAM0FEIFMSK	ECCFLRAMFEIFMSK	ECCEEP0FEIFMSK	—	—	—	LVILFEIFMSK
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 7A.10 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
31	DMAFEIFMSK	INTDMAERR interrupt mask 0: Not masked 1: Masked
30	ECCMMCA0RAMFEIFMSK*1	INTECCMMCA0RAM interrupt mask 0: Not masked 1: Masked
29	ECCETH1FEIFMSK*1	INTECCETH1 interrupt mask 0: Not masked 1: Masked
28	OSTM9FEIFMSK	INTOSTM9_FE interrupt mask 0: Not masked 1: Masked
27	OSTM8FEIFMSK	INTOSTM8_FE interrupt mask 0: Not masked 1: Masked
26	OSTM7FEIFMSK	INTOSTM7_FE interrupt mask 0: Not masked 1: Masked
25	OSTM6FEIFMSK	INTOSTM6_FE interrupt mask 0: Not masked 1: Masked
24	OSTM5FEIFMSK	INTOSTM5_FE interrupt mask 0: Not masked 1: Masked
23	ECCETH0FEIFMSK	INTECCETH0 interrupt mask 0: Not masked 1: Masked

Table 7A.10 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
22	ECCCNFDRAM1 FEIFMSK*1	INTECCCNFDRAM1 interrupt mask 0: Not masked 1: Masked
21	OSTM4 FEIFMSK	INTOSTM4_FE interrupt mask 0: Not masked 1: Masked
20	OSTM3 FEIFMSK	INTOSTM3_FE interrupt mask 0: Not masked 1: Masked
19	OSTM2 FEIFMSK	INTOSTM2_FE interrupt mask 0: Not masked 1: Masked
18	OSTM1 FEIFMSK	INTOSTM1_FE interrupt mask 0: Not masked 1: Masked
17	GUARD FEIFMSK	INTGUARD interrupt mask 0: Not masked 1: Masked
16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	LVIHFEIFMSK	INTLVIH interrupt mask 0: Not masked 1: Masked
14	OSTM0FEI FMSK	INTOSTM0_FE interrupt mask 0: Not masked 1: Masked
13	ECCRAMFEI FMSK	INTECCRAM interrupt mask 0: Not masked 1: Masked
12	ECCFLI0 FEIFMSK	INTECCFLI0 interrupt mask 0: Not masked 1: Masked
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECCCSIH3 FEIFMSK	INTECCCSIH3 interrupt mask 0: Not masked 1: Masked
9	ECCCSIH2 FEIFMSK	INTECCCSIH2 interrupt mask 0: Not masked 1: Masked
8	ECCCSIH1 FEIFMSK	INTECCCSIH1 interrupt mask 0: Not masked 1: Masked
7	ECCCSIH0 FEIFMSK	INTECCCSIH0 interrupt mask 0: Not masked 1: Masked
6	ECCCNFDRAM0 FEIFMSK	INTECCCNFDRAM0 interrupt mask 0: Not masked 1: Masked
5	ECCFLRAM FEIFMSK	INTECCFLRAM interrupt mask 0: Not masked 1: Masked

Table 7A.10 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
4	ECCEEP0 FEIFMSK	INTECCEEP0 interrupt mask 0: Not masked 1: Masked
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVILFEIFMSK	INTLVIL interrupt mask 0: Not masked 1: Masked

Note 1. For the supported products, see **Table 7A.3, FE Level Maskable Interrupt Requests (RH850/F1KH-D8)**.

7A.2.3.4 FEINTFC — FEINT Factor Clear Register

This register clears the bits of the FEINT factor register (FEINTF).

CAUTION

After confirming that handling by both handlers is complete after the completion of processing by a handler for an FEINT interrupt of CPU1 or CPU2, use the FEINT factor clear register (FEINTFC) to clear the flag for the given FEINT interrupt. For example, if processing by CPU1 takes less time than that by CPU2, checking only for the advance completion of processing by CPU1 creates a concern of repeated processing of the interrupt.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0108$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIFC	ECCMMCA0RAMFEIFC	ECCETH1FEIFC	OSTM9FEIFC	OSTM8FEIFC	OSTM7FEIFC	OSTM6FEIFC	OSTM5FEIFC	ECCETH0FEIFC	ECCCNFDRAM1FEIFC	OSTM4FEIFC	OSTM3FEIFC	OSTM2FEIFC	OSTM1FEIFC	GUARDFEIFC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIFC	OSTM0FEIFC	ECCRAMFEIFC	ECCFLI0FEIFC	ECCCSIH4FEIFC	ECCCSIH3FEIFC	ECCCSIH2FEIFC	ECCCSIH1FEIFC	ECCCSIH0FEIFC	ECCCNFDRAMFEIFC	ECCFLRAMFEIFC	ECCEEP0FEIFC	—	—	—	LVILFEIFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	R	R	R	W

Table 7A.11 FEINTFC Register Contents

Bit Position	Bit Name	Function
31	DMAFEIFC	DMAFEIF flag clear 0: — 1: Clear
30	ECCMMCA0RAMFEIFC*1	ECCMMCA0RAMFEIF flag clear 0: — 1: Clear
29	ECCETH1FEIFC*1	ECCETH1FEIF flag clear 0: — 1: Clear
28	OSTM9FEIFC	OSTM9FEIF flag clear 0: — 1: Clear
27	OSTM8FEIFC	OSTM8FEIF flag clear 0: — 1: Clear
26	OSTM7FEIFC	OSTM7FEIF flag clear 0: — 1: Clear
25	OSTM6FEIFC	OSTM6FEIF flag clear 0: — 1: Clear

Table 7A.11 FEINTFC Register Contents

Bit Position	Bit Name	Function
24	OSTM5FEIFC	OSTM5FEIF flag clear 0: — 1: Clear
23	ECCETH0FEIFC	ECCETH0FEIF flag clear 0: — 1: Clear
22	ECCCNFDRA M1FEIFC*[1]	ECCCNFDRAM1FEIF flag clear 0: — 1: Clear
21	OSTM4FEIFC	OSTM4FEIF flag clear 0: — 1: Clear
20	OSTM3FEIFC	OSTM3FEIF flag clear 0: — 1: Clear
19	OSTM2FEIFC	OSTM2FEIF flag clear 0: — 1: Clear
18	OSTM1FEIFC	OSTM1FEIF flag clear 0: — 1: Clear
17	GUARDFEIFC	GUARDFEIF flag clear 0: — 1: Clear
16	Reserved	When writing, write the value after reset.
15	LVIHFEIFC	LVIHFEIF flag clear 0: — 1: Clear
14	OSTM0 FEIFC	OSTM0FEIF flag clear 0: — 1: Clear
13	ECCRAM FEIFC	ECCRAMFEIF flag clear 0: — 1: Clear
12	ECCFLI0FEIFC	ECCFLI0FEIF flag clear 0: — 1: Clear
11	ECCCSIH4 FEIFC	ECCCSIH4FEIF flag clear 0: — 1: Clear
10	ECCCSIH3 FEIFC	ECCCSIH3FEIF flag clear 0: — 1: Clear
9	ECCCSIH2 FEIFC	ECCCSIH2FEIF flag clear 0: — 1: Clear
8	ECCCSIH1 FEIFC	ECCCSIH1FEIF flag clear 0: — 1: Clear

Table 7A.11 FEINTFC Register Contents

Bit Position	Bit Name	Function
7	ECCCSIH0 FEIFC	ECCCSIH0FEIF flag clear 0: — 1: Clear
6	ECCCNFDRAM0 FEIFC	ECCCNFDRAM0FEIF flag clear 0: — 1: Clear
5	ECCFLRAM FEIFC	ECCFLRAMFEIF flag clear 0: — 1: Clear
4	ECCEEP0 FEIFC	ECCEEP0FEIF flag clear 0: — 1: Clear
3 to 1	Reserved	When writing, write the value after reset.
0	LVILFEIFC	LVILFEIF flag clear 0: — 1: Clear

Note 1. For the supported products, see **Table 7A.3, FE Level Maskable Interrupt Requests (RH850/F1KH-D8)**.

7A.3 Edge/Level Detection

External interrupts (TNMI and INTPm) can be specified to be generated when a rising edge, falling edge, rising or falling edge, low level, or high level is detected at an external interrupt pin.

The following registers are used to specify the edge and level of each interrupt:

Table 7A.12 External Interrupt Edge/Level Detection Registers (RH850/F1KH-D8)

Interrupt	Register
TNMI	FCLA0CTL0_NMI
INTP0	FCLA0CTL0_INTPL
INTP1	FCLA0CTL1_INTPL
INTP2	FCLA0CTL2_INTPL
INTP3	FCLA0CTL3_INTPL
INTP4	FCLA0CTL4_INTPL
INTP5	FCLA0CTL5_INTPL
INTP6	FCLA0CTL6_INTPL
INTP7	FCLA0CTL7_INTPL
INTP8	FCLA0CTL0_INTPH
INTP9	FCLA0CTL1_INTPH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12	FCLA0CTL4_INTPH
INTP13	FCLA0CTL5_INTPH
INTP14*[1]	FCLA0CTL6_INTPH
INTP15*[1]	FCLA0CTL7_INTPH
INTP16*[1]	FCLA0CTL0_INTPU
INTP17*[1]	FCLA0CTL1_INTPU
INTP18*[1]	FCLA0CTL2_INTPU
INTP19*[1]	FCLA0CTL3_INTPU
INTP20*[1]	FCLA0CTL4_INTPU
INTP21*[1]	FCLA0CTL5_INTPU
INTP22*[1]	FCLA0CTL6_INTPU
INTP23*[1]	FCLA0CTL7_INTPU

Note 1. For the supported products, see **Table 7A.4, EI Level Maskable Interrupt Sources.**

See **Section 2A, Pin Function of RH850/F1KH-D8** for details of these registers.

7A.4 Interrupt Controller Control Registers

Writing to the ICxxx, IMRm (m = 0 to 11), IBDxxx, FNC, and FIC registers is enabled only in supervisor mode (PSW.UM = 0).

7A.4.1 List of Registers

The following table lists the Interrupt Controller Control registers.

Table 7A.13 List of Registers

Module Name	Register Name	Symbol	Address
INTC1	EI level interrupt control registers	ICxxx*[1]	See **Table 7A.4, EI Level Maskable Interrupt Sources**
	EI level interrupt mask registers	IMR0	FFFE EAF0$_H$
	EI level interrupt binding registers	IBDxxx*[1]	See **Table 7A.17, List of IBDxxx Registers**
INTC2	EI level interrupt control registers	ICxxx*[2]	See **Table 7A.4, EI Level Maskable Interrupt Sources**
	EI level interrupt mask registers	IMRm	FFFF B400$_H$ + (04$_H$ × m)
	EI level interrupt binding registers	IBDxxx*[2]	See **Table 7A.17, List of IBDxxx Registers**
INTC1	FE level NMI status register	FNC	FFFE EA78$_H$
	FE level maskable interrupt status register	FIC	FFFE EA7A$_H$

Note 1. Channel No.0 to 31.

Note 2. Channel No.32 or more.

Among the registers shown in **Table 7A.16, IBDxxx Register Contents**, ICxxx, IMR0, and IBDxxx, which correspond to interrupt channel numbers 0 to 31, are located in INTC1 of the CPU peripheral field included in each CPU. Each register of these can only be accessed from CPU1 or CPU2 which includes it.

Writing is only possible in supervisor mode (PSW.UM = 0).

Among the registers shown in **Table 7A.16, IBDxxx Register Contents**, ICxxx, IMR1 to IMR11, and IBDxxx, which correspond to interrupt channel numbers 32 to 381, are located in INTC2 of peripheral group 0. Writing to these registers is only possible in supervisor mode (UM = 0) by a PE bound to IBDxxx or in supervisor mode (UM = 0) by CPU1.

When writing to IMR1 to IMR11, only the bits corresponding to the conditions described above are overwritten; other bits are not updated.

Among the registers shown in **Table 7A.16, IBDxxx Register Contents**, the values of those listed as reserved for the given channel numbers in **Table 7A.4, EI Level Maskable Interrupt Sources**, must retain their values after a reset.

7A.4.2 ICxxx — EI Level Interrupt Control Registers

One of these registers is assigned to each EI level maskable interrupt (EIINT) channel and is used to set the conditions for controlling that channel. This register is initialized by any reset. For each source, see **Table 7A.4, EI Level Maskable Interrupt Sources**.

CAUTION

If 0 is written to the RFxxx bit immediately after a peripheral module generates the corresponding interrupt request in edge detection mode (before an interrupt is accepted by the CPU), the request may be lost.

Writing 1 to the RFxxx bit immediately after an interrupt is accepted by the CPU may lead to re-issuing of the request.

Writing method to these registers contains the use of bit manipulation instructions (set1, clr1, and not1).

For bit-manipulation instructions, see also **Section 3A.5.3, Access to Registers by Using Bit-Manipulation Instructions**.

Executing a bit-manipulation instruction to the lower bytes including the MKxxx bit has no effect on the RFxxx bit.

Access: ICxxx can be read or written in 16-bit units.

ICxxxH and ICxxxL can be read or written in 8- or 1-bit units.

Access to bits 14, 13, 11 to 8, 5, and 4 by using a SET1, CLR1, or NOT1 instruction is prohibited.

Address: See **Table 7A.4, EI Level Maskable Interrupt Sources**.

Value after reset: $008F_H$ (edge detection), $808F_H$ (level detection)*[1]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTxxx	—	—	RFxxx	—	—	—	—	MKxxx	TBxxx	—	—	P3xxx	P2xxx	P1xxx	P0xxx
Value after reset	0/1*[1]	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Note 1. The value after reset differs depending on the detection type of a given interrupt (edge detection: 0, level detection: 1). For details, see **Table 7A.4, EI Level Maskable Interrupt Sources**.

Table 7A.14 ICxxx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CTxxx</td><td>This bit indicates the type of interrupt detection. This bit is read only.
0: Edge detection
1: Level detection
When writing in 8-bit or 16-bit units, write the value after reset.</td></tr>
<tr><td>14, 13</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>12</td><td>RFxxx</td><td>This is an interrupt request flag.
The RFxxx bit can be written from a program. Setting the RFxxx bit to 1 generates an EI level maskable interrupt n (EIINTn), just as when an interrupt request is acknowledged.
0: No interrupt request is made.
1: Interrupt request is made.
<table>
<tr><th>Input Interface</th><th>Operation</th></tr>
<tr><td>Edge detection (CTxxx = 0)</td><td>This bit is automatically cleared when an interrupt request is acknowledged by the CPU core. It can be set and cleared by software.</td></tr>
<tr><td>Level detection (CTxxx = 1)</td><td>This bit cannot be set or cleared by software. It can only be read.
It is not cleared when an interrupt request is acknowledged by the CPU core.</td></tr>
</table>
</td></tr>
</table>

Table 7A.14 ICxxx Register Contents

Bit Position	Bit Name	Function
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MKxxx	This is the interrupt request mask bit. When the MKxxx bit is set, interrupt requests from the channel are masked and are not issued to the CPU core. The interrupt pending status is not reflected in the ICSR.PMEI bit for any channels that are masked. When the interrupt request from the channel is masked with MKxxx = 1, the RFxxx still reflects the interrupt request for the channel and can be polled in software. When the MKxxx bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the MKxxx bit is also reflected in the corresponding IMRm register. 0: Enables interrupt processing 1: Disables interrupt processing
6	TBxxx	This bit is used to select the way to determine the interrupt vector. 0: Direct jumping to an address determined from the level of priority 1: Table reference For details on the way to determine the interrupt vector, see *the RH850G3KH User's Manual: Software*.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	P3xxx to P0xxx	These bits specify the interrupt priority as one of 16 levels, with 0 as the highest and 15 as the lowest. When multiple EI level-interrupt requests are made simultaneously, the interrupt from the source with the highest priority setting in these bits is selected and conveyed to the CPU core for servicing first. When P3xxx to P0xxx bits specify the same priority level for simultaneously occurring interrupt requests, the source with the lower channel number takes priority.

CAUTIONS

1. Do not access ICxxx registers of interrupt channels listed as “Reserved” in **Table 7A.4, EI Level Maskable Interrupt Sources**, and of the channels which are not incorporated in the product.
2. When a channel n is defined as broadcast interrupt (IBDxxx.CST = 1), MKxxx and RFxxx bits of the ICxxx register of the channel must be set to 0 after an initial configuration of the channel. In the period when the EIINTn interrupt is enabled, it is prohibited to mask (MKxxx = 1) an interrupt processing of the channel. When it is necessary to mask a broadcast interrupt, MKBN0 of ICBN0 registers in INTC1 of each PE can be used to mask the corresponding broadcast interrupt.

7A.4.3 IMRm — EI Level Interrupt Mask Registers (m = 0 to 11)

These registers are a collection of the MKxxx bits of the ICxxx registers. Each bit of IMRm reflects the setting of the corresponding MKxxx bit. The setting for IMRm is also reflected in the corresponding MKxxx bit. This register is initialized by any reset.

Access: IMRm can be read or written in 32-bit units.
IMRmH and IMRmL can be read or written in 16-bit units.
IMRmHH, IMRmHL, IMRmLH, and IMRmLL can be read or written in 8- or 1-bit units.

Address: IMR0: FFFE EAF0$_{H}$
IMR0L: FFFE EAF0$_{H,}$
IMR0H: FFFE EAF2$_{H}$
IMR0LH: FFFE EAF1$_{H,}$
IMR0HL: FFFE EAF2$_{H,}$
IMR0HH: FFFE EAF3$_{H}$
IMRm (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m)
IMRmL (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m)
IMRmH (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m) + 2$_{H,}$
IMRmLL (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m)
IMRmLH (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m) + 1$_{H,}$
IMRmHL (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m) + 2$_{H,}$
IMRmHH (m = 1 to 11): FFFF B400$_{H}$ + (04$_{H}$ × m) + 3$_{H}$

Value after reset: FFFF FFFF$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMRm EIMK (m × 32 + 31)	IMRm EIMK (m × 32 + 30)	IMRm EIMK (m × 32 + 29)	IMRm EIMK (m × 32 + 28)	IMRm EIMK (m × 32 + 27)	IMRm EIMK (m × 32 + 26)	IMRm EIMK (m × 32 + 25)	IMRm EIMK (m × 32 + 24)	IMRm EIMK (m × 32 + 23)	IMRm EIMK (m × 32 + 22)	IMRm EIMK (m × 32 + 21)	IMRm EIMK (m × 32 + 20)	IMRm EIMK (m × 32 + 19)	IMRm EIMK (m × 32 + 18)	IMRm EIMK (m × 32 + 17)	IMRm EIMK (m × 32 + 16)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMRm EIMK (m × 32 + 15)	IMRm EIMK (m × 32 + 14)	IMRm EIMK (m × 32 + 13)	IMRm EIMK (m × 32 + 12)	IMRm EIMK (m × 32 + 11)	IMRm EIMK (m × 32 + 10)	IMRm EIMK (m × 32 + 9)	IMRm EIMK (m × 32 + 8)	IMRm EIMK (m × 32 + 7)	IMRm EIMK (m × 32 + 6)	IMRm EIMK (m × 32 + 5)	IMRm EIMK (m × 32 + 4)	IMRm EIMK (m × 32 + 3)	IMRm EIMK (m × 32 + 2)	IMRm EIMK (m × 32 + 1)	IMRm EIMK (m × 32 + 0)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7A.15 IMRm Register Contents

Bit Position	Bit Name	Function
31 to 0	IMRmEIMK (m × 32 + 31) to IMRmEIMK (m × 32 + 0)	These are interrupt mask bits for EI level maskable interrupt (EIINT) channels 0 to 381. 0: Enables interrupt servicing 1: Disables interrupt servicing

CAUTIONS

1. MKxxx bits which correspond to channels listed as “Reserved” in **Table 7A.4, EI Level Maskable Interrupt Sources** and to channels which are not incorporated in the product must be set to “1”.
2. When a channel is used as broadcast interrupt (IBDxxx.CST = 1), IMRmEIMKn bit corresponding to the channel must be set to 0.

7A.4.4 IBDxxx — EI Level Interrupt Binding Registers

These registers, each of which is for an EI level maskable interrupt (EIINT), are used to set the conditions for control of the interrupts. The registers are initialized by any type of reset.

Access: This register can be read or written in 32-bit units.

Address: See **Table 7A.17 List of IBDxxx Registers.**

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CST	—	—	—	—	—	—	BCP0	—	—	—	—	—	—	GPID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	*1		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Note 1. IBDxxx registers of CPU1 (PE1) (interrupt number 0 to 31): PEID[2:0] = 001B
IBDxxx registers of CPU2 (PE2) (interrupt number 0 to 31): PEID[2:0] = 010B
IBDxxx registers (from interrupt number 32): PEID[2:0]= 001B

Table 7A.16 IBDxxx Register Contents

Bit Position	Bit Name	Function
31	CST	This bit specifies whether interrupt input to the corresponding interrupt channel is or is not conveyed to multiple destinations (broadcast). It can only be set in INTC2. The bit in IBDxxx for interrupt number 0 to 31 (INTC1) has the fixed value 0. 0: The interrupt is handled as a normal interrupt (default value). 1: The interrupt is conveyed to multiple destinations.
30 to 25	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
24	BCP0	These bits specify the number of the broadcast port for output when the CST bit is 1. If the CST bit is 0, the setting of these bits are ignored. The bits in IBDxxx for interrupt number 0 to 31 (INTC1) has the fixed value 0. 0: Output to broadcast port 0 1: Output to broadcast port 1
23 to 18	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
17, 16	GPID[1:0]	These bits are implemented only in the IBDxxx registers (the interrupt number 32 or later). Select one of the followings for setting according to PEID settings. 00: When CPU1 is selected for a binding destination by using PEID 01: When CPU2 is selected for a binding destination by using PEID Other than the above: Setting prohibited These bits are reserved in the IBDxxx registers corresponding to interrupt number 0 to 31. When writing, write 0. These bits are always read as 0.
15 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Table 7A.16 IBDxxx Register Contents

Bit Position	Bit Name	Function
2 to 0	PEID[2:0]	Specify destination for binding (requesting) an interrupt. However, the PEID bit in the IBDxxx registers (interrupt number 0 to 31) multiplexed by each CPU (PE) is fixed to each CPU (PE) number and cannot be changed. 001: An interrupt is bound to CPU1. 010: An interrupt is bound to CPU2. When an interrupt corresponding to a source is performed, select one of the above settings.

CAUTIONS

1. Do not change the value of the corresponding IBDxxx register while the request for EIINT is being processed.
2. Only INTBN0 and INTBN1 support broadcast interrupts in this product. So, total 2 IBDxxx register of CST = 1 can be set.
3. The channel of INTC2 that can be bind to broadcast communication input is only edge detection type.

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
0	IBDIPIR0	FFFE EB00$_H$	✓	✓	—	✓
1	IBDIPIR1	FFFE EB04$_H$	✓	✓	—	✓
2	IBDIPIR2	FFFE EB08$_H$	✓	✓	—	✓
3	IBDIPIR3	FFFE EB0C$_H$	✓	✓	—	✓
4	IBDBN0	FFFE EB10$_H$	✓	✓	—	✓
5	IBDBN1	FFFE EB14$_H$	✓	✓	—	✓
6	—	—	—	—	—	—
7	—	—	—	—	—	—
8	IBDTAUD0I0 IBDCSIH2IC_1	FFFE EB20$_H$	✓	✓	—	✓
9	IBDTAUD0I2 IBDCSIH3IC_1	FFFE EB24$_H$	✓	✓	—	✓
10	IBDTAUD0I4 IBDCSIH2IR_1	FFFE EB28$_H$	✓	✓	—	✓
11	IBDTAUD0I6 IBDCSIH2IRE_1	FFFE EB2C$_H$	✓	✓	—	✓
12	IBDTAUD0I8 IBDCSIH2IJC_1	FFFE EB30$_H$	✓	✓	—	✓
13	IBDTAUD0I10 IBDCSIH3IR_1	FFFE EB34$_H$	✓	✓	—	✓
14	IBDTAUD0I12 IBDCSIH3IRE_1	FFFE EB38$_H$	✓	✓	—	✓
15	IBDTAUD0I14 IBDCSIH3IJC_1	FFFE EB3C$_H$	✓	✓	—	✓
16	IBDTAPA0IPEK0 IBDCSIH1IC_1	FFFE EB40$_H$	✓	✓	—	✓
17	IBDTAPA0IVLY0 IBDCSIH1IR_1	FFFE EB44$_H$	✓	✓	—	✓
18	IBDADCA0I0	FFFE EB48$_H$	✓	✓	—	✓
19	IBDADCA0I1	FFFE EB4C$_H$	✓	✓	—	✓
20	IBDADCA0I2 IBDCSIH0IJC_1	FFFE EB50$_H$	✓	✓	—	✓
21	IBDDCUTDI	FFFE EB54$_H$	✓	✓	—	✓
22	IBDRCANGERR0	FFFE EB58$_H$	✓	✓	—	✓
23	IBDRCANGRECC0	FFFE EB5C$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
24	IBDRCAN0ERR	FFFE EB60$_H$	✓	✓	—	✓
25	IBDRCAN0REC	FFFE EB64$_H$	✓	✓	—	✓
26	IBDRCAN0TRX	FFFE EB68$_H$	✓	✓	—	✓
27	IBDCSIG0IC IBDCSIH1IRE_1	FFFE EB6C$_H$	✓	✓	—	✓
28	IBDCSIG0IR IBDCSIH1IJC_1	FFFE EB70$_H$	✓	✓	—	✓
29	IBDCSIH0IC	FFFE EB74$_H$	✓	✓	—	✓
30	IBDCSIH0IR	FFFE EB78$_H$	✓	✓	—	✓
31	IBDCSIH0IRE	FFFE EB7C$_H$	✓	✓	—	✓
32	IBDCSIH0IJC IBDADCA0I2_2	FFFF B880$_H$	✓	✓	—	✓
33	IBDRLIN30	FFFF B884$_H$	✓	✓	—	✓
34	IBDRLIN30UR0	FFFF B888$_H$	✓	✓	—	✓
35	IBDRLIN30UR1	FFFF B88C$_H$	✓	✓	—	✓
36	IBDRLIN30UR2	FFFF B890$_H$	✓	✓	—	✓
37	IBDP0	FFFF B894$_H$	✓	✓	—	✓
38	IBDP1	FFFF B898$_H$	✓	✓	—	✓
39	IBDP2	FFFF B89C$_H$	✓	✓	—	✓
40	IBDWDTA0	FFFF B8A0$_H$	✓	✓	—	✓
41	IBDWDTA1	FFFF B8A4$_H$	✓	✓	—	✓
42	IBDWDTA2	FFFF B8A8$_H$	✓	✓	—	✓
43	IBDP3	FFFF B8AC$_H$	✓	✓	—	✓
44	IBDP4	FFFF B8B0$_H$	✓	✓	—	✓
45	IBDP5	FFFF B8B4$_H$	✓	✓	—	✓
46	IBDP10	FFFF B8B8$_H$	✓	✓	—	✓
47	IBDP11	FFFF B8BC$_H$	✓	✓	—	✓
48	IBDTAUD0I1	FFFF B8C0$_H$	✓	✓	—	✓
49	IBDTAUD0I3	FFFF B8C4$_H$	✓	✓	—	✓
50	IBDTAUD0I5	FFFF B8C8$_H$	✓	✓	—	✓
51	IBDTAUD0I7	FFFF B8CC$_H$	✓	✓	—	✓
52	IBDTAUD0I9	FFFF B8D0$_H$	✓	✓	—	✓
53	IBDTAUD0I11	FFFF B8D4$_H$	✓	✓	—	✓
54	IBDTAUD0I13	FFFF B8D8$_H$	✓	✓	—	✓
55	IBDTAUD0I15	FFFF B8DC$_H$	✓	✓	—	✓
56	IBDADCA0ERR	FFFF B8E0$_H$	✓	✓	—	✓
57	IBDCSIG0IRE	FFFF B8E4$_H$	✓	✓	—	✓
58	IBDRLIN20	FFFF B8E8$_H$	✓	✓	—	✓
59	IBDRLIN21	FFFF B8EC$_H$	✓	✓	—	✓
60	IBDDMA0	FFFF B8F0$_H$	✓	✓	—	✓
61	IBDDMA1	FFFF B8F4$_H$	✓	✓	—	✓
62	IBDDMA2	FFFF B8F8$_H$	✓	✓	—	✓
63	IBDDMA3	FFFF B8FC$_H$	✓	✓	—	✓
64	IBDDMA4	FFFF B900$_H$	✓	✓	—	✓
65	IBDDMA5	FFFF B904$_H$	✓	✓	—	✓
66	IBDDMA6	FFFF B908$_H$	✓	✓	—	✓
67	IBDDMA7	FFFF B90C$_H$	✓	✓	—	✓
68	IBDDMA8	FFFF B910$_H$	✓	✓	—	✓
69	IBDDMA9	FFFF B914$_H$	✓	✓	—	✓
70	IBDDMA10	FFFF B918$_H$	✓	✓	—	✓
71	IBDDMA11	FFFF B91C$_H$	✓	✓	—	✓
72	IBDDMA12	FFFF B920$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
73	IBDDMA13	FFFF B924$_H$	✓	✓	—	✓
74	IBDDMA14	FFFF B928$_H$	✓	✓	—	✓
75	IBDDMA15	FFFF B92C$_H$	✓	✓	—	✓
76	IBDRIIC0TI	FFFF B930$_H$	✓	✓	—	✓
77	IBDRIIC0EE	FFFF B934$_H$	✓	✓	—	✓
78	IBDRIIC0RI	FFFF B938$_H$	✓	✓	—	✓
79	IBDRIIC0TEI	FFFF B93C$_H$	✓	✓	—	✓
80	IBDTAUJ0I0	FFFF B940$_H$	✓	✓	—	✓
81	IBDTAUJ0I1	FFFF B944$_H$	✓	✓	—	✓
82	IBDTAUJ0I2	FFFF B948$_H$	✓	✓	—	✓
83	IBDTAUJ0I3	FFFF B94C$_H$	✓	✓	—	✓
84	IBDOSTM0	FFFF B950$_H$	✓	✓	—	✓
85	IBDENCA0IOV	FFFF B954$_H$	✓	✓	—	✓
86	IBDENCA0IUD	FFFF B958$_H$	✓	✓	—	✓
87	IBDENCA0I0	FFFF B95C$_H$	✓	✓	—	✓
88	IBDENCA0I1	FFFF B960$_H$	✓	✓	—	✓
89	IBDENCA0IEC	FFFF B964$_H$	✓	✓	—	✓
90	IBDKR0	FFFF B968$_H$	✓	✓	—	✓
91	IBDQFULL	FFFF B96C$_H$	✓	✓	—	✓
92	IBDPWGAG00	FFFF B970$_H$	✓	✓	—	✓
93	IBDPWGAG01	FFFF B974$_H$	✓	✓	—	✓
94	IBDPWGAG02	FFFF B978$_H$	✓	✓	—	✓
95	IBDPWGAG10	FFFF B97C$_H$	✓	✓	—	✓
96	IBDPWGAG11	FFFF B980$_H$	✓	✓	—	✓
97	IBDPWGAG12	FFFF B984$_H$	✓	✓	—	✓
98	IBDCSIG4IC	FFFF B988$_H$	✓	✓	—	✓
99	IBDCSIG4IR	FFFF B98C$_H$	✓	✓	—	✓
100	IBDCSIG4IRE	FFFF B990$_H$	✓	✓	—	✓
101	IBDDMA32	FFFF B994$_H$	✓	✓	—	✓
102	IBDDMA33	FFFF B998$_H$	✓	✓	—	✓
103	IBDDMA34	FFFF B99C$_H$	✓	✓	—	✓
104	—	—	—	—	—	—
105	—	—	—	—	—	—
106	IBDDMA35	FFFF B9A8$_H$	✓	✓	—	✓
107	—	—	—	—	—	—
108	—	—	—	—	—	—
109	—	—	—	—	—	—
110	IBDFLERR	FFFF B9B8$_H$	✓	✓	—	✓
111	IBDFLENDNM	FFFF B9BC$_H$	✓	✓	—	✓
112	IBDCWEND	FFFF B9C0$_H$	✓	✓	—	✓
113	IBDRCAN1ERR	FFFF B9C4$_H$	✓	✓	—	✓
114	IBDRCAN1REC	FFFF B9C8$_H$	✓	✓	—	✓
115	IBDRCAN1TRX	FFFF B9CC$_H$	✓	✓	—	✓
116	IBDCSIH1IC IBDTAPA0IPEK0_2	FFFF B9D0$_H$	✓	✓	—	✓
117	IBDCSIH1IR IBDTAPA0IVLY0_2	FFFF B9D4$_H$	✓	✓	—	✓
118	IBDCSIH1IRE IBDCSIG0IC_2	FFFF B9D8$_H$	✓	✓	—	✓
119	IBDCSIH1IJC IBDCSIG0IR_2	FFFF B9DC$_H$	✓	✓	—	✓
120	IBDRLIN31	FFFF B9E0$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
121	IBDRLIN31UR0	FFFF B9E4$_H$	✓	✓	—	✓
122	IBDRLIN31UR1	FFFF B9E8$_H$	✓	✓	—	✓
123	IBDRLIN31UR2	FFFF B9EC$_H$	✓	✓	—	✓
124	IBDCSIH4IC	FFFF B9F0$_H$	✓	✓	—	✓
125	IBDCSIH4IR	FFFF B9F4$_H$	✓	✓	—	✓
126	IBDCSIH4IRE	FFFF B9F8$_H$	✓	✓	—	✓
127	IBDCSIH4IJC	FFFF B9FC$_H$	✓	✓	—	✓
128	IBDP6	FFFF BA00$_H$	✓	✓	—	✓
129	IBDP7	FFFF BA04$_H$	✓	✓	—	✓
130	IBDP8	FFFF BA08$_H$	✓	✓	—	✓
131	IBDP12	FFFF BA0C$_H$	✓	✓	—	✓
132	IBDCSIH2IC IBDTAUD0I0_2	FFFF BA10$_H$	✓	✓	—	✓
133	IBDCSIH2IR IBDTAUD0I4_2	FFFF BA14$_H$	✓	✓	—	✓
134	IBDCSIH2IRE IBDTAUD0I6_2	FFFF BA18$_H$	✓	✓	—	✓
135	IBDCSIH2IJC IBDTAUD0I8_2	FFFF BA1C$_H$	✓	✓	—	✓
136	—	—	—	—	—	—
137	—	—	—	—	—	—
138	—	—	—	—	—	—
139	—	—	—	—	—	—
140	—	—	—	—	—	—
141	—	—	—	—	—	—
142	IBDTAUB0I0	FFFF BA38$_H$	✓	✓	—	✓
143	IBDTAUB0I1	FFFF BA3C$_H$	✓	✓	—	✓
144	IBDTAUB0I2	FFFF BA40$_H$	✓	✓	—	✓
145	IBDTAUB0I3	FFFF BA44$_H$	✓	✓	—	✓
146	IBDTAUB0I4	FFFF BA48$_H$	✓	✓	—	✓
147	IBDTAUB0I5	FFFF BA4C$_H$	✓	✓	—	✓
148	IBDTAUB0I6	FFFF BA50$_H$	✓	✓	—	✓
149	IBDTAUB0I7	FFFF BA54$_H$	✓	✓	—	✓
150	IBDTAUB0I8	FFFF BA58$_H$	✓	✓	—	✓
151	IBDTAUB0I9	FFFF BA5C$_H$	✓	✓	—	✓
152	IBDTAUB0I10	FFFF BA60$_H$	✓	✓	—	✓
153	IBDTAUB0I11	FFFF BA64$_H$	✓	✓	—	✓
154	IBDTAUB0I12	FFFF BA68$_H$	✓	✓	—	✓
155	IBDTAUB0I13	FFFF BA6C$_H$	✓	✓	—	✓
156	IBDTAUB0I14	FFFF BA70$_H$	✓	✓	—	✓
157	IBDTAUB0I15	FFFF BA74$_H$	✓	✓	—	✓
158	IBDCSIH3IC IBDTAUD0I2_2	FFFF BA78$_H$	✓	✓	—	✓
159	IBDCSIH3IR IBDTAUD0I10_2	FFFF BA7C$_H$	✓	✓	—	✓
160	IBDCSIH3IRE IBDTAUD0I12_2	FFFF BA80$_H$	✓	✓	—	✓
161	IBDCSIH3IJC IBDTAUD0I14_2	FFFF BA84$_H$	✓	✓	—	✓
162	IBDRLIN22	FFFF BA88$_H$	✓	✓	—	✓
163	IBDRLIN23	FFFF BA8C$_H$	✓	✓	—	✓
164	IBDRLIN32	FFFF BA90$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
165	IBDRLIN32UR0	FFFF BA94$_H$	✓	✓	—	✓
166	IBDRLIN32UR1	FFFF BA98$_H$	✓	✓	—	✓
167	IBDRLIN32UR2	FFFF BA9C$_H$	✓	✓	—	✓
168	IBDTAUJ1I0	FFFF BAA0$_H$	✓	✓	—	✓
169	IBDTAUJ1I1	FFFF BAA4$_H$	✓	✓	—	✓
170	IBDTAUJ1I2	FFFF BAA8$_H$	✓	✓	—	✓
171	IBDTAUJ1I3	FFFF BAAC$_H$	✓	✓	—	✓
172	—	—	—	—	—	—
173	IBDFLXA0FDA	FFFF BAB4$_H$	✓	✓	—	✓
174	IBDFLXA0FW	FFFF BAB8$_H$	✓	✓	—	✓
175	IBDFLXA0IQE	FFFF BABC$_H$	✓	✓	—	✓
176	IBDFLXA0IQF	FFFF BAC0$_H$	✓	✓	—	✓
177	IBDFLXA0OT	FFFF BAC4$_H$	✓	✓	—	✓
178	IBDFLXA0OW	FFFF BAC8$_H$	✓	✓	—	✓
179	IBDFLXA0LINE0	FFFF BACC$_H$	✓	✓	—	✓
180	IBDFLXA0LINE1	FFFF BAD0$_H$	✓	✓	—	✓
181	IBDFLXA0TIM0	FFFF BAD4$_H$	✓	✓	—	✓
182	IBDFLXA0TIM1	FFFF BAD8$_H$	✓	✓	—	✓
183	IBDFLXA0TIM2	FFFF BADC$_H$	✓	✓	—	✓
184	IBDRLIN212	FFFF BAE0$_H$	—	—	—	✓
185	IBDRLIN213	FFFF BAE4$_H$	—	—	—	✓
186	IBDRLIN214	FFFF BAE8$_H$	—	—	—	✓
187	IBDRLIN215	FFFF BAEC$_H$	—	—	—	✓
188	IBDDMA36	FFFF BAF0$_H$	✓	✓	—	✓
189	IBDDMA37	FFFF BAF4$_H$	✓	✓	—	✓
190	IBDDMA38	FFFF BAF8$_H$	✓	✓	—	✓
191	IBDDMA39	FFFF BAFC$_H$	✓	✓	—	✓
192	IBDDMA40	FFFF BB00$_H$	✓	✓	—	✓
193	IBDDMA41	FFFF BB04$_H$	✓	✓	—	✓
194	IBDDMA42	FFFF BB08$_H$	✓	✓	—	✓
195	IBDDMA43	FFFF BB0C$_H$	✓	✓	—	✓
196	IBDDMA44	FFFF BB10$_H$	✓	✓	—	✓
197	IBDDMA45	FFFF BB14$_H$	✓	✓	—	✓
198	IBDDMA46	FFFF BB18$_H$	✓	✓	—	✓
199	IBDDMA47	FFFF BB1C$_H$	✓	✓	—	✓
200	IBDDMA48	FFFF BB20$_H$	✓	✓	—	✓
201	IBDDMA49	FFFF BB24$_H$	✓	✓	—	✓
202	IBDDMA50	FFFF BB28$_H$	✓	✓	—	✓
203	IBDDMA51	FFFF BB2C$_H$	✓	✓	—	✓
204	IBDDMA52	FFFF BB30$_H$	✓	✓	—	✓
205	IBDP9	FFFF BB34$_H$	✓	✓	—	✓
206	IBDP13	FFFF BB38$_H$	✓	✓	—	✓
207	IBDP14	FFFF BB3C$_H$	✓	✓	—	✓
208	IBDP15	FFFF BB40$_H$	✓	✓	—	✓
209	IBDRTCA01S	FFFF BB44$_H$	✓	✓	—	✓
210	IBDRTCA0AL	FFFF BB48$_H$	✓	✓	—	✓
211	IBDRTCA0R	FFFF BB4C$_H$	✓	✓	—	✓
212	IBDADCA1ERR	FFFF BB50$_H$	✓	✓	—	✓
213	IBDADCA1I0	FFFF BB54$_H$	✓	✓	—	✓
214	IBDADCA1I1	FFFF BB58$_H$	✓	✓	—	✓
215	IBDADCA1I2	FFFF BB5C$_H$	✓	✓	—	✓
216	IBDDMA53	FFFF BB60$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
217	IBDRCAN2ERR	FFFF BB64$_H$	✓	✓	—	✓
218	IBDRCAN2REC	FFFF BB68$_H$	✓	✓	—	✓
219	IBDRCAN2TRX	FFFF BB6C$_H$	✓	✓	—	✓
220	IBDRCAN3ERR	FFFF BB70$_H$	✓	✓	—	✓
221	IBDRCAN3REC	FFFF BB74$_H$	✓	✓	—	✓
222	IBDRCAN3TRX	FFFF BB78$_H$	✓	✓	—	✓
223	IBDCSIG1IC	FFFF BB7C$_H$	✓	✓	—	✓
224	IBDCSIG1IR	FFFF BB80$_H$	✓	✓	—	✓
225	IBDCSIG1IRE	FFFF BB84$_H$	✓	✓	—	✓
226	IBDRLIN24	FFFF BB88$_H$	✓	✓	—	✓
227	IBDRLIN25	FFFF BB8C$_H$	✓	✓	—	✓
228	IBDRLIN33	FFFF BB90$_H$	✓	✓	—	✓
229	IBDRLIN33UR0	FFFF BB94$_H$	✓	✓	—	✓
230	IBDRLIN33UR1	FFFF BB98$_H$	✓	✓	—	✓
231	IBDRLIN33UR2	FFFF BB9C$_H$	✓	✓	—	✓
232	IBDRLIN34	FFFF BBA0$_H$	✓	✓	—	✓
233	IBDRLIN34UR0	FFFF BBA4$_H$	✓	✓	—	✓
234	IBDRLIN34UR1	FFFF BBA8$_H$	✓	✓	—	✓
235	IBDRLIN34UR2	FFFF BBAC$_H$	✓	✓	—	✓
236	IBDRLIN35	FFFF BBB0$_H$	✓	✓	—	✓
237	IBDRLIN35UR0	FFFF BBB4$_H$	✓	✓	—	✓
238	IBDRLIN35UR1	FFFF BBB8$_H$	✓	✓	—	✓
239	IBDRLIN35UR2	FFFF BBBC$_H$	✓	✓	—	✓
240	IBDRIIC1TI	FFFF BBC0$_H$	✓	✓	—	✓
241	IBDRIIC1EE	FFFF BBC4$_H$	✓	✓	—	✓
242	IBDRIIC1RI	FFFF BBC8$_H$	✓	✓	—	✓
243	IBDRIIC1TEI	FFFF BBCC$_H$	✓	✓	—	✓
244	IBDRCAN8ERR	FFFF BBD0$_H$	—	—	—	✓
245	IBDRCAN8REC	FFFF BBD4$_H$	—	—	—	✓
246	IBDRCAN8TRX	FFFF BBD8$_H$	—	—	—	✓
247	IBDRCAN9ERR	FFFF BBDC$_H$	—	—	—	✓
248	IBDRCAN9REC	FFFF BBE0$_H$	—	—	—	✓
249	IBDRCAN9TRX	FFFF BBE4$_H$	—	—	—	✓
250	IBDRCAN10ERR	FFFF BBE8$_H$	—	—	—	✓
251	IBDRCAN10REC	FFFF BBEC$_H$	—	—	—	✓
252	IBDRCAN10TRX	FFFF BBF0$_H$	—	—	—	✓
253	IBDRCAN11ERR	FFFF BBF4$_H$	—	—	—	✓
254	IBDRCAN11REC	FFFF BBF8$_H$	—	—	—	✓
255	IBDRCAN11TRX	FFFF BBFC$_H$	—	—	—	✓
256	IBDTAUB1I0	FFFF BC00$_H$	✓	✓	—	✓
257	IBDTAUB1I1	FFFF BC04$_H$	✓	✓	—	✓
258	IBDTAUB1I2	FFFF BC08$_H$	✓	✓	—	✓
259	IBDTAUB1I3	FFFF BC0C$_H$	✓	✓	—	✓
260	IBDTAUB1I4	FFFF BC10$_H$	✓	✓	—	✓
261	IBDTAUB1I5	FFFF BC14$_H$	✓	✓	—	✓
262	IBDTAUB1I6	FFFF BC18$_H$	✓	✓	—	✓
263	IBDTAUB1I7	FFFF BC1C$_H$	✓	✓	—	✓
264	IBDTAUB1I8	FFFF BC20$_H$	✓	✓	—	✓
265	IBDTAUB1I9	FFFF BC24$_H$	✓	✓	—	✓
266	IBDTAUB1I10	FFFF BC28$_H$	✓	✓	—	✓
267	IBDTAUB1I11	FFFF BC2C$_H$	✓	✓	—	✓
268	IBDTAUB1I12	FFFF BC30$_H$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
269	IBDTAUB1I13	FFFF BC34$_H$	✓	✓	—	✓
270	IBDTAUB1I14	FFFF BC38$_H$	✓	✓	—	✓
271	IBDTAUB1I15	FFFF BC3C$_H$	✓	✓	—	✓
272	IBDRCAN4ERR	FFFF BC40$_H$	✓	✓	—	✓
273	IBDRCAN4REC	FFFF BC44$_H$	✓	✓	—	✓
274	IBDRCAN4TRX	FFFF BC48$_H$	✓	✓	—	✓
275	IBDRLIN26	FFFF BC4C$_H$	✓	✓	—	✓
276	IBDRLIN27	FFFF BC50$_H$	✓	✓	—	✓
277	IBDTAUJ2I0	FFFF BC54$_H$	✓	✓	—	✓
278	IBDTAUJ2I1	FFFF BC58$_H$	✓	✓	—	✓
279	IBDTAUJ2I2	FFFF BC5C$_H$	✓	✓	—	✓
280	IBDTAUJ2I3	FFFF BC60$_H$	✓	✓	—	✓
281	IBDTAUJ3I0	FFFF BC64$_H$	✓	✓	—	✓
282	IBDTAUJ3I1	FFFF BC68$_H$	✓	✓	—	✓
283	IBDTAUJ3I2	FFFF BC6C$_H$	✓	✓	—	✓
284	IBDTAUJ3I3	FFFF BC70$_H$	✓	✓	—	✓
285	IBDRLIN28	FFFF BC74$_H$	✓	✓	—	✓
286	IBDRLIN29	FFFF BC78$_H$	✓	✓	—	✓
287	IBDRCAN5ERR	FFFF BC7C$_H$	✓	✓	—	✓
288	IBDRCAN5REC	FFFF BC80$_H$	✓	✓	—	✓
289	IBDRCAN5TRX	FFFF BC84$_H$	✓	✓	—	✓
290	IBDDMA54	FFFF BC88$_H$	✓	✓	—	✓
291	IBDDMA55	FFFF BC8C$_H$	✓	✓	—	✓
292	IBDDMA56	FFFF BC90$_H$	✓	✓	—	✓
293	IBDDMA57	FFFF BC94$_H$	✓	✓	—	✓
294	IBDDMA58	FFFF BC98$_H$	✓	✓	—	✓
295	IBDDMA59	FFFF BC9C$_H$	✓	✓	—	✓
296	IBDDMA60	FFFF BCA0$_H$	✓	✓	—	✓
297	IBDDMA61	FFFF BCA4$_H$	✓	✓	—	✓
298	IBDDMA16	FFFF BCA8$_H$	✓	✓	—	✓
299	IBDDMA17	FFFF BCAC$_H$	✓	✓	—	✓
300	IBDDMA18	FFFF BCB0$_H$	✓	✓	—	✓
301	IBDDMA19	FFFF BCB4$_H$	✓	✓	—	✓
302	IBDDMA20	FFFF BCB8$_H$	✓	✓	—	✓
303	IBDDMA21	FFFF BCBC$_H$	✓	✓	—	✓
304	IBDDMA22	FFFF BCC0$_H$	✓	✓	—	✓
305	IBDDMA23	FFFF BCC4$_H$	✓	✓	—	✓
306	IBDDMA24	FFFF BCC8$_H$	✓	✓	—	✓
307	IBDDMA25	FFFF BCCC$_H$	✓	✓	—	✓
308	IBDDMA26	FFFF BCD0$_H$	✓	✓	—	✓
309	IBDDMA27	FFFF BCD4$_H$	✓	✓	—	✓
310	IBDDMA28	FFFF BCD8$_H$	✓	✓	—	✓
311	IBDDMA29	FFFF BCDC$_H$	✓	✓	—	✓
312	IBDDMA30	FFFF BCE0$_H$	✓	✓	—	✓
313	IBDDMA31	FFFF BCE4$_H$	✓	✓	—	✓
314	IBDOSTM5	FFFF BCE8$_H$	✓	✓	—	✓
315	IBDETNB0DATA	FFFF BCEC$_H$	✓	✓	—	✓
316	IBDETNB0ERR	FFFF BCF0$_H$	✓	✓	—	✓
317	IBDETNB0MNG	FFFF BCF4$_H$	✓	✓	—	✓
318	IBDETNB0MAC	FFFF BCF8$_H$	✓	✓	—	✓
319	IBDRCANGERR1	FFFF BCFC$_H$	—	—	—	✓
320	IBDRCANGRECC1	FFFF BD00$_H$	—	—	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
321	IBDRCAN6ERR	FFFF BD04$_{H}$	✓	✓	—	✓
322	IBDRCAN6REC	FFFF BD08$_{H}$	✓	✓	—	✓
323	IBDRCAN6TRX	FFFF BD0C$_{H}$	✓	✓	—	✓
324	IBDRLIN210	FFFF BD10$_{H}$	—	✓	—	✓
325	IBDRLIN211	FFFF BD14$_{H}$	—	✓	—	✓
326	IBDCSIG2IC	FFFF BD18$_{H}$	✓	✓	—	✓
327	IBDCSIG2IR	FFFF BD1C$_{H}$	✓	✓	—	✓
328	IBDCSIG2IRE	FFFF BD20$_{H}$	✓	✓	—	✓
329	IBDCSIG3IC	FFFF BD24$_{H}$	✓	✓	—	✓
330	IBDCSIG3IR	FFFF BD28$_{H}$	✓	✓	—	✓
331	IBDCSIG3IRE	FFFF BD2C$_{H}$	✓	✓	—	✓
332	IBDRCAN7ERR	FFFF BD30$_{H}$	✓	✓	—	✓
333	IBDRCAN7REC	FFFF BD34$_{H}$	✓	✓	—	✓
334	IBDRCAN7TRX	FFFF BD38$_{H}$	✓	✓	—	✓
335	—	—	—	—	—	—
336	—	—	—	—	—	—
337	—	—	—	—	—	—
338	—	—	—	—	—	—
339	—	—	—	—	—	—
340	—	—	—	—	—	—
341	—	—	—	—	—	—
342	—	—	—	—	—	—
343	—	—	—	—	—	—
344	—	—	—	—	—	—
345	—	—	—	—	—	—
346	—	—	—	—	—	—
347	IBDSENT0SI	FFFF BD6C$_{H}$	✓	✓	—	✓
348	IBDSENT0RI	FFFF BD70$_{H}$	✓	✓	—	✓
349	IBDSENT1SI	FFFF BD74$_{H}$	✓	✓	—	✓
350	IBDSENT1RI	FFFF BD78$_{H}$	✓	✓	—	✓
351	—	—	—	—	—	—
352	—	—	—	—	—	—
353	—	—	—	—	—	—
354	IBDMMCA0	FFFF BD88$_{H}$	—	—	—	✓
355	IBDDMA62	FFFF BD8C$_{H}$	✓	✓	—	✓
356	IBDDPE	FFFF BD90$_{H}$	✓	✓	—	✓
357	IBDAPE	FFFF BD94$_{H}$	✓	✓	—	✓
358	—	—	—	—	—	—
359	—	—	—	—	—	—
360	IBDRLIN36	FFFF BDA0$_{H}$	✓	✓	—	✓
361	IBDRLIN36UR0	FFFF BDA4$_{H}$	✓	✓	—	✓
362	IBDRLIN36UR1	FFFF BDA8$_{H}$	✓	✓	—	✓
363	IBDRLIN36UR2	FFFF BDAC$_{H}$	✓	✓	—	✓
364	IBDRLIN37	FFFF BDB0$_{H}$	✓	✓	—	✓
365	IBDRLIN37UR0	FFFF BDB4$_{H}$	✓	✓	—	✓
366	IBDRLIN37UR1	FFFF BDB8$_{H}$	✓	✓	—	✓
367	IBDRLIN37UR2	FFFF BDBC$_{H}$	✓	✓	—	✓
368	IBDP16	FFFF BDC0$_{H}$	✓	✓	—	✓
369	IBDP17	FFFF BDC4$_{H}$	✓	✓	—	✓
370	IBDP18	FFFF BDC8$_{H}$	✓	✓	—	✓
371	IBDP19	FFFF BDCC$_{H}$	✓	✓	—	✓
372	IBDP20	FFFF BDD0$_{H}$	✓	✓	—	✓

Table 7A.17 List of IBDxxx Registers

Interrupt Number	Register Name	Address	176 Pins	233 Pins	272 Pins	324 Pins
373	IBDP21	FFFF BDD4$_H$	✓	✓	—	✓
374	IBDP22	FFFF BDD8$_H$	✓	✓	—	✓
375	IBDP23	FFFF BDDC$_H$	✓	✓	—	✓
376	IBDGRZF	FFFF BDE0$_H$	✓	✓	—	✓
377	IBDETNB1DATA	FFFF BDE4$_H$	—	—	—	✓
378	IBDETNB1ERR	FFFF BDE8$_H$	—	—	—	✓
379	IBDETNB1MNG	FFFF BDEC$_H$	—	—	—	✓
380	IBDETNB1MAC	FFFF BDF0$_H$	—	—	—	✓
381	IBDDMA63	FFFF BDF4$_H$	✓	✓	—	✓

7A.4.5 FNC — FE Level NMI Status Register

This register indicates the status of an FE level non-maskable interrupt (FENMI).

This register is initialized by any reset.

Access:	FNC is a read-only register that can be read in 16-bit units. FNCH is a read-only register that can be read in 8- or 1-bit units.
Address:	FNC: FFFE EA78$_H$ FNCH: FFFE EA79$_H$
Value after reset:	0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FNRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7A.18 FNC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FNRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit is automatically cleared when an FE level NMI interrupt request is acknowledged by the CPU core.
11 to 0	Reserved	When read, the value after reset is returned.

7A.4.6 FIC — FE Level Maskable Interrupt Status Register

This register indicates the status of an FE level maskable interrupt (FEINT).

This register is initialized by any reset.

Access: FIC is a read-only register that can be read in 16-bit units.

FICH is a read-only register that can be read in 8- or 1-bit units.

Address: FIC: FFFE EA7A$_H$

FICH: FFFE EA7B$_H$

Value after reset: 8000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FIRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7A.19 FIC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FIRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit cannot be set or cleared by software. It can only be read.
11 to 0	Reserved	When read, the value after reset is returned.

7A.5 EI Level Maskable Interrupt Select Register

The following registers are used to select an EI level maskable interrupt.

7A.5.1 List of Registers

The following table lists the EI Level Maskable Interrupt Select register.

Table 7A.20 List of Register

Module Name	Register Name	Symbol	Address
SL_INTC	INTC1 interrupt select register	SELB_INTC1	FFC0 1000_H

7A.5.2 SELB_INTC1 — INTC1 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled. This register is initialized by any reset.

NOTE

The channel described in each bit setting indicates the channel of an interrupt and the priority. For details on channels, see **Table 7A.4, EI Level Maskable Interrupt Sources**.

Access: This register can be read or written in 16-bit units.

Address: FFC0 1000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SELB_INTC1_12	SELB_INTC1_11	SELB_INTC1_10	SELB_INTC1_9	SELB_INTC1_8	SELB_INTC1_7	SELB_INTC1_6	SELB_INTC1_5	SELB_INTC1_4	SELB_INTC1_3	SELB_INTC1_2	SELB_INTC1_1	SELB_INTC1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7A.21 SELB_INTC1 Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SELB_INTC1_12	Interrupt channel selection 0: INTADCA0I2 (Channel 20) INTCSIH0IJC (Channel 32) 1: INTCSIH0IJC_1 (Channel 20) INTADCA0I2_2 (Channel 32)
11	SELB_INTC1_11	Interrupt channel selection 0: INTTAUD0I14 (Channel 15) INTCSIH3IJC (Channel 161) 1: INTCSIH3IJC_1 (Channel 15) INTTAUD0I14_2 (Channel 161)
10	SELB_INTC1_10	Interrupt channel selection 0: INTTAUD0I12 (Channel 14) INTCSIH3IRE (Channel 160) 1: INTCSIH3IRE_1 (Channel 14) INTTAUD0I12_2 (Channel 160)
9	SELB_INTC1_9	Interrupt channel selection 0: INTTAUD0I10 (Channel 13) INTCSIH3IR (Channel 159) 1: INTCSIH3IR_1 (Channel 13) INTTAUD0I10_2 (Channel 159)
8	SELB_INTC1_8	Interrupt channel selection 0: INTTAUD0I2 (Channel 9) INTCSIH3IC (Channel 158) 1: INTCSIH3IC_1 (Channel 9) INTTAUD0I2_2 (Channel 158)

Table 7A.21 SELB_INTC1 Register Contents

Bit Position	Bit Name	Function
7	SELB_INTC1_7	Interrupt channel selection 0: INTTAUD0I8 (Channel 12) INTCSIH2IJC (Channel 135) 1: INTCSIH2IJC_1 (Channel 12) INTTAUD0I8_2 (Channel 135)
6	SELB_INTC1_6	Interrupt channel selection 0: INTTAUD0I6 (Channel 11) INTCSIH2IRE (Channel 134) 1: INTCSIH2IRE_1 (Channel 11) INTTAUD0I6_2 (Channel 134)
5	SELB_INTC1_5	Interrupt channel selection 0: INTTAUD0I4 (Channel 10) INTCSIH2IR (Channel 133) 1: INTCSIH2IR_1 (Channel 10) INTTAUD0I4_2 (Channel 133)
4	SELB_INTC1_4	Interrupt channel selection 0: INTTAUD0I0 (Channel 8) INTCSIH2IC (Channel 132) 1: INTCSIH2IC_1 (Channel 8) INTTAUD0I0_2 (Channel 132)
3	SELB_INTC1_3	Interrupt channel selection 0: INTCSIG0IR (Channel 28) INTCSIH1IJC (Channel 119) 1: INTCSIH1IJC_1 (Channel 28) INTCSIG0IR_2 (Channel 119)
2	SELB_INTC1_2	Interrupt channel selection 0: INTCSIG0IC (Channel 27) INTCSIH1IRE (Channel 118) 1: INTCSIH1IRE_1 (Channel 27) INTCSIG0IC_2 (Channel 118)
1	SELB_INTC1_1	Interrupt channel selection 0: INTTAPA0IVLY0 (Channel 17) INTCSIH1IR (Channel 117) 1: INTCSIH1IR_1 (Channel 17) INTTAPA0IVLY0_2 (Channel 117)
0	SELB_INTC1_0	Interrupt channel selection 0: INTTAPA0IPEK0 (Channel 16) INTCSIH1IC (Channel 116) 1: INTCSIH1IC_1 (Channel 16) INTTAPA0IPEK0_2 (Channel 116)

NOTE

For the supported products, see **Table 7A.4, EI Level Maskable Interrupt Sources**.

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB_INTC1.

7A.6 Interrupt Function System Registers

See **Table 3A.31, Interrupt Function System Registers**.

7A.6.1 FPIPR — FPI Exception Interrupt Priority

See **Table 3A.32, FPIPR Register Contents**.

7A.6.2 ISPR — Priority of Interrupt being Serviced

See **Table 3A.33, ISPR Register Contents**.

7A.6.3 PMR — Interrupt Priority Masking

See **Table 3A.34, PMR Register Contents**.

7A.6.4 ICSR — Interrupt Control Status

See **Table 3A.35, ICSR Register Contents**.

7A.6.5 INTCFG — Interrupt Function Setting

See **Table 3A.36, INTCFG Register Contents**.

7A.7 Operation when Acknowledging an Interrupt

Check whether each interrupt that is reported during instruction execution is acknowledged according to the priority. The procedure for acknowledging each interrupt is shown below.

(1) Check whether the acknowledgment conditions are satisfied and whether interrupts are acknowledged according to their priority.

(2) Calculate the exception handler address according to the current PSW value.*[1]

(3) For FE-level non-maskable/maskable interrupts, the following processing is performed:

- Save the PC to the FEPC.
- Save the PSW to the FEPSW.
- Store the exception source code in the FEIC.
- Update the PSW.*[2]
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

(4) For EI level exceptions, the following processing is performed:

- Save the PC to the EIPC.
- Save the PSW to the EIPSW.
- Store the exception source code in the EIIC.
- Update the PSW.*[2]
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

Note 1. For details, see **Section 7A.10, Exception Handler Address**.

Note 2. For the values to be updated, see *Table 4.1 Exception Cause List* in *the RH850G3KH User's Manual: Software*.

The following figure shows steps (1) to (4).

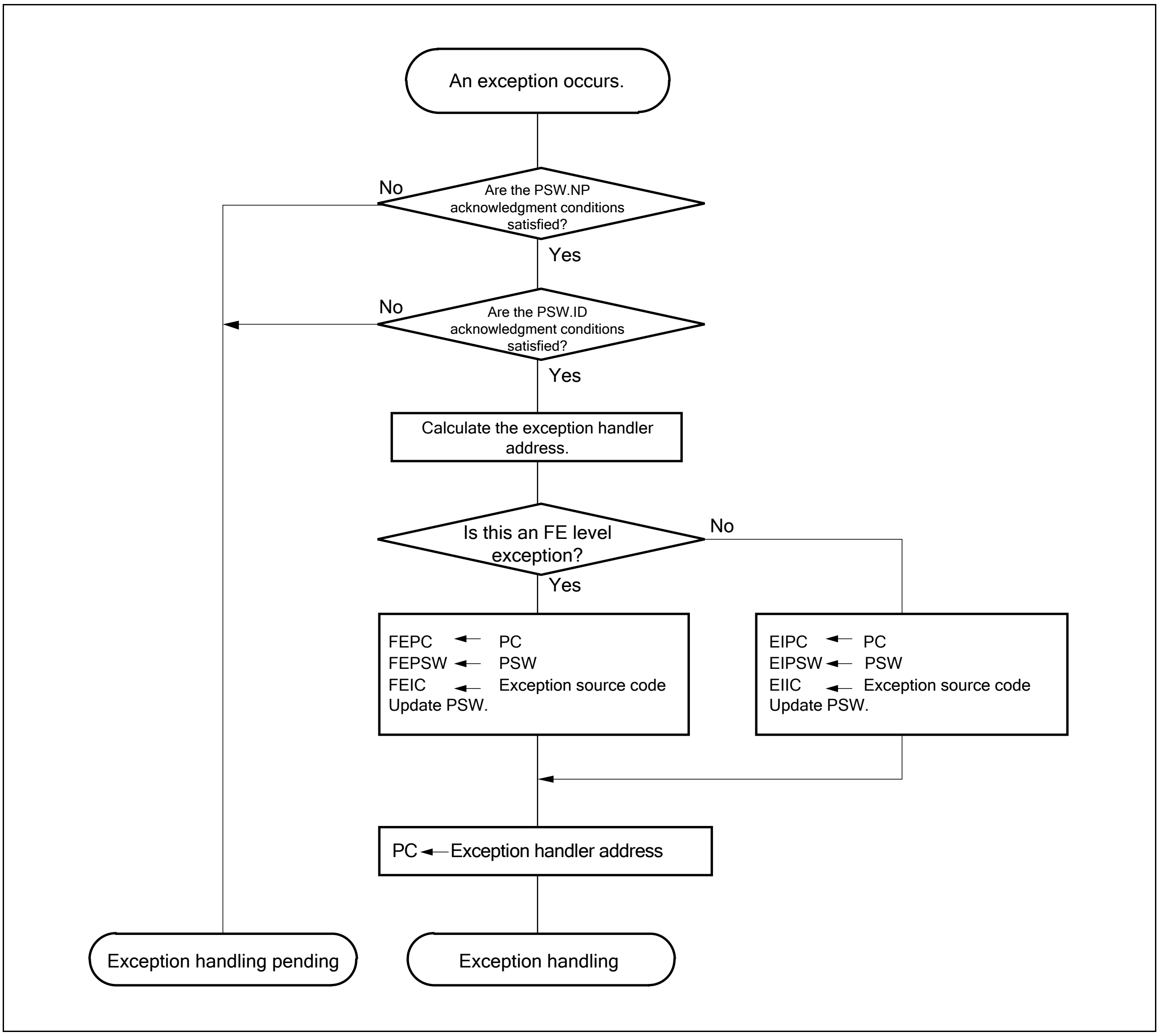


Figure 7A.2 Operation when Acknowledging an Interrupt

7A.7.1 Exception Source Codes for Different Types of SYSERR Exceptions

The following table lists exception source codes for the different types of SYSERR exceptions.

Table 7A.22 Exception Source Codes for Different Types of SYSERR Exceptions

Exception Source Code	Source of SYSERR Generation
11_H	● Detection of an error during the fetching of an instruction from the code flash memory area
13_H	● Detection of an error during the fetching of an instruction from the local, global or retention RAM areas
14_H	● Detection of an error during access to data in the code flash area ● Detection of an error during read access to a module via the system interconnect or PBUS The exception source code reports an occurrence of a system error factor corresponding to VCIF bit of SEGFLAG register*[1]
16_H	● Detection of an error during access to data in the own local RAM areas The exception source code reports an occurrence of a system error factor corresponding to TCMF bit of SEGFLAG register*[1]
18_H	● Detection of an IPG error The exception source code reports an occurrence of a system error factor corresponding to VCIF, IPGF bit of SEGFLAG register*[1]
19_H	● Detection of an error during write access to a module via the PBUS The exception source code reports an occurrence of a system error factor corresponding to APIF bit of SEGFLAG register*[1]
$1A_H$	● Detection of an error inside system interconnect The exception source code reports an occurrence of a system error factor corresponding to VCSF bit of SEGFLAG register*[1]

Note 1. See **Section 3A.2.4.3, System Error Generator Function (SEG)** for details.

7A.8 Return from Interrupts

To return from interrupt handling, execute the return instruction (EIRET or FERET) corresponding to each relevant interrupt level.

When a context has been saved in a stack and the like, the context must be restored before executing the return instruction.

The EIRET instruction is used to return from the EI level maskable interrupt handling and the FERET instruction is used to return from FE-level maskable interrupt handling.

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes its control to the return PC address:

(1) When returning from the service routine for an EI-level exception, the PC and PSW values on return are loaded from the EIPC and EIPSW registers.
When returning from the service routine for an FE-level exception, the PC and PSW values on return are loaded from the FEPC and FEPSW registers.

(2) Control is passed to the addresses indicated by the return PC and PSW that were loaded.

(3) When EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.

The flows for returning from exception handling using the EIRET and FERET instructions are shown below.

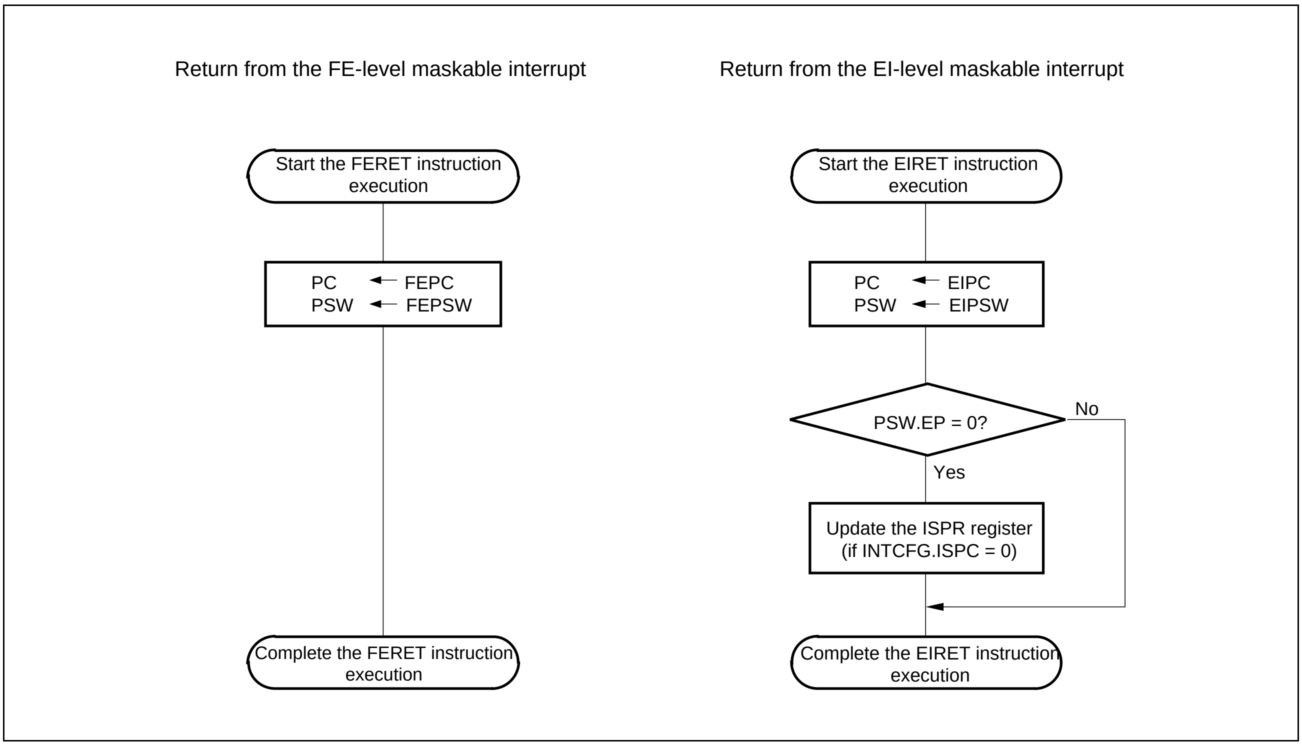


Figure 7A.3 Flow of Return from Interrupts

7A.9 Interrupt Operation

7A.9.1 Interrupt Mask Function of EI Level Maskable Interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by the following register settings.

Table 7A.23 Operation of the MKxxx Bit

ICxxx.MKxxx	Operation
1	Masks interrupt
0	Enables interrupt

The ICxxx.MKxxx bits can also be read and written via the corresponding IMRmEIMKn bits of the IMRm registers. The interrupt mask state is reflected in both the ICxxx registers and the IMRm registers.

[Operation example]

(1) When a 1 is written to an IMRm.IMRmEIMKn bit, interrupts are prohibited for the corresponding channel.

(2) When the corresponding ICxxx.MKxxx bit is read, 1 is returned.

CAUTION

If the MKxxx bit is set to 0 while an interrupt request is pending (RFxxx = 1), the interrupt service routine will be executed at that time (subject to the rules of interrupt prioritization). Even if an interrupt request is issued in software by setting the RFxxx bit to 1, the interrupt will not occur as long as the interrupt is masked with MKxxx = 1.
To cancel an interrupt request that is pending, clear the corresponding RFxxx bit in software.

7A.9.2 Interrupt Priority Level Judgment

When FE level non-maskable interrupts (FENMI), FE level maskable interrupts (FEINT), and EI level maskable interrupts (EIINT) are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is processed. Exceptions occurred at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

FENMI > FEINT > EIINT

See *the RH850G3KH User's Manual: Software* for other exceptions.

For EIINT(INTn) interrupts, the interrupt priority can be set independently for each interrupt source. Specify the interrupt priority with the bits P3xxx to P0xxx. The interrupt priority levels can be set from 0 to 15: 0 is the highest and 15 is the lowest. Among multiple EIINT(INTn) interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

Table 7A.24 Example of EIINT (INTn) Interrupt Priority Level Settings and Priority Levels during Operation

EIINT (INTn)	ICxxx.P[3:0]xxx Setting	Priority Level During Operation
INT0	3	10
INT1	4	11
INT2	0	1
INT3	0	2
INT4	1	3
INT5	2	6
INT6	2	7
INT7	1	4
INT8	1	5
INT9	2	8
INT10	2	9

The interrupt controller executes multiple interrupt handling when another interrupt request is acknowledged while an interrupt processing is being executed. When multiple EIINT (INTn) interrupts occur at the same time, the interrupt to be acknowledged is determined by the following procedure.

7A.9.2.1 Comparison with the Priority Level of the Interrupt Currently being Handled

Interrupts with the same or lower priority level as the interrupt currently being handled are held pending.

The priority level of the interrupt currently being handled is stored in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being handled proceed to the next priority judgment stage.

7A.9.2.2 Masking through Priority Mask Register (PMR)

Only interrupts enabled by the PMR register proceed to the next priority judgment stage.

For the PMR register, see **Table 3A.34, PMR Register Contents**, or *the RH850G3KH User's Manual: Software*.

7A.9.2.3 The Requested Interrupt Source with the Highest Priority Level is Selected

When interrupts are requested simultaneously from multiple sources, the interrupt set the highest priority by ICxxx.P[3:0]xxx bits takes priority.

When there are multiple highest priority interrupts, the lowest interrupt channel number is selected.

7A.9.2.4 Interrupt Suspended by CPU

Interrupt acknowledgment is held according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT, FEINT and FENMI interrupts are performed even while interrupt acknowledgment is pending, and the interrupt with the highest priority is selected when the acknowledgment condition is satisfied.

Example

An EIINT interrupt with the priority level 5 has already been requested and interrupt generation is pending because the value of the PSW.ID bit is 1. If a subsequent EIINT interrupt with the priority level 3 is requested and the PSW.ID bit is cleared to 0, the latter EIINT interrupt (with the priority level 3) will be generated.

Figure 7A.4, Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1) shows an example of multiple interrupt handling when another interrupt request is acknowledged while interrupt processing is being executed.

When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, the ID flag should be cleared to 0 to execute multiple interrupt handling. Specifically, execute the EI instruction and the like in an interrupt handling program to enable the interrupt.

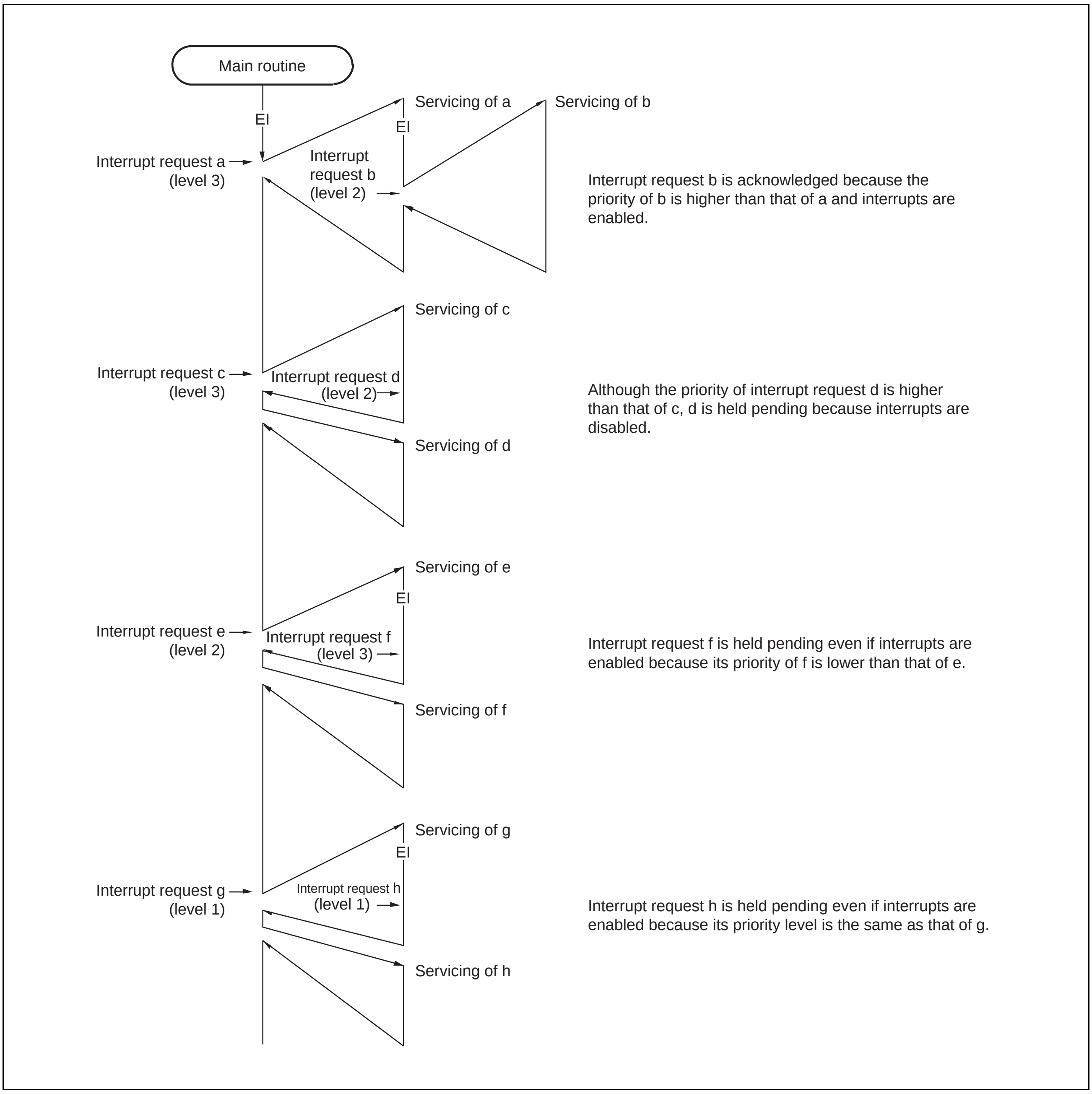
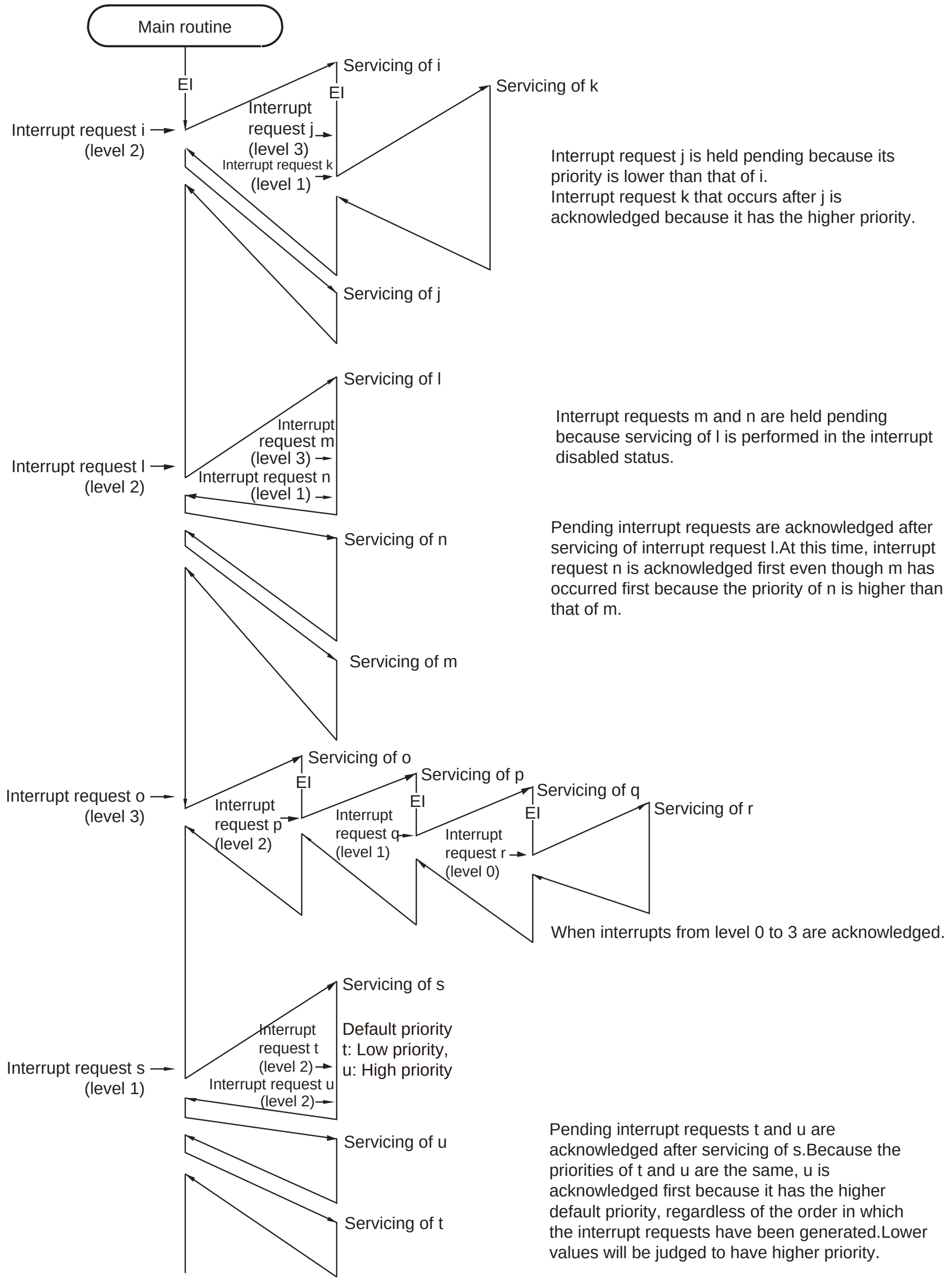


Figure 7A.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1)



Interrupt request j is held pending because its priority is lower than that of i. Interrupt request k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

When interrupts from level 0 to 3 are acknowledged.

Pending interrupt requests t and u are acknowledged after servicing of s.Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.Lower values will be judged to have higher priority.

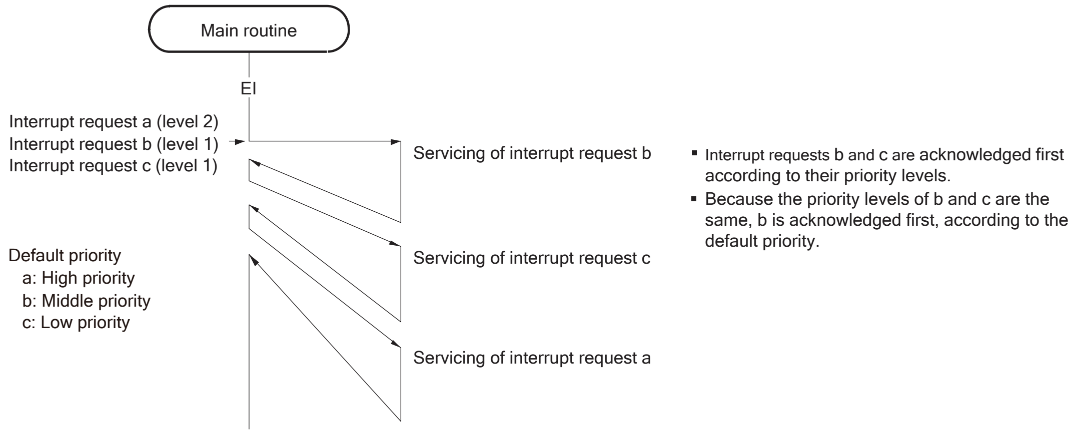
Note 1. “a” to “u” in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7A.5 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (2)

CAUTION

To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.



Note 1. “a” to “c” in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7A.6 Example of Servicing Simultaneously Generated Interrupt Requests

7A.9.3 Interrupt Request Acknowledgement Conditions and the Priority

See *the RH850G3KH User's Manual: Software*.

7A.9.4 Exception Priority of Interrupts and the Priority Mask

See *the RH850G3KH User's Manual: Software*.

7A.9.5 Interrupt Priority Mask

See *the RH850G3KH User's Manual: Software*.

7A.9.6 Priority Mask Function

The priority mask function prohibits all EIINT interrupts of the specified interrupt priority level.

The interrupt priority levels to be masked are specified in the PMR register. Masking and acknowledgment can be set for each priority level.

The following operations are possible using this function:

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

Table 7A.25 Operation of the PMR.PMm Bit

PMR.PMm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note: m = 0 to 15

The presence of EIINT interrupts held pending with this function can be checked with **Section 7A.9.7, Exception Management**.

For details on the PMR register, see **Table 3A.34, PMR Register Contents**, or *the RH850G3KH User's Manual: Software*.

7A.9.7 Exception Management

Pending interrupts can be checked in the RH850/F1KH. For details, see *the RH850G3KH User's Manual: Software*.

7A.9.8 Inter-Processor Interrupts

Four registers (IPIR_CHn) for conveying interrupts between PEs are provided.

IPIR_CH0 to IPIR_CH3 are assigned to CH0 to CH3 of user interrupt (EIINT). An interrupt for specific PEs (including own PE) can be requested by manipulating bits corresponding to respective PEs.

Interrupt priority levels can be set for each source in interrupt control registers, 16 levels for CPU1 and CPU2.

7A.9.9 Broadcast Function (Broadcast Communication Function)

All interrupts have an interrupt register and an interrupt-binding register.

Any single interrupt can be allocated to each PE by setting IBDxxx.CST to 0 (value after a reset) and IBDxxx.PEID[2:0] to a desired value.

Each interrupt which is type of edge detection can be conveyed to both PEs by using the broadcast function (setting IBDxxx.CST to 1). For detection type of each interrupt, refer to **Table 7A.4, EI Level Maskable Interrupt Sources**.

This product only supports the use of two broadcast interrupt. Accordingly, only two IBDxxx.CST bit can be set to 1 and broadcase destination (INTBN0 or INTBN1) can be selected by BCP0 bit setting.

All interrupts for INTC1 that have interrupt number 4 or 5 are allocated to the broadcast interrupt, INTBN0 or INTBN1.

- Allocating multiple interrupt channels to the broadcast interrupt is prohibited.
- The setting of the ICxxx.P[3:0] bits for the interrupt priority level of an INTxxx allocated as a broadcast interrupt is ignored.
- The ICBN0.MKBN0, ICBN1.MKBN1 interrupt mask bit of INTC1 can be used to mask broadcast interrupts for PEs.
- The interrupt vector for the broadcast interrupt of INTC1 (INTBN0, INTBN1) is the same for both PEs.

7A.9.9.1 Example of Operation

- The example is described about INTOSTM0 interrupt, which is allocated to interrupt number 84 of INTC2.
- Multiple-conveyance (broadcast) port 0 is in use (INTBN0 is in use).
- The broadcast interrupt (INTBN0) is connected to INTC1 as interrupt number 4.
- Set the IBDOSTM0.CST bit to 1_B (enabling broadcasting).
- Set the IBDOSTM0.BCP0 bit to 0_B (selecting broadcast port 0).
- Doing so leads to the allocation of the INTOSTM0 interrupt to interrupt number 4 of PE1 and PE2.
- The ICBN0.P[3:0]BN0 bits define the order of priority for this interrupt.
- The allocated mask bit is ICBN0.MKBN0 and ICBN0.RFBN0 is the interrupt request flag (accordingly, INTBN0 can be separately enabled or disabled for each PE in each INTC1).

  When INTOSTM0 is generated, this interrupt request is conveyed from INTC2 to INTC1 through the broadcast interrupt.

  Accordingly, the ICBN0.RFBN0 bit for INTC1 is set to 1_B in both PE1 and PE2.

  Both PEs execute the corresponding interrupt routine and each ICBN0.RFBN0 is automatically cleared.

  All register settings are listed in the table overleaf.

Table 7A.26 Register Settings (INTC2)

Register Name	Bit Name	Setting Value	Description
ICOSTM0	CTOSTM0	0	The user can only read the value.
	RFOSTM0	—	No function if the setting is for broadcast interrupts
	MKOSTM0	1/0	The user can set the value.
	TBOSTM0	—	No function if the setting is for broadcast interrupts
	P[3:0]OSTM0	—	No function if the setting is for broadcast interrupts
IBDOSTM0	CST	1	Interrupt input is conveyed to multiple destinations (broadcast).
	BCP0	0	Output to broadcast port 0
	PEID[2:0]	—	No function if the setting is for broadcast interrupts

Table 7A.27 Register Settings (INTC1)

Register Name	Bit Name	Setting Value	Description
ICBN0	CTBN0	0	The user can only read the value.
	RFBN0	1/0	Interrupt flag for broadcast interrupts
	MKBN0	1/0	Mask flag for broadcast interrupts
	TBBN0	1/0	The user can set the value.
	P[3:0]BN0	0 to 15	The user can set the value.
IBDBN0	CST	0	Fixed to 0
	BCP0	0	Fixed to 0
	PEID[2:0]	001 (PE1) 010 (PE2)	Fixed

7A.9.9.2 Inter-Processor Interrupt Flow

Figure 7A.7, Example of Inter-Processor Interrupt Processing Flow shows a flow example of inter-processor interrupt processing.

- Inter-processor interrupt generates an interrupt request by writing 1 to applicable bits of PE to which an interrupt of the inter-processor interrupt register (IPIR_CHn (n = 0 to 3)) are requested.
- The settings of interrupt request of the inter-PE interrupt registers (IPIR_CHn (n = 0 to 3)) are automatically cleared to 0 after notification of an interrupt request is complete.

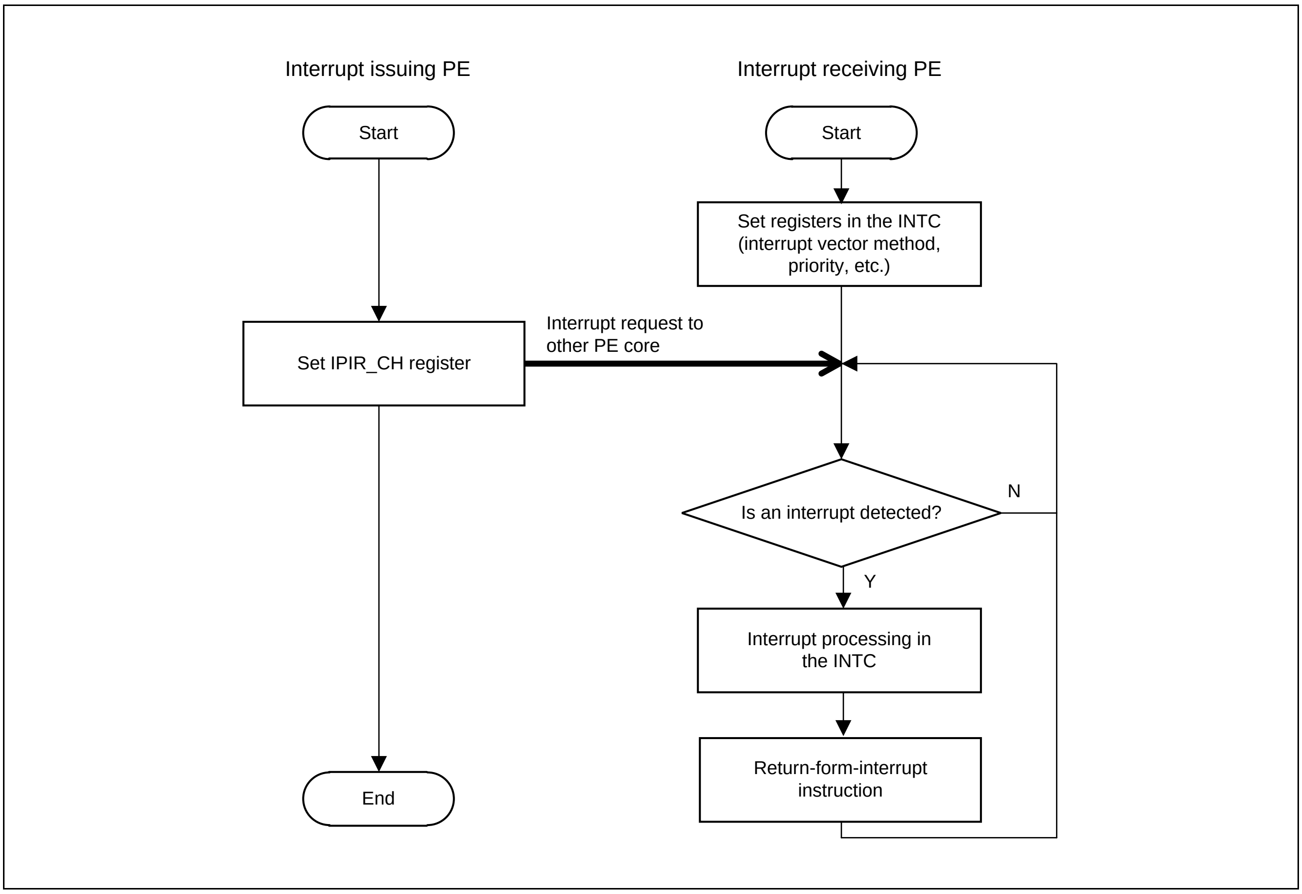


Figure 7A.7 Example of Inter-Processor Interrupt Processing Flow

7A.10 Exception Handler Address

In the RH850/F1KH, the exception handler address from which the handler is executed after a reset is input or when an exception or interrupt is acknowledged can be changed according to a setting.

The exception handler address for resets and exceptions (including interrupts) is determined with the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, the RBASE register, and the EBASE register. For interrupts, the direct vector method and table reference method can be selected for each channel. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory.

CAUTION

The exception handler address of EIINT (INTn) selected using the direct vector method differs from that of the V850E2 core products. In the V850E2 core products, a different exception handler address is individually assigned to each interrupt channel (EIINT (INTn)). In the RH850/F1KH, one exception handler address is assigned to each interrupt priority. Consequently, interrupts that have the same priority level branch to the same exception handler.

7A.10.1 Direct Vector Method

The CPU uses the result of adding the offset shown in **Table 7A.28, Selection of Base Register/Offset Address** to the base address indicated by the RBASE or EBASE register as the exception handler address.

Select whether the RBASE or EBASE register is used as the base address by using the PSW.EBV bit*[1]. When the PSW.EBV bit is set to 1, the value of the EBASE register is used as the base address. When the PSW.EBV bit is cleared to 0, the value of the RBASE register is used as the base address.

For reset input the RBASE register is always used for reference.

In addition, user interrupts see the RINT bit of the corresponding base register, and reduce the offset address according to the bit status. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, all user interrupts are handled using an offset of 100_H. If the bit is cleared to 0, the offset address is determined according to **Table 7A.28, Selection of Base Register/Offset Address**.

Note 1. Exception acknowledgment itself may sometimes update the status of the PSW.EBV bit. In this case, the base register is selected based on the updated value.

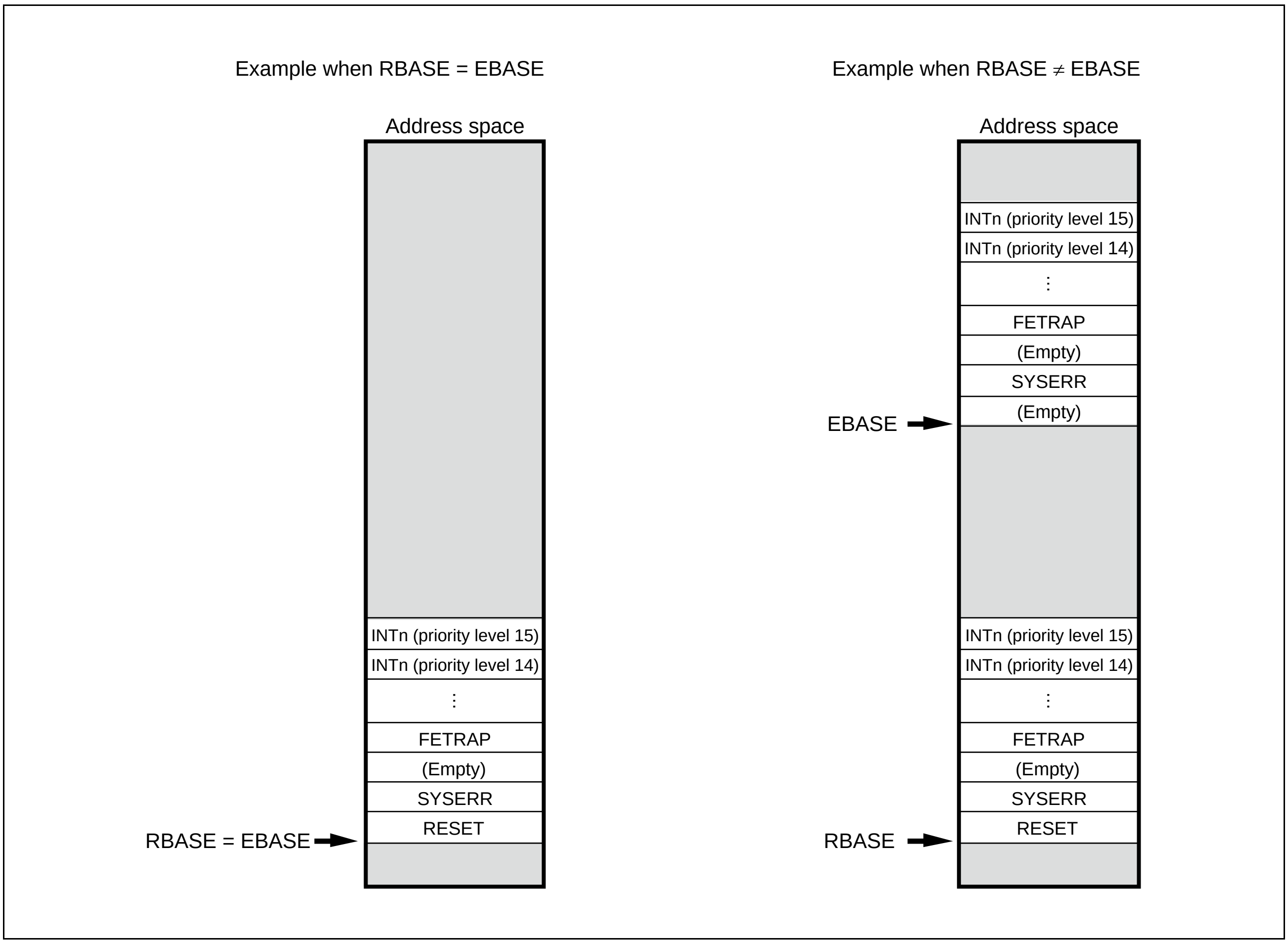


Figure 7A.8 Direct Vector Method

The table below shows how base register selection and offset address reduction function for each exception to determine the exception handler address. The value of the PSW bit determines the exception handler address on the basis of the value updated by the acknowledgment of an exception.

Table 7A.28 Selection of Base Register/Offset Address

	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
Exception/Interrupt	Base Register		Offset Address	
RESET	RBASE	N.A.	000_H	000_H
SYSERR		EBASE	010_H	010_H
Reserved			020_H	020_H
FETRAP			030_H	030_H
TRAP0			040_H	040_H
TRAP1			050_H	050_H
RIE			060_H	060_H
FPP/FPI			070_H	070_H
UCPOP			080_H	080_H
MIP/MDP			090_H	090_H
PIE			$0A0_H$	$0A0_H$
MAE			$0C0_H$	$0C0_H$
Reserved			$0D0_H$	$0D0_H$
FENMI			$0E0_H$	$0E0_H$
FEINT			$0F0_H$	$0F0_H$
INTn (Priority level 0)			100_H	100_H
INTn (Priority level 1)			110_H	
INTn (Priority level 2)			120_H	
INTn (Priority level 3)			130_H	
INTn (Priority level 4)			140_H	
INTn (Priority level 5)			150_H	
INTn (Priority level 6)			160_H	
INTn (Priority level 7)			170_H	
INTn (Priority level 8)			180_H	
INTn (Priority level 9)			190_H	
INTn (Priority level 10)			$1A0_H$	
INTn (Priority level 11)			$1B0_H$	
INTn (Priority level 12)			$1C0_H$	
INTn (Priority level 13)			$1D0_H$	
INTn (Priority level 14)			$1E0_H$	
INTn (Priority level 15)			$1F0_H$	

Base register selection is used to execute exception handling for resets and some hardware errors by using the programs in a relatively reliable area such as ROM instead of the areas that are easily affected by software errors such as RAM and cache area. The user interrupt offset address reduction function is used to reduce the memory occupation size required by the exception handler for specific system-internal operating modes. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

7A.10.2 Table Reference Method

With the direct vector method, there is one user-interrupt exception handler for each interrupt priority level, and interrupt channels that indicate multiple interrupts with the same priority branch to the same interrupt handler, but some users might want to use different code areas for each interrupt handler from the beginning.

The RH850/F1KH uses the table reference method for interrupts that assume the above usage.

If the table reference method is specified as the interrupt channel vector selection method in the interrupt controller and the like, the method for determining the exception handler address when an interrupt request corresponding to that interrupt channel is acknowledged differs as follows:

<1> In any of the following cases, the exception handler address is determined by using the direct vector method:

- When PSW.EBV = 0 and RBASE.RINT = 1
- When PSW.EBV = 1 and EBASE.RINT = 1
- When the interrupt channel setting is not the table reference method

<2> In cases other than <1>, calculate the table reference position.
Exception handler address read position = INTBP register + channel number × 4 bytes

<3> Read word data starting at the interrupt handler address read position calculated in <2>.

<4> Use the word data read in <3> as the exception handler address.

Table 7A.29, Exception Handler Address Expansion shows the exception handler address read positions corresponding to each interrupt channel and
Figure 7A.9, Table Reference Method shows an overview of the allocation in memory.

Table 7A.29 Exception Handler Address Expansion

Type of Interrupt	Exception Handler Address Read Position
EI level maskable interrupt channel 0	INTBP register value + 0 × 4
EI level maskable interrupt channel 1	INTBP register value + 1 × 4
EI level maskable interrupt channel 2	INTBP register value + 2 × 4
:	:
EI level maskable interrupt channel 380	INTBP register value + 380 × 4
EI level maskable interrupt channel 381	INTBP register value + 381 × 4

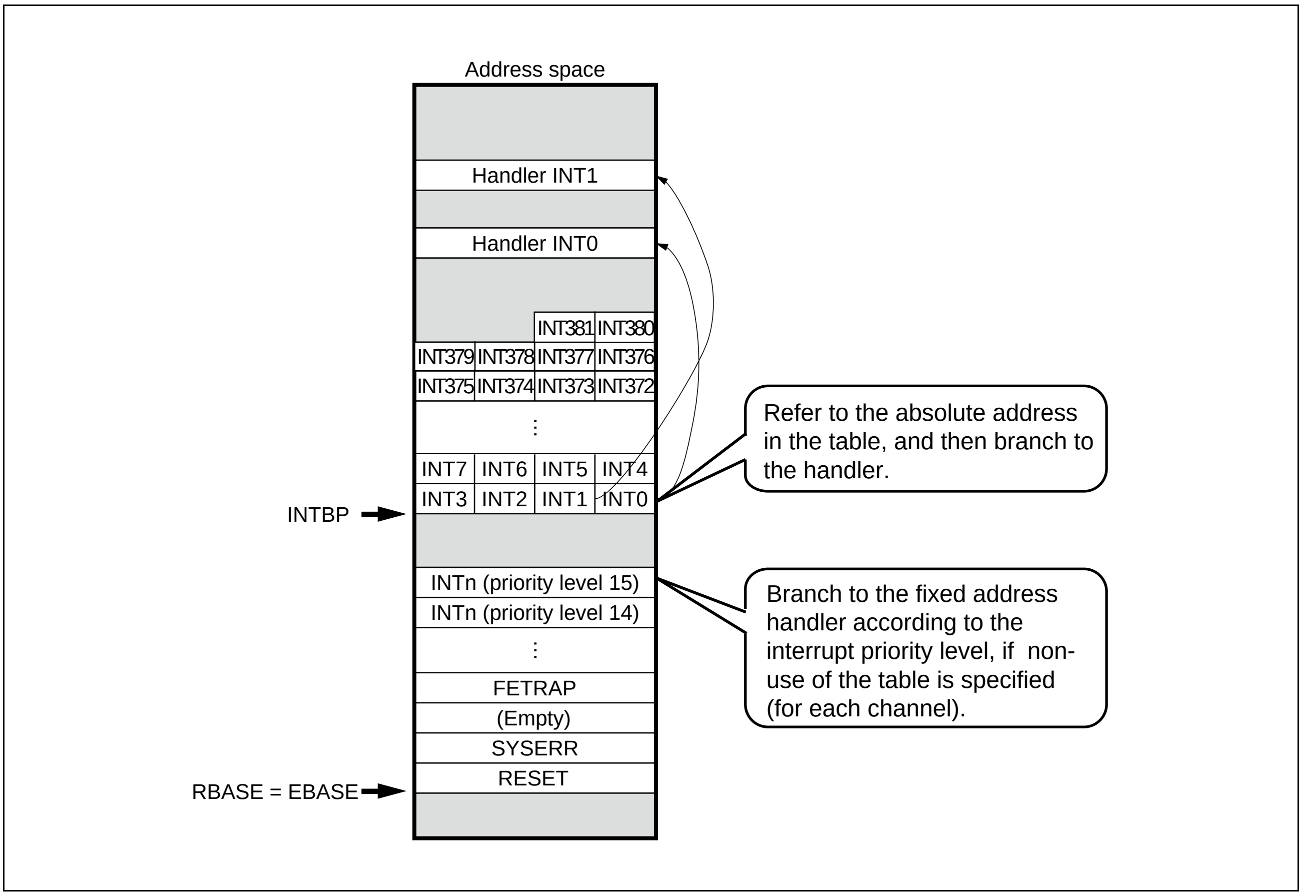


Figure 7A.9 Table Reference Method

Section 7BC Exception/Interrupts of RH850/F1KM

7BC.1 Features of RH850/F1KM Exception/Interrupts

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions.

The details on exceptions are described in *the RH850G3KH User's Manual: Software*.

Table 7BC.1 List of Exception Sources

Name	Symbol	Source	Priority	Saved to
Reset	RESET	Reset input	High	—
FE level non-maskable interrupt*[1]	FENMI	FENMI input		FE
System error exception	SYSERR	SYSERR input		FE
FE level maskable interrupt*[1]	FEINT	FEINT input		FE
Floating-point arithmetic exception (imprecise)	FPI	Execution of FPU instruction		EI
EI level maskable interrupt*[1]	EIINT	Interrupt controller		EI
Memory protection exception (execution right)	MIP	Memory protection violation		FE
System error exception	SYSERR	Error input at instruction fetch		FE
Reserved instruction exception	RIE	Execution of reserved instruction		FE
Coprocessor unusable exception	UCPOP	Execution of coprocessor instruction/access right violation		FE
Privileged instruction exception	PIE	Execution of privileged instruction		FE
Misaligned exception	MAE	Generation of misaligned access		FE
Memory protection exception (access right)	MDP	Memory protection violation		FE
Floating-point arithmetic exception (precise)	FPP	Execution of FPU instruction		EI
System call	SYSCALL	Execution of SYSCALL instruction		EI
FE level trap	FETRAP	Execution of FETRAP instruction		FE
EI level trap 0	TRAP0	Execution of TRAP instruction		EI
EI level trap 1	TRAP1	Execution of TRAP instruction	Low	EI

Note 1. These interrupt exceptions are described in this section.

(1) Interrupts

The following three exceptions in **Table 7BC.1, List of Exception Sources** are called interrupts, and are described in this section.

- FE level non-maskable interrupt (FENMI)
  An FENMI interrupt is acknowledged even if another FE level interrupt - FEINT - has been generated.
  - An FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
  - Return from an FENMI interrupt is not possible and neither is recovery.
- FE level maskable interrupt (FEINT)
  - FEINT can be acknowledged if the CPU system register PSW.NP = 0. It is masked if PSW.NP = 1.
  - Return from an FEINT interrupt is possible and so is recovery.
- EI level maskable interrupt (EIINT)
  An EIINT interrupt can be acknowledged if an FE level interrupt - FENMI or FEINT - has not been generated.
  - EIINT can be acknowledged if the CPU system register PSW.NP = 0.
    It is masked if PSW.NP = 1, EIINT with a higher priority is being processed, or PSW.ID = 1.
  - Return from an EIINT interrupt is possible and so is recovery.
  - Interrupt masking can be specified for each interrupt channel.
  - 16 interrupt priority levels can be specified for each interrupt channel
  - In this section, the EIINT that corresponds to interrupt channel n is indicated by “INTn”, whereas the EIINT that corresponds to interrupt source xxx is indicated by “INTxxx”.

For the PSW register, see **Table 3BC.10, PSW Register Contents** and *the RH850G3KH User’s Manual: Software*.

NOTE

Return:	Indicates whether or not the program can resume from where it was interrupted.
Recovery:	Indicates whether or not the processor status (status of processor resources including general-purpose registers and system registers) can be restored to the status they were in when the program was interrupted.

These interrupt sources are described in **Section 7BC.2, Interrupt Sources**.

(2) Overview of interrupts

- Priority levels for interrupt
  16 priority levels of maskable interrupts by request can be set by interrupt control register.
- Detecting methods of external interrupts (TNMI/INTPm)
  A method of detecting external interrupts (TNMI and INTPm) can be selected from five types:
  rising edge, falling edge, both edges, low level, and high level.
- 2 types of interrupt handler address setting Direct branching method or table referencing method is selectable by register setting.

7BC.2 Interrupt Sources

7BC.2.1 Interrupt Sources

7BC.2.1.1 FE Level Non-Maskable Interrupts

(1) Priority

See **Table 7BC.1, List of Exception Sources**.

(2) Return PC

Return or recovery from an FE non-maskable interrupt is not possible.

(3) Status Register

See **Section 7BC.4.4, FNC — FE Level NMI Status Register**.

(4) Return Instruction

None

Table 7BC.2 FE Level Non-Maskable Interrupt Requests

Interrupt			Interrupt Request					
	Control Register							
Symbol	Name	Address	Name	Source	Unit	Priority	Exception Source Code	Handler Address 00000...
FENMI	FNC	FFFE $EA78_H$	TNMI	NMI pin	Port	*1	$0E0_H$	$0E0_H$
			WDTA0NMI	WDTA0 FENMI interrupt	WDTA0			
			WDTA1NMI	WDTA1 FENMI interrupt	WDTA1			

Note 1. See **Table 7BC.1, List of Exception Sources**.

The source of the FENMI interrupt can be evaluated by a dedicated flag register. See **Section 7BC.2.2, FE Level Non-Maskable Interrupt Sources** for details.

7BC.2.1.2 FE Level Maskable Interrupts

(1) Priority

See **Table 7BC.1, List of Exception Sources**.

(2) Return PC

The return PC returned from an interrupt handling routine by the FERET instruction is the PC from when the program was suspended (current PC).

(3) Status Register

See **Section 7BC.4.5, FIC — FE Level Maskable Interrupt Status Register**.

(4) Return Instruction

FERET

Table 7BC.3 FE Level Maskable Interrupt Requests (RH850/F1KM-S4)

Interrupt			Interrupt Request										
	Control Register												
Symbol	Name	Address	Name	Source	Unit	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Priority	Exception Source Code	Handler Address 00000...
FEINT	FIC	FFFE EA7A$_H$	INTLVIL	LVI voltage detection (falling)	LVI	✓	✓	✓	✓	✓	*1	0F0$_H$	0F0$_H$
			INTECCEEP0	Data flash ECC 1-bit error or 2-bit error interrupt	Data flash	✓	✓	✓	✓	✓			
			INTECCFLRAM	FLXA0 ECC 1-bit error or 2-bit error interrupt	FLXA0	✓	✓	✓	✓	✓			
			INTECCCNFDRAM	RCFDC0 ECC 1-bit error or 2-bit error interrupt	RCFDC0	✓	✓	✓	✓	✓			
			INTECCCSIH0	CSIH0 ECC 1-bit error or 2-bit error interrupt	CSIH0	✓	✓	✓	✓	✓			
			INTECCCSIH1	CSIH1 ECC 1-bit error or 2-bit error interrupt	CSIH1	✓	✓	✓	✓	✓			
			INTECCCSIH2	CSIH2 ECC 1-bit error or 2-bit error interrupt	CSIH2	✓	✓	✓	✓	✓			
			INTECCCSIH3	CSIH3 ECC 1-bit error or 2-bit error interrupt	CSIH3	✓	✓	✓	✓	✓			
			INTECCFLI0	Code Flash ECC 1-bit error or 2-bit error interrupt	Code flash	✓	✓	✓	✓	✓			
			INTECCRAM	RAM ECC 1-bit error or 2-bit error interrupt	RAM	✓	✓	✓	✓	✓			
			INTOSTM0_FE*2	OSTM0 interrupt	OSTM0	✓	✓	✓	✓	✓			
			INTLVIH	LVI voltage detection (rising)	LVI	✓	✓	✓	✓	✓			
			INTGUARD	PE Guard (PEG) error or Global RAM Guard (GRG) error or Peripheral Guard (PBG/HBG/PBGC)	PEG, GRG, PBG/HBG/PBGC	✓	✓	✓	✓	✓			
			INTOSTM1_FE	OSTM1 interrupt	OSTM1	✓	✓	✓	✓	✓			
			INTOSTM2_FE	OSTM2 interrupt	OSTM2	✓	✓	✓	✓	✓			
			INTOSTM3_FE	OSTM3 interrupt	OSTM3	✓	✓	✓	✓	✓			
			INTOSTM4_FE	OSTM4 interrupt	OSTM4	✓	✓	✓	✓	✓			
			INTECCETH	ETNB0 ECC 1-bit error or 2-bit error interrupt	ETNB0	—	—	✓	✓	✓			
			INTDMAERR	DMA transfer error interrupt	PDMA0	✓	✓	✓	✓	✓			

Note 1. See **Table 7BC.1, List of Exception Sources**.

Note 2. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is prohibited. When INTOSTM0 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM0.

* Timing monitor (TSU)
This prevents the illicit use of CPU time by non-trusted programs, manages properties, and controls the intervals over which interrupts are disabled.

Table 7BC.4 FE Level Maskable Interrupt Requests (RH850/F1KM-S1)

Interrupt			Interrupt Request									
	Control Register											
Symbol	Name	Address	Name	Source	Unit	48 Pins	64 Pins	80 Pins	100 Pins	Priority	Exception Source Code	Handler Address 00000...
FEINT	FIC	FFFE EA7A_H	INTLVIL	LVI voltage detection (falling)	LVI	✓	✓	✓	✓	*1	0F0_H	0F0_H
			INTECCEEP0	Data flash ECC 1-bit error or 2-bit error interrupt	Data flash	✓	✓	✓	✓			
			INTECCCNFDRAM	RCFDC0 ECC 1-bit error or 2-bit error interrupt	RCFDC0	✓	✓	✓	✓			
			INTECCCSIH0	CSIH0 ECC 1-bit error or 2-bit error interrupt	CSIH0	✓	✓	✓	✓			
			INTECCCSIH1	CSIH1 ECC 1-bit error or 2-bit error interrupt	CSIH1	—	—	✓	✓			
			INTECCCSIH2	CSIH2 ECC 1-bit error or 2-bit error interrupt	CSIH2	—	—	✓	✓			
			INTECCCSIH3	CSIH3 ECC 1-bit error or 2-bit error interrupt	CSIH3	—	—	—	✓			
			INTECCFLI0	Code Flash ECC 1-bit error or 2-bit error interrupt	Code flash	✓	✓	✓	✓			
			INTECCRAM	RAM ECC 1-bit error or 2-bit error interrupt	RAM	✓	✓	✓	✓			
			INTOSTM0_FE*2	OSTM0 interrupt	OSTM0	✓	✓	✓	✓			
			INTLVIH	LVI voltage detection (rising)	LVI	✓	✓	✓	✓			
			INTGUARD	PE Guard (PEG) error or Peripheral Guard (PBG/PBGC)	PEG, PBG/PBGC	✓	✓	✓	✓			
			INTDMAERR	DMA transfer error interrupt	PDMA0	✓	✓	✓	✓			

Note 1. See **Table 7BC.1, List of Exception Sources**.

Note 2. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is prohibited. When INTOSTM0 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM0.

* Timing monitor (TSU)
This prevents the illicit use of CPU time by non-trusted programs, manages properties, and controls the intervals over which interrupts are disabled.

7BC.2.1.3 EI Level Maskable Interrupts

(1) Interrupt Naming Rules

The composition of the interrupt request signal names, their assigned interrupt control registers and the bits in these registers follow special rules.

In the following the name of the specific interrupt request is represented by <*name*>.

For details of the names used in **IC**<*name*>, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

- Interrupt request name: **INT**<*name*>
  The prefix "**INT**" is appended to <*name*>.
- Interrupt request control register: **IC**<*name*>
  The prefix "**IC**" is appended to <*name*>.
  The 16-bit register **IC**<*name*> can also be accessed in byte units:
  - Low byte (bits [7:0]) of the **IC** <*name*> register: **IC** <*name*>L
    The suffix "L" is appended to the register name **IC**<*name*>.
  - High byte (bits [15:8]) of the **IC** <*name*> register: **IC** <*name*>H
    The suffix "H" is appended to the register name **IC**<*name*>.
- Interrupt control register bit names: **CT**<*name*>, **RF**<*name*>, **MK**<*name*>, **TB**<*name*>, **P3**<*name*>, **P2**<*name*>, **P1**<*name*>, **P0**<*name*>
  The bit prefix "**CT**", "**RF**", "**MK**", "**TB**", "**P3**", "**P2**", "**P1**", or "**P0**" is appended to the interrupt <*name*>.

Example

The interrupt request from channel 2 of TAUD0 channel (<*name*> = *TAUD0I2*) is named

INTTAUD0I2

The related interrupt control registers are

ICTAUD0I2, **IC**TAUD0I2L, **IC**TAUD0I2H

The bits in this register are

CTTAUD0I2,RFTAUD0I2, MKTAUD0I2, TBTAUD0I2, P3TAUD0I2, P2TAUD0I2,
P1TAUD0I2, P0TAUD0I2

(2) Priority

See **Table 7BC.1, List of Exception Sources**.

(3) Return PC

The return PC returned from an interrupt handling routine by the EIRET instruction is the PC from when the program was suspended (current PC).

(4) Control Register

EI level maskable interrupt control register

See **Section 7BC.4.2, ICxxx — EI Level Interrupt Control Registers**.

(5) Return Instruction

EIRET instruction

(6) Configuration

EI-level maskable interrupts are controlled by the two controllers, INTC1 and INTC2. The interrupts are supported on a total of 377 channels with a cascade connection of INTC1 and INTC2.

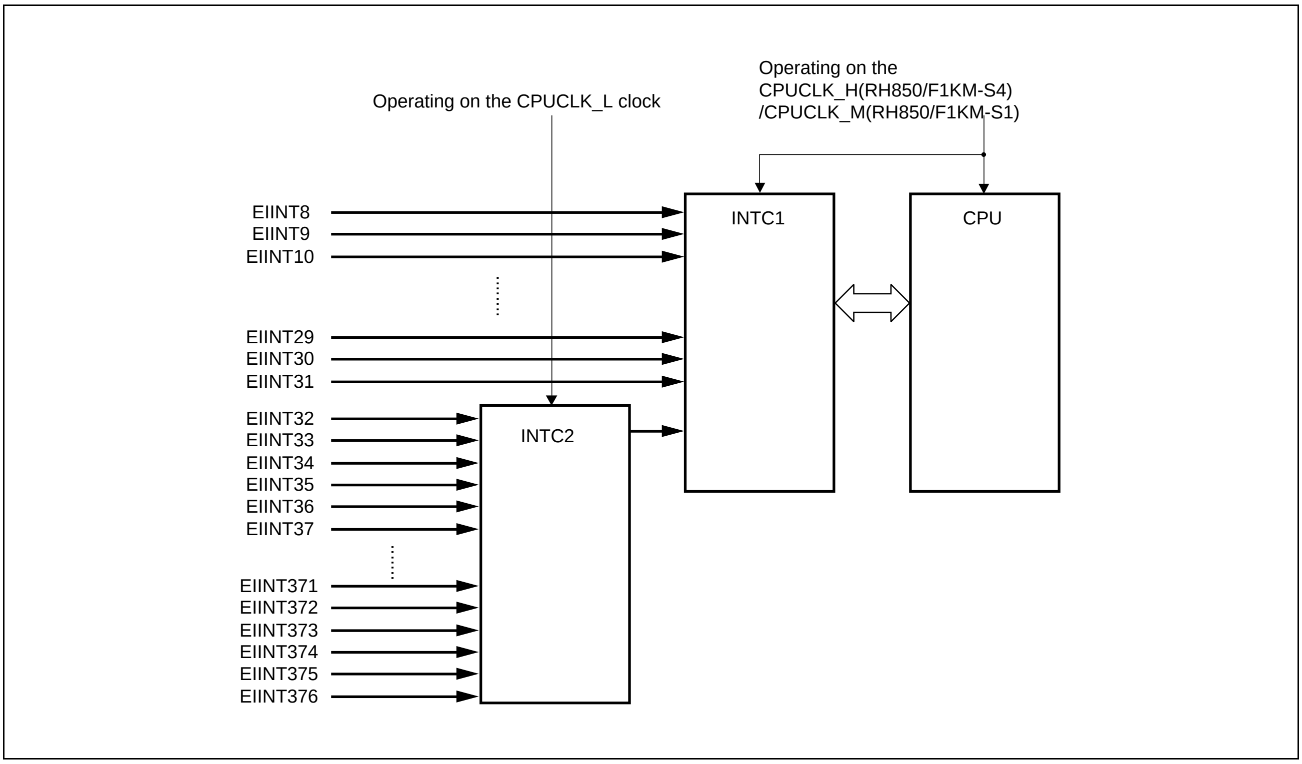


Figure 7BC.1 Configuration Diagram of EI Level Maskable Interrupt

CAUTION

As CPUCLK_L is the operating clock for INTC2, the EIINT32 to EIINT376 interrupts, which are connected to INTC2, are delayed more than the interrupts directly connected to INTC1.

Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4), **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)** lists EI level maskable interrupts.

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2							Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
0	Reserved	FFFE EA00$_H$					1000$_H$	—	—	—	—	—	*3	*4	+000$_H$
1	Reserved	FFFE EA02$_H$					1001$_H$	—	—	—	—	—			+004$_H$
2	Reserved	FFFE EA04$_H$					1002$_H$	—	—	—	—	—			+008$_H$
3	Reserved	FFFE EA06$_H$					1003$_H$	—	—	—	—	—			+00C$_H$
4	Reserved	FFFE EA08$_H$					1004$_H$	—	—	—	—	—			+010$_H$
5	Reserved	FFFE EA0AH					1005$_H$	—	—	—	—	—			+014$_H$
6	Reserved	FFFE EA0C$_H$					1006$_H$	—	—	—	—	—			+018$_H$
7	Reserved	FFFE EA0E$_H$					1007$_H$	—	—	—	—	—			+01C$_H$
8	ICTAUD0I0	FFFE EA10$_H$	INTTAUD0I0*6	Interrupt for CH0 of TAUD0	TAUD0	Edge	1008$_H$	✓	✓	✓	✓	✓			+020$_H$
	ICCSIH2IC_1		INTCSIH2IC_1*6	CSIH2 communication status interrupt	CSIH2	Edge									
9	ICTAUD0I2	FFFE EA12$_H$	INTTAUD0I2*6	Interrupt for CH2 of TAUD0	TAUD0	Edge	1009$_H$	✓	✓	✓	✓	✓			+024$_H$
	ICCSIH3IC_1		INTCSIH3IC_1*6	CSIH3 communication status interrupt	CSIH3	Edge									
10	ICTAUD0I4	FFFE EA14$_H$	INTTAUD0I4*6	Interrupt for CH4 of TAUD0	TAUD0	Edge	100A$_H$	✓	✓	✓	✓	✓			+028$_H$
	ICCSIH2IR_1		INTCSIH2IR_1*6	CSIH2 receive status interrupt	CSIH2	Edge									
11	ICTAUD0I6	FFFE EA16$_H$	INTTAUD0I6*6	Interrupt for CH6 of TAUD0	TAUD0	Edge	100B$_H$	✓	✓	✓	✓	✓			+02C$_H$
	ICCSIH2IRE_1		INTCSIH2IRE_1*6	CSIH2 communication error interrupt	CSIH2	Edge									
12	ICTAUD0I8	FFFE EA18$_H$	INTTAUD0I8*6	Interrupt for CH8 of TAUD0	TAUD0	Edge	100C$_H$	✓	✓	✓	✓	✓			+030$_H$
	ICCSIH2IJC_1		INTCSIH2IJC_1*6	CSIH2 job completion interrupt	CSIH2	Edge									
13	ICTAUD0I10	FFFE EA1A$_H$	INTTAUD0I10*6	Interrupt for CH10 of TAUD0	TAUD0	Edge	100D$_H$	✓	✓	✓	✓	✓			+034$_H$
	ICCSIH3IR_1		INTCSIH3IR_1*6	CSIH3 receive status interrupt	CSIH3	Edge									
14	ICTAUD0I12	FFFE EA1C$_H$	INTTAUD0I12*6	Interrupt for CH12 of TAUD0	TAUD0	Edge	100E$_H$	✓	✓	✓	✓	✓			+038$_H$
	ICCSIH3IRE_1		INTCSIH3IRE_1*6	CSIH3 communication error interrupt	CSIH3	Edge									
15	ICTAUD0I14	FFFE EA1E$_H$	INTTAUD0I14*6	Interrupt for CH14 of TAUD0	TAUD0	Edge	100F$_H$	✓	✓	✓	✓	✓			+03C$_H$
	ICCSIH3IJC_1		INTCSIH3IJC_1*6	CSIH3 job completion interrupt	CSIH3	Edge									
16	ICTAPA0IPEK0	FFFE EA20$_H$	INTTAPA0IPEK0*6	TAPA0 peak interrupt 0	TAPA0	Edge	1010$_H$	✓	✓	✓	✓	✓			+040$_H$
	ICCSIH1IC_1		INTCSIH1IC_1*6	CSIH1 communication status interrupt	CSIH1	Edge									
17	ICTAPA0IVLY0	FFFE EA22$_H$	INTTAPA0IVLY0*6	TAPA0 valley interrupt 0	TAPA0	Edge	1011$_H$	✓	✓	✓	✓	✓			+044$_H$
	ICCSIH1IR_1		INTCSIH1IR_1*6	CSIH1 receive status interrupt	CSIH1	Edge									
18	ICADCA0I0	FFFE EA24$_H$	INTADCA0I0	ADCA0 scan group 1 (SG1) end interrupt	ADCA0	Edge	1012$_H$	✓	✓	✓	✓	✓			+048$_H$
19	ICADCA0I1	FFFE EA26$_H$	INTADCA0I1	ADCA0 scan group 2 (SG2) end interrupt	ADCA0	Edge	1013$_H$	✓	✓	✓	✓	✓			+04C$_H$
20	ICADCA0I2	FFFE EA28$_H$	INTADCA0I2*6	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge	1014$_H$	✓	✓	✓	✓	✓			+050$_H$
	ICCSIH0IJC_1		INTCSIH0IJC_1*6	CSIH0 job completion interrupt	CSIH0	Edge									
21	ICDCUTDI	FFFE EA2A$_H$	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	1015$_H$	✓	✓	✓	✓	✓			+054$_H$
22	ICRCANGERR0	FFFE EA2C$_H$	INTRCANGERR0	CAN global error interrupt	RCFDC0	Level	1016$_H$	✓	✓	✓	✓	✓			+058$_H$
23	ICRCANGRECC0	FFFE EA2E$_H$	INTRCANGRECC0	CAN receive FIFO interrupt	RCFDC0	Level	1017$_H$	✓	✓	✓	✓	✓			+05C$_H$
24	ICRCAN0ERR	FFFE EA30$_H$	INTRCAN0ERR	CAN0 error interrupt	RCFDC0	Level	1018$_H$	✓	✓	✓	✓	✓			+060$_H$
25	ICRCAN0REC	FFFE EA32$_H$	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1019$_H$	✓	✓	✓	✓	✓			+064$_H$
26	ICRCAN0TRX	FFFE EA34$_H$	INTRCAN0TRX	CAN0 transmit interrupt	RCFDC0	Level	101A$_H$	✓	✓	✓	✓	✓			+068$_H$
27	ICCSIG0IC	FFFE EA36$_H$	INTCSIG0IC*6	CSIG0 communication status interrupt	CSIG0	Edge	101B$_H$	✓	✓	✓	✓	✓			+06C$_H$
	ICCSIH1IRE_1		INTCSIH1IRE_1*6	CSIH1 communication error interrupt	CSIH1	Edge									

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register												Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							RINT = 0	RINT = 1	
28	ICCSIG0IR	FFFE EA38$_H$	INTCSIG0IR*6	CSIG0 receive status interrupt	CSIG0	Edge	101C$_H$	✓	✓	✓	✓	✓	*3	*4	+070$_H$
	ICCSIH1IJC_1		INTCSIH1IJC_1*6	CSIH1 job completion interrupt	CSIH1	Edge									
29	ICCSIH0IC	FFFE EA3A$_H$	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	101D$_H$	✓	✓	✓	✓	✓			+074$_H$
30	ICCSIH0IR	FFFE EA3C$_H$	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	101E$_H$	✓	✓	✓	✓	✓			+078$_H$
31	ICCSIH0IRE	FFFE EA3E$_H$	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	101F$_H$	✓	✓	✓	✓	✓			+07C$_H$
32	ICCSIH0IJC	FFFF B040$_H$	INTCSIH0IJC*6	CSIH0 job completion interrupt	CSIH0	Edge	1020$_H$	✓	✓	✓	✓	✓			+080$_H$
	ICADCA0I2_2		INTADCA0I2_2*6	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge									
33	ICRLIN30	FFFF B042$_H$	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1021$_H$	✓	✓	✓	✓	✓			+084$_H$
34	ICRLIN30UR0	FFFF B044$_H$	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	1022$_H$	✓	✓	✓	✓	✓			+088$_H$
35	ICRLIN30UR1	FFFF B046$_H$	INTRLIN30UR1	RLIN30 receive complete interrupt	RLIN30	Edge	1023$_H$	✓	✓	✓	✓	✓			+08C$_H$
36	ICRLIN30UR2	FFFF B048$_H$	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	1024$_H$	✓	✓	✓	✓	✓			+090$_H$
37	ICP0	FFFF B04A$_H$	INTP0	External interrupt	Port	Edge	1025$_H$	✓	✓	✓	✓	✓			+094$_H$
38	ICP1	FFFF B04C$_H$	INTP1	External interrupt	Port	Edge	1026$_H$	✓	✓	✓	✓	✓			+098$_H$
39	ICP2	FFFF B04E$_H$	INTP2	External interrupt	Port	Edge	1027$_H$	✓	✓	✓	✓	✓			+09C$_H$
40	ICWDTA0	FFFF B050$_H$	INTWDTA0	WDTA0 75% interrupt	WDTA0	Edge	1028$_H$	✓	✓	✓	✓	✓			+0A0$_H$
41	ICWDTA1	FFFF B052$_H$	INTWDTA1	WDTA1 75% interrupt	WDTA1	Edge	1029$_H$	✓	✓	✓	✓	✓			+0A4$_H$
42	Reserved	FFFF B054$_H$					102A$_H$	—	—	—	—	—			+0A8$_H$
43	ICP3	FFFF B056$_H$	INTP3	External interrupt	Port	Edge	102B$_H$	✓	✓	✓	✓	✓			+0AC$_H$
44	ICP4	FFFF B058$_H$	INTP4	External interrupt	Port	Edge	102C$_H$	✓	✓	✓	✓	✓			+0B0$_H$
45	ICP5	FFFF B05A$_H$	INTP5	External interrupt	Port	Edge	102D$_H$	✓	✓	✓	✓	✓			+0B4$_H$
46	ICP10	FFFF B05C$_H$	INTP10	External interrupt	Port	Edge	102E$_H$	✓	✓	✓	✓	✓			+0B8$_H$
47	ICP11	FFFF B05E$_H$	INTP11	External interrupt	Port	Edge	102F$_H$	✓	✓	✓	✓	✓			+0BC$_H$
48	ICTAUD0I1	FFFF B060$_H$	INTTAUD0I1	Interrupt for TAUD0 channel 1	TAUD0	Edge	1030$_H$	✓	✓	✓	✓	✓			+0C0$_H$
49	ICTAUD0I3	FFFF B062$_H$	INTTAUD0I3	Interrupt for TAUD0 channel 3	TAUD0	Edge	1031$_H$	✓	✓	✓	✓	✓			+0C4$_H$
50	ICTAUD0I5	FFFF B064$_H$	INTTAUD0I5	Interrupt for TAUD0 channel 5	TAUD0	Edge	1032$_H$	✓	✓	✓	✓	✓			+0C8$_H$
51	ICTAUD0I7	FFFF B066$_H$	INTTAUD0I7	Interrupt for TAUD0 channel 7	TAUD0	Edge	1033$_H$	✓	✓	✓	✓	✓			+0CC$_H$
52	ICTAUD0I9	FFFF B068$_H$	INTTAUD0I9	Interrupt for TAUD0 channel 9	TAUD0	Edge	1034$_H$	✓	✓	✓	✓	✓			+0D0$_H$
53	ICTAUD0I11	FFFF B06A$_H$	INTTAUD0I11	Interrupt for TAUD0 channel 11	TAUD0	Edge	1035$_H$	✓	✓	✓	✓	✓			+0D4$_H$
54	ICTAUD0I13	FFFF B06C$_H$	INTTAUD0I13	Interrupt for TAUD0 channel 13	TAUD0	Edge	1036$_H$	✓	✓	✓	✓	✓			+0D8$_H$
55	ICTAUD0I15	FFFF B06E$_H$	INTTAUD0I15	Interrupt for TAUD0 channel 15	TAUD0	Edge	1037$_H$	✓	✓	✓	✓	✓			+0DC$_H$
56	ICADCA0ERR	FFFF B070$_H$	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	1038$_H$	✓	✓	✓	✓	✓			+0E0$_H$
57	ICCSIG0IRE	FFFF B072$_H$	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1039$_H$	✓	✓	✓	✓	✓			+0E4$_H$
58	ICRLIN20	FFFF B074$_H$	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	103A$_H$	✓	✓	✓	✓	✓			+0E8$_H$
59	ICRLIN21	FFFF B076$_H$	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	103B$_H$	✓	✓	✓	✓	✓			+0EC$_H$
60	ICDMA0	FFFF B078$_H$	INTDMA0	DMA00 transfer completion	PDMA0	Edge	103C$_H$	✓	✓	✓	✓	✓			+0F0$_H$
61	ICDMA1	FFFF B07A$_H$	INTDMA1	DMA01 transfer completion	PDMA0	Edge	103D$_H$	✓	✓	✓	✓	✓			+0F4$_H$
62	ICDMA2	FFFF B07C$_H$	INTDMA2	DMA02 transfer completion	PDMA0	Edge	103E$_H$	✓	✓	✓	✓	✓			+0F8$_H$
63	ICDMA3	FFFF B07E$_H$	INTDMA3	DMA03 transfer completion	PDMA0	Edge	103F$_H$	✓	✓	✓	✓	✓			+0FC$_H$
64	ICDMA4	FFFF B080$_H$	INTDMA4	DMA04 transfer completion	PDMA0	Edge	1040$_H$	✓	✓	✓	✓	✓			+100$_H$
65	ICDMA5	FFFF B082$_H$	INTDMA5	DMA05 transfer completion	PDMA0	Edge	1041$_H$	✓	✓	✓	✓	✓			+104$_H$
66	ICDMA6	FFFF B084$_H$	INTDMA6	DMA06 transfer completion	PDMA0	Edge	1042$_H$	✓	✓	✓	✓	✓			+108$_H$
67	ICDMA7	FFFF B086$_H$	INTDMA7	DMA07 transfer completion	PDMA0	Edge	1043$_H$	✓	✓	✓	✓	✓			+10C$_H$
68	ICDMA8	FFFF B088$_H$	INTDMA8	DMA10 transfer completion	PDMA0	Edge	1044$_H$	✓	✓	✓	✓	✓			+110$_H$
69	ICDMA9	FFFF B08A$_H$	INTDMA9	DMA11 transfer completion	PDMA0	Edge	1045$_H$	✓	✓	✓	✓	✓			+114$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register					Detection Type[*2]							Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
70	ICDMA10	FFFF B08C$_H$	INTDMA10	DMA12 transfer completion	PDMA0	Edge	1046$_H$	✓	✓	✓	✓	✓	[*3]	[*4]	+118$_H$
71	ICDMA11	FFFF B08E$_H$	INTDMA11	DMA13 transfer completion	PDMA0	Edge	1047$_H$	✓	✓	✓	✓	✓			+11C$_H$
72	ICDMA12	FFFF B090$_H$	INTDMA12	DMA14 transfer completion	PDMA0	Edge	1048$_H$	✓	✓	✓	✓	✓			+120$_H$
73	ICDMA13	FFFF B092$_H$	INTDMA13	DMA15 transfer completion	PDMA0	Edge	1049$_H$	✓	✓	✓	✓	✓			+124$_H$
74	ICDMA14	FFFF B094$_H$	INTDMA14	DMA16 transfer completion	PDMA0	Edge	104A$_H$	✓	✓	✓	✓	✓			+128$_H$
75	ICDMA15	FFFF B096$_H$	INTDMA15	DMA17 transfer completion	PDMA0	Edge	104B$_H$	✓	✓	✓	✓	✓			+12C$_H$
76	ICRIIC0TI	FFFF B098$_H$	INTRIIC0TI	RIIC0 transmit data empty interrupt	RIIC0	Edge	104C$_H$	✓	✓	✓	✓	✓			+130$_H$
77	ICRIIC0EE	FFFF B09A$_H$	INTRIIC0EE	RIIC0 receive error/event interrupt	RIIC0	Level	104D$_H$	✓	✓	✓	✓	✓			+134$_H$
78	ICRIIC0RI	FFFF B09C$_H$	INTRIIC0RI	RIIC0 receive complete interrupt	RIIC0	Edge	104E$_H$	✓	✓	✓	✓	✓			+138$_H$
79	ICRIIC0TEI	FFFF B09E$_H$	INTRIIC0TEI	RIIC0 transmit complete interrupt	RIIC0	Level	104F$_H$	✓	✓	✓	✓	✓			+13C$_H$
80	ICTAUJ0I0	FFFF B0A0$_H$	INTTAUJ0I0	Interrupt for TAUJ0 channel 0	TAUJ0	Edge	1050$_H$	✓	✓	✓	✓	✓			+140$_H$
81	ICTAUJ0I1	FFFF B0A2$_H$	INTTAUJ0I1	Interrupt for TAUJ0 channel 1	TAUJ0	Edge	1051$_H$	✓	✓	✓	✓	✓			+144$_H$
82	ICTAUJ0I2	FFFF B0A4$_H$	INTTAUJ0I2	Interrupt for TAUJ0 channel 2	TAUJ0	Edge	1052$_H$	✓	✓	✓	✓	✓			+148$_H$
83	ICTAUJ0I3	FFFF B0A6$_H$	INTTAUJ0I3	Interrupt for TAUJ0 channel 3	TAUJ0	Edge	1053$_H$	✓	✓	✓	✓	✓			+14C$_H$
84	ICOSTM0	FFFF B0A8$_H$	INTOSTM0[*9]	OSTM0 interrupt	OSTM0	Edge	1054$_H$	✓	✓	✓	✓	✓			+150$_H$
85	ICENCA0IOV	FFFF B0AA$_H$	INTENCA0IOV	ENCA0 overflow interrupt	ENCA0	Edge	1055$_H$	✓	✓	✓	✓	✓			+154$_H$
86	ICENCA0IUD	FFFF B0AC$_H$	INTENCA0IUD	ENCA0 underflow interrupt	ENCA0	Edge	1056$_H$	✓	✓	✓	✓	✓			+158$_H$
87	ICENCA0I0	FFFF B0AE$_H$	INTENCA0I0	ENCA0 match/capture interrupt 0	ENCA0	Edge	1057$_H$	✓	✓	✓	✓	✓			+15C$_H$
88	ICENCA0I1	FFFF B0B0$_H$	INTENCA0I1	ENCA0 match/capture interrupt 1	ENCA0	Edge	1058$_H$	✓	✓	✓	✓	✓			+160$_H$
89	ICENCA0IEC	FFFF B0B2$_H$	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	Edge	1059$_H$	✓	✓	✓	✓	✓			+164$_H$
90	ICKR0	FFFF B0B4$_H$	INTKR0	KR0 key interrupt	KR0	Edge	105A$_H$	✓	✓	✓	✓	✓			+168$_H$
91	ICQFULL	FFFF B0B6$_H$	INTQFULL	PWSA0 queue full interrupt	PWSA0	Edge	105B$_H$	✓	✓	✓	✓	✓			+16C$_H$
92	ICPWGAG00	FFFF B0B8$_H$	INTPWGAG00[*7]	PWGA interrupt group 00 (PWGA0 to PWGA31)	PWGA0-31	Level	105C$_H$	✓	✓	✓	✓	✓			+170$_H$
93	ICPWGAG01	FFFF B0BA$_H$	INTPWGAG01[*7]	PWGA interrupt group 01 (PWGA32 to PWGA63)	PWGA32-63	Level	105D$_H$	✓	✓	✓	✓	✓			+174$_H$
94	ICPWGAG02	FFFF B0BC$_H$	INTPWGAG02[*7]	PWGA interrupt group 02 (PWGA64 to PWGA95)	PWGA64-95	Level	105E$_H$	—	—	✓	✓	✓			+178$_H$
95	Reserved	FFFF B0BE$_H$					105F$_H$	—	—	—	—	—			+17C$_H$
96	Reserved	FFFF B0C0$_H$					1060$_H$	—	—	—	—	—			+180$_H$
97	Reserved	FFFF B0C2$_H$					1061$_H$	—	—	—	—	—			+184$_H$
98	Reserved	FFFF B0C4$_H$					1062$_H$	—	—	—	—	—			+188$_H$
99	Reserved	FFFF B0C6$_H$					1063$_H$	—	—	—	—	—			+18C$_H$
100	Reserved	FFFF B0C8$_H$					1064$_H$	—	—	—	—	—			+190$_H$
101	Reserved	FFFF B0CA$_H$					1065$_H$	—	—	—	—	—			+194$_H$
102	Reserved	FFFF B0CC$_H$					1066$_H$	—	—	—	—	—			+198$_H$
103	Reserved	FFFF B0CE$_H$					1067$_H$	—	—	—	—	—			+19C$_H$
104	Reserved	FFFF B0D0$_H$					1068$_H$	—	—	—	—	—			+1A0$_H$
105	Reserved	FFFF B0D2$_H$					1069$_H$	—	—	—	—	—			+1A4$_H$
106	Reserved	FFFF B0D4$_H$					106A$_H$	—	—	—	—	—			+1A8$_H$
107	Reserved	FFFF B0D6$_H$					106B$_H$	—	—	—	—	—			+1AC$_H$
108	Reserved	FFFF B0D8$_H$					106C$_H$	—	—	—	—	—			+1B0$_H$
109	Reserved	FFFF B0DA$_H$					106D$_H$	—	—	—	—	—			+1B4$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2							Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
110	ICFLERR	FFFF B0DC$_H$	INTFLERR*10	Flash sequencer end error interrupt	FACI	Level	106E$_H$	✓	✓	✓	✓	✓	*3	*4	+1B8$_H$
111	ICFLENDNM	FFFF B0DE$_H$	INTFLENDNM*10	Flash sequencer end interrupt	FACI	Edge	106F$_H$	✓	✓	✓	✓	✓			+1BC$_H$
112	ICCWEND	FFFF B0E0$_H$	INTCWEND	LPS0 port polling end interrupt	LPS0	Edge	1070$_H$	✓	✓	✓	✓	✓			+1C0$_H$
113	ICRCAN1ERR	FFFF B0E2$_H$	INTRCAN1ERR	CAN1 error interrupt	RCFDC0	Level	1071$_H$	✓	✓	✓	✓	✓			+1C4$_H$
114	ICRCAN1REC	FFFF B0E4$_H$	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1072$_H$	✓	✓	✓	✓	✓			+1C8$_H$
115	ICRCAN1TRX	FFFF B0E6$_H$	INTRCAN1TRX	CAN1 transmit interrupt	RCFDC0	Level	1073$_H$	✓	✓	✓	✓	✓			+1CC$_H$
116	ICCSIH1IC	FFFF B0E8$_H$	INTCSIH1IC*6	CSIH1 communication status interrupt	CSIH1	Edge	1074$_H$	✓	✓	✓	✓	✓			+1D0$_H$
	ICTAPA0IPEK0_2		INTTAPA0IPEK0_2*6	TAPA0 peak interrupt 0	TAPA0	Edge									
117	ICCSIH1IR	FFFF B0EA$_H$	INTCSIH1IR*6	CSIH1 receive status interrupt	CSIH1	Edge	1075$_H$	✓	✓	✓	✓	✓			+1D4$_H$
	ICTAPA0IVLY0_2		INTTAPA0IVLY0_2*6	TAPA0 valley interrupt 0	TAPA0	Edge									
118	ICCSIH1IRE	FFFF B0EC$_H$	INTCSIH1IRE*6	CSIH1 communication error interrupt	CSIH1	Edge	1076$_H$	✓	✓	✓	✓	✓			+1D8$_H$
	ICCSIG0IC_2		INTCSIG0IC_2*6	CSIG0 communication status interrupt	CSIG0	Edge									
119	ICCSIH1IJC	FFFF B0EE$_H$	INTCSIH1IJC*6	CSIH1 job completion interrupt	CSIH1	Edge	1077$_H$	✓	✓	✓	✓	✓			+1DC$_H$
	ICCSIG0IR_2		INTCSIG0IR_2*6	CSIG0 receive status interrupt	CSIG0	Edge									
120	ICRLIN31	FFFF B0F0$_H$	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1078$_H$	✓	✓	✓	✓	✓			+1E0$_H$
121	ICRLIN31UR0	FFFF B0F2$_H$	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1079$_H$	✓	✓	✓	✓	✓			+1E4$_H$
122	ICRLIN31UR1	FFFF B0F4$_H$	INTRLIN31UR1	RLIN31 receive complete interrupt	RLIN31	Edge	107A$_H$	✓	✓	✓	✓	✓			+1E8$_H$
123	ICRLIN31UR2	FFFF B0F6$_H$	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	107B$_H$	✓	✓	✓	✓	✓			+1EC$_H$
124	Reserved	FFFF B0F8$_H$					107C$_H$	—	—	—	—	—			+1F0$_H$
125	Reserved	FFFF B0FA$_H$					107D$_H$	—	—	—	—	—			+1F4$_H$
126	Reserved	FFFF B0FC$_H$					107E$_H$	—	—	—	—	—			+1F8$_H$
127	Reserved	FFFF B0FE$_H$					107F$_H$	—	—	—	—	—			+1FC$_H$
128	ICP6	FFFF B100$_H$	INTP6	External interrupt	Port	Edge	1080$_H$	✓	✓	✓	✓	✓			+200$_H$
129	ICP7	FFFF B102$_H$	INTP7	External interrupt	Port	Edge	1081$_H$	✓	✓	✓	✓	✓			+204$_H$
130	ICP8	FFFF B104$_H$	INTP8	External interrupt	Port	Edge	1082$_H$	✓	✓	✓	✓	✓			+208$_H$
131	ICP12	FFFF B106$_H$	INTP12	External interrupt	Port	Edge	1083$_H$	✓	✓	✓	✓	✓			+20C$_H$
132	ICCSIH2IC	FFFF B108$_H$	INTCSIH2IC*6	CSIH2 communication status interrupt	CSIH2	Edge	1084$_H$	✓	✓	✓	✓	✓			+210$_H$
	ICTAUD0I0_2		INTTAUD0I0_2*6	Interrupt for TAUD0 channel 0	TAUD0	Edge									
133	ICCSIH2IR	FFFF B10A$_H$	INTCSIH2IR*6	CSIH2 receive status interrupt	CSIH2	Edge	1085$_H$	✓	✓	✓	✓	✓			+214$_H$
	ICTAUD0I4_2		INTTAUD0I4_2*6	Interrupt for TAUD0 channel 4	TAUD0	Edge									
134	ICCSIH2IRE	FFFF B10C$_H$	INTCSIH2IRE*6	CSIH2 communication error interrupt	CSIH2	Edge	1086$_H$	✓	✓	✓	✓	✓			+218$_H$
	ICTAUD0I6_2		INTTAUD0I6_2*6	Interrupt for TAUD0 channel 6	TAUD0	Edge									
135	ICCSIH2IJC	FFFF B10E$_H$	INTCSIH2IJC*6	CSIH2 job completion interrupt	CSIH2	Edge	1087$_H$	✓	✓	✓	✓	✓			+21C$_H$
	ICTAUD0I8_2		INTTAUD0I8_2*6	Interrupt for TAUD0 channel 8	TAUD0	Edge									
136	Reserved	FFFF B110$_H$					1088$_H$	—	—	—	—	—			+220$_H$
137	Reserved	FFFF B112$_H$					1089$_H$	—	—	—	—	—			+224$_H$
138	Reserved	FFFF B114$_H$					108A$_H$	—	—	—	—	—			+228$_H$
139	Reserved	FFFF B116$_H$					108B$_H$	—	—	—	—	—			+22C$_H$
140	Reserved	FFFF B118$_H$					108C$_H$	—	—	—	—	—			+230$_H$
141	Reserved	FFFF B11A$_H$					108D$_H$	—	—	—	—	—			+234$_H$
142	ICTAUB0I0	FFFF B11C$_H$	INTTAUB0I0	Interrupt for TAUB0 channel 0	TAUB0	Edge	108E$_H$	✓	✓	✓	✓	✓			+238$_H$
143	ICTAUB0I1	FFFF B11E$_H$	INTTAUB0I1	Interrupt for TAUB0 channel 1	TAUB0	Edge	108F$_H$	✓	✓	✓	✓	✓			+23C$_H$
144	ICTAUB0I2	FFFF B120$_H$	INTTAUB0I2	Interrupt for TAUB0 channel 2	TAUB0	Edge	1090$_H$	✓	✓	✓	✓	✓			+240$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register					Detection Type[*2]							Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
145	ICTAUB0I3	FFFF B122$_H$	INTTAUB0I3	Interrupt for TAUB0 channel 3	TAUB0	Edge	1091$_H$	✓	✓	✓	✓	✓	[*3]	[*4]	+244$_H$
146	ICTAUB0I4	FFFF B124$_H$	INTTAUB0I4	Interrupt for TAUB0 channel 4	TAUB0	Edge	1092$_H$	✓	✓	✓	✓	✓			+248$_H$
147	ICTAUB0I5	FFFF B126$_H$	INTTAUB0I5	Interrupt for TAUB0 channel 5	TAUB0	Edge	1093$_H$	✓	✓	✓	✓	✓			+24C$_H$
148	ICTAUB0I6	FFFF B128$_H$	INTTAUB0I6	Interrupt for TAUB0 channel 6	TAUB0	Edge	1094$_H$	✓	✓	✓	✓	✓			+250$_H$
149	ICTAUB0I7	FFFF B12A$_H$	INTTAUB0I7	Interrupt for TAUB0 channel 7	TAUB0	Edge	1095$_H$	✓	✓	✓	✓	✓			+254$_H$
150	ICTAUB0I8	FFFF B12C$_H$	INTTAUB0I8	Interrupt for TAUB0 channel 8	TAUB0	Edge	1096$_H$	✓	✓	✓	✓	✓			+258$_H$
151	ICTAUB0I9	FFFF B12E$_H$	INTTAUB0I9	Interrupt for TAUB0 channel 9	TAUB0	Edge	1097$_H$	✓	✓	✓	✓	✓			+25C$_H$
152	ICTAUB0I10	FFFF B130$_H$	INTTAUB0I10	Interrupt for TAUB0 channel 10	TAUB0	Edge	1098$_H$	✓	✓	✓	✓	✓			+260$_H$
153	ICTAUB0I11	FFFF B132$_H$	INTTAUB0I11	Interrupt for TAUB0 channel 11	TAUB0	Edge	1099$_H$	✓	✓	✓	✓	✓			+264$_H$
154	ICTAUB0I12	FFFF B134$_H$	INTTAUB0I12	Interrupt for TAUB0 channel 12	TAUB0	Edge	109A$_H$	✓	✓	✓	✓	✓			+268$_H$
155	ICTAUB0I13	FFFF B136$_H$	INTTAUB0I13	Interrupt for TAUB0 channel 13	TAUB0	Edge	109B$_H$	✓	✓	✓	✓	✓			+26C$_H$
156	ICTAUB0I14	FFFF B138$_H$	INTTAUB0I14	Interrupt for TAUB0 channel 14	TAUB0	Edge	109C$_H$	✓	✓	✓	✓	✓			+270$_H$
157	ICTAUB0I15	FFFF B13A$_H$	INTTAUB0I15	Interrupt for TAUB0 channel 15	TAUB0	Edge	109D$_H$	✓	✓	✓	✓	✓			+274$_H$
158	ICCSIH3IC	FFFF B13C$_H$	INTCSIH3IC[*6]	CSIH3 communication status interrupt	CSIH3	Edge	109E$_H$	✓	✓	✓	✓	✓			+278$_H$
	ICTAUD0I2_2		INTTAUD0I2_2[*6]	Interrupt for TAUD0 channel 2	TAUD0	Edge									
159	ICCSIH3IR	FFFF B13E$_H$	INTCSIH3IR[*6]	CSIH3 receive status interrupt	CSIH3	Edge	109F$_H$	✓	✓	✓	✓	✓			+27C$_H$
	ICTAUD0I10_2		INTTAUD0I10_2[*6]	Interrupt for TAUD0 channel 10	TAUD0	Edge									
160	ICCSIH3IRE	FFFF B140$_H$	INTCSIH3IRE[*6]	CSIH3 communication error interrupt	CSIH3	Edge	10A0$_H$	✓	✓	✓	✓	✓			+280$_H$
	ICTAUD0I12_2		INTTAUD0I12_2[*6]	Interrupt for TAUD0 channel 12	TAUD0	Edge									
161	ICCSIH3IJC	FFFF B142$_H$	INTCSIH3IJC[*6]	CSIH3 job completion interrupt	CSIH3	Edge	10A1$_H$	✓	✓	✓	✓	✓			+284$_H$
	ICTAUD0I14_2		INTTAUD0I14_2[*6]	Interrupt for TAUD0 channel 14	TAUD0	Edge									
162	ICRLIN22	FFFF B144$_H$	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	10A2$_H$	✓	✓	✓	✓	✓			+288$_H$
163	ICRLIN23	FFFF B146$_H$	INTRLIN23	RLIN23 interrupt	RLIN240	Edge	10A3$_H$	—	✓	✓	✓	✓			+28C$_H$
164	ICRLIN32	FFFF B148$_H$	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	10A4$_H$	✓	✓	✓	✓	✓			+290$_H$
165	ICRLIN32UR0	FFFF B14A$_H$	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	10A5$_H$	✓	✓	✓	✓	✓			+294$_H$
166	ICRLIN32UR1	FFFF B14C$_H$	INTRLIN32UR1	RLIN32 receive complete interrupt	RLIN32	Edge	10A6$_H$	✓	✓	✓	✓	✓			+298$_H$
167	ICRLIN32UR2	FFFF B14E$_H$	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	10A7$_H$	✓	✓	✓	✓	✓			+29C$_H$
168	ICTAUJ1I0	FFFF B150$_H$	INTTAUJ1I0	Interrupt for TAUJ1 channel 0	TAUJ1	Edge	10A8$_H$	✓	✓	✓	✓	✓			+2A0$_H$
169	ICTAUJ1I1	FFFF B152$_H$	INTTAUJ1I1	Interrupt for TAUJ1 channel 1	TAUJ1	Edge	10A9$_H$	✓	✓	✓	✓	✓			+2A4$_H$
170	ICTAUJ1I2	FFFF B154$_H$	INTTAUJ1I2	Interrupt for TAUJ1 channel 2	TAUJ1	Edge	10AA$_H$	✓	✓	✓	✓	✓			+2A8$_H$
171	ICTAUJ1I3	FFFF B156$_H$	INTTAUJ1I3	Interrupt for TAUJ1 channel 3	TAUJ1	Edge	10AB$_H$	✓	✓	✓	✓	✓			+2AC$_H$
172	Reserved	FFFF B158$_H$					10AC$_H$	—	—	—	—	—			+2B0$_H$
173	ICFLXA0FDA	FFFF B15A$_H$	INTFLXA0FDA	FIFO transfer interrupt	FLXA0	Level	10AD$_H$	✓	✓	✓	✓	✓			+2B4$_H$
174	ICFLXA0FW	FFFF B15C$_H$	INTFLXA0FW	FIFO transfer warning interrupt	FLXA0	Level	10AE$_H$	✓	✓	✓	✓	✓			+2B8$_H$
175	ICFLXA0IQE	FFFF B15E$_H$	INTFLXA0IQE	Input queue empty interrupt	FLXA0	Level	10AF$_H$	✓	✓	✓	✓	✓			+2BC$_H$
176	ICFLXA0IQF	FFFF B160$_H$	INTFLXA0IQF	Input queue full interrupt	FLXA0	Level	10B0$_H$	✓	✓	✓	✓	✓			+2C0$_H$
177	ICFLXA0OT	FFFF B162$_H$	INTFLXA0OT	Output transfer end interrupt	FLXA0	Level	10B1$_H$	✓	✓	✓	✓	✓			+2C4$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register												Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							RINT = 0	RINT = 1	
178	ICFLXA0OW	FFFF B164$_H$	INTFLXA0OW	Output transfer warning interrupt	FLXA0	Level	10B2$_H$	✓	✓	✓	✓	✓	*3	*4	+2C8$_H$
179	ICFLXA0LINE0	FFFF B166$_H$	INTFLXA0LINE0	FlexRay0 interrupt	FLXA0	Level	10B3$_H$	✓	✓	✓	✓	✓			+2CC$_H$
180	ICFLXA0LINE1	FFFF B168$_H$	INTFLXA0LINE1	FlexRay1 interrupt	FLXA0	Level	10B4$_H$	✓	✓	✓	✓	✓			+2D0$_H$
181	ICFLXA0TIM0	FFFF B16A$_H$	INTFLXA0TIM0	Timer 0 interrupt	FLXA0	Level	10B5$_H$	✓	✓	✓	✓	✓			+2D4$_H$
182	ICFLXA0TIM1	FFFF B16C$_H$	INTFLXA0TIM1	Timer 1 interrupt	FLXA0	Level	10B6$_H$	✓	✓	✓	✓	✓			+2D8$_H$
183	ICFLXA0TIM2	FFFF B16E$_H$	INTFLXA0TIM2	Timer 2 interrupt	FLXA0	Level	10B7$_H$	✓	✓	✓	✓	✓			+2DC$_H$
184	Reserved	FFFF B170$_H$					10B8$_H$	—	—	—	—	—			+2E0$_H$
185	Reserved	FFFF B172$_H$					10B9$_H$	—	—	—	—	—			+2E4$_H$
186	Reserved	FFFF B174$_H$					10BA$_H$	—	—	—	—	—			+2E8$_H$
187	Reserved	FFFF B176$_H$					10BB$_H$	—	—	—	—	—			+2EC$_H$
188	Reserved	FFFF B178$_H$					10BC$_H$	—	—	—	—	—			+2F0$_H$
189	Reserved	FFFF B17A$_H$					10BD$_H$	—	—	—	—	—			+2F4$_H$
190	Reserved	FFFF B17C$_H$					10BE$_H$	—	—	—	—	—			+2F8$_H$
191	Reserved	FFFF B17E$_H$					10BF$_H$	—	—	—	—	—			+2FC$_H$
192	Reserved	FFFF B180$_H$					10C0$_H$	—	—	—	—	—			+300$_H$
193	Reserved	FFFF B182$_H$					10C1$_H$	—	—	—	—	—			+304$_H$
194	Reserved	FFFF B184$_H$					10C2$_H$	—	—	—	—	—			+308$_H$
195	Reserved	FFFF B186$_H$					10C3$_H$	—	—	—	—	—			+30C$_H$
196	Reserved	FFFF B188$_H$					10C4$_H$	—	—	—	—	—			+310$_H$
197	Reserved	FFFF B18A$_H$					10C5$_H$	—	—	—	—	—			+314$_H$
198	Reserved	FFFF B18C$_H$					10C6$_H$	—	—	—	—	—			+318$_H$
199	Reserved	FFFF B18E$_H$					10C7$_H$	—	—	—	—	—			+31C$_H$
200	Reserved	FFFF B190$_H$					10C8$_H$	—	—	—	—	—			+320$_H$
201	Reserved	FFFF B192$_H$					10C9$_H$	—	—	—	—	—			+324$_H$
202	Reserved	FFFF B194$_H$					10CA$_H$	—	—	—	—	—			+328$_H$
203	Reserved	FFFF B196$_H$					10CB$_H$	—	—	—	—	—			+32C$_H$
204	Reserved	FFFF B198$_H$					10CC$_H$	—	—	—	—	—			+330$_H$
205	ICP9	FFFF B19A$_H$	INTP9	External interrupt	Port	Edge	10CD$_H$	✓	✓	✓	✓	✓			+334$_H$
206	ICP13	FFFF B19C$_H$	INTP13	External interrupt	Port	Edge	10CE$_H$	✓	✓	✓	✓	✓			+338$_H$
207	ICP14	FFFF B19E$_H$	INTP14	External interrupt	Port	Edge	10CF$_H$	—	✓	✓	✓	✓			+33C$_H$
208	ICP15	FFFF B1A0$_H$	INTP15	External interrupt	Port	Edge	10D0$_H$	—	✓	✓	✓	✓			+340$_H$
209	ICRTCA01S	FFFF B1A2$_H$	INTRTCA01S	RTCA0 1-second interval interrupt	RTCA0	Edge	10D1$_H$	✓	✓	✓	✓	✓			+344$_H$
210	ICRTCA0AL	FFFF B1A4$_H$	INTRTCA0AL	RTCA0 alarm interrupt	RTCA0	Edge	10D2$_H$	✓	✓	✓	✓	✓			+348$_H$
211	ICRTCA0R	FFFF B1A6$_H$	INTRTCA0R	RTCA0 fixed interval interrupt	RTCA0	Edge	10D3$_H$	✓	✓	✓	✓	✓			+34C$_H$
212	ICADCA1ERR	FFFF B1A8$_H$	INTADCA1ERR	ADCA1 error interrupt	ADCA1	Edge	10D4$_H$	—	✓	✓	✓	✓			+350$_H$
213	ICADCA1I0	FFFF B1AA$_H$	INTADCA1I0	ADCA1 scan group 1 (SG1) end interrupt	ADCA1	Edge	10D5$_H$	—	✓	✓	✓	✓			+354$_H$
214	ICADCA1I1	FFFF B1AC$_H$	INTADCA1I1	ADCA1 scan group 2 (SG2) end interrupt	ADCA1	Edge	10D6$_H$	—	✓	✓	✓	✓			+358$_H$
215	ICADCA1I2	FFFF B1AE$_H$	INTADCA1I2	ADCA1 scan group 3 (SG3) end interrupt	ADCA1	Edge	10D7$_H$	—	✓	✓	✓	✓			+35C$_H$
216	Reserved	FFFF B1B0$_H$					10D8$_H$	—	—	—	—	—			+360$_H$
217	ICRCAN2ERR	FFFF B1B2$_H$	INTRCAN2ERR	CAN2 error interrupt	RCFDC0	Level	10D9$_H$	✓	✓	✓	✓	✓			+364$_H$
218	ICRCAN2REC	FFFF B1B4$_H$	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DA$_H$	✓	✓	✓	✓	✓			+368$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register												Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							RINT = 0	RINT = 1	
219	ICRCAN2TRX	FFFF B1B6$_H$	INTRCAN2TRX	CAN2 transmit interrupt	RCFDC0	Level	10DB$_H$	✓	✓	✓	✓	✓	*3	*4	+36C$_H$
220	ICRCAN3ERR	FFFF B1B8$_H$	INTRCAN3ERR	CAN3 error interrupt	RCFDC0	Level	10DC$_H$	✓	✓	✓	✓	✓			+370$_H$
221	ICRCAN3REC	FFFF B1BA$_H$	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DD$_H$	✓	✓	✓	✓	✓			+374$_H$
222	ICRCAN3TRX	FFFF B1BC$_H$	INTRCAN3TRX	CAN3 transmit interrupt	RCFDC0	Level	10DE$_H$	✓	✓	✓	✓	✓			+378$_H$
223	ICCSIG1IC	FFFF B1BE$_H$	INTCSIG1IC	CSIG1 communication status interrupt	CSIG1	Edge	10DF$_H$	—	✓	✓	✓	✓			+37C$_H$
224	ICCSIG1IR	FFFF B1C0$_H$	INTCSIG1IR	CSIG1 receive status interrupt	CSIG1	Edge	10E0$_H$	—	✓	✓	✓	✓			+380$_H$
225	ICCSIG1IRE	FFFF B1C2$_H$	INTCSIG1IRE	CSIG1 communication error interrupt	CSIG1	Edge	10E1$_H$	—	✓	✓	✓	✓			+384$_H$
226	ICRLIN24	FFFF B1C4$_H$	INTRLIN24	RLIN24 interrupt	RLIN241	Edge	10E2$_H$	—	✓	✓	✓	✓			+388$_H$
227	ICRLIN25	FFFF B1C6$_H$	INTRLIN25	RLIN25 interrupt	RLIN241	Edge	10E3$_H$	—	✓	✓	✓	✓			+38C$_H$
228	ICRLIN33	FFFF B1C8$_H$	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10E4$_H$	—	✓	✓	✓	✓			+390$_H$
229	ICRLIN33UR0	FFFF B1CA$_H$	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10E5$_H$	—	✓	✓	✓	✓			+394$_H$
230	ICRLIN33UR1	FFFF B1CC$_H$	INTRLIN33UR1	RLIN33 receive complete interrupt	RLIN33	Edge	10E6$_H$	—	✓	✓	✓	✓			+398$_H$
231	ICRLIN33UR2	FFFF B1CE$_H$	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10E7$_H$	—	✓	✓	✓	✓			+39C$_H$
232	ICRLIN34	FFFF B1D0$_H$	INTRLIN34	RLIN34 interrupt	RLIN34	Edge	10E8$_H$	—	✓	✓	✓	✓			+3A0$_H$
233	ICRLIN34UR0	FFFF B1D2$_H$	INTRLIN34UR0	RLIN34 transmit interrupt	RLIN34	Edge	10E9$_H$	—	✓	✓	✓	✓			+3A4$_H$
234	ICRLIN34UR1	FFFF B1D4$_H$	INTRLIN34UR1	RLIN34 receive complete interrupt	RLIN34	Edge	10EA$_H$	—	✓	✓	✓	✓			+3A8$_H$
235	ICRLIN34UR2	FFFF B1D6$_H$	INTRLIN34UR2	RLIN34 status interrupt	RLIN34	Edge	10EB$_H$	—	✓	✓	✓	✓			+3AC$_H$
236	ICRLIN35	FFFF B1D8$_H$	INTRLIN35	RLIN35 interrupt	RLIN35	Edge	10EC$_H$	—	✓	✓	✓	✓			+3B0$_H$
237	ICRLIN35UR0	FFFF B1DA$_H$	INTRLIN35UR0	RLIN35 transmit interrupt	RLIN35	Edge	10ED$_H$	—	✓	✓	✓	✓			+3B4$_H$
238	ICRLIN35UR1	FFFF B1DC$_H$	INTRLIN35UR1	RLIN35 receive complete interrupt	RLIN35	Edge	10EE$_H$	—	✓	✓	✓	✓			+3B8$_H$
239	ICRLIN35UR2	FFFF B1DE$_H$	INTRLIN35UR2	RLIN35 status interrupt	RLIN35	Edge	10EF$_H$	—	✓	✓	✓	✓			+3BC$_H$
240	ICRIIC1TI	FFFF B1E0$_H$	INTRIIC1TI	RIIC1 transmit data empty interrupt	RIIC1	Edge	10F0$_H$	✓	✓	✓	✓	✓			+3C0$_H$
241	ICRIIC1EE	FFFF B1E2$_H$	INTRIIC1EE	RIIC1 receive error/event interrupt	RIIC1	Level	10F1$_H$	✓	✓	✓	✓	✓			+3C4$_H$
242	ICRIIC1RI	FFFF B1E4$_H$	INTRIIC1RI	RIIC1 receive complete interrupt	RIIC1	Edge	10F2$_H$	✓	✓	✓	✓	✓			+3C8$_H$
243	ICRIIC1TEI	FFFF B1E6$_H$	INTRIIC1TEI	RIIC1 transmit complete interrupt	RIIC1	Level	10F3$_H$	✓	✓	✓	✓	✓			+3CC$_H$
244	Reserved	FFFF B1E8$_H$					10F4$_H$	—	—	—	—	—			+3D0$_H$
245	Reserved	FFFF B1EA$_H$					10F5$_H$	—	—	—	—	—			+3D4$_H$
246	Reserved	FFFF B1EC$_H$					10F6$_H$	—	—	—	—	—			+3D8$_H$
247	Reserved	FFFF B1EE$_H$					10F7$_H$	—	—	—	—	—			+3DC$_H$
248	Reserved	FFFF B1F0$_H$					10F8$_H$	—	—	—	—	—			+3E0$_H$
249	Reserved	FFFF B1F2$_H$					10F9$_H$	—	—	—	—	—			+3E4$_H$
250	Reserved	FFFF B1F4$_H$					10FA$_H$	—	—	—	—	—			+3E8$_H$
251	Reserved	FFFF B1F6$_H$					10FB$_H$	—	—	—	—	—			+3EC$_H$
252	Reserved	FFFF B1F8$_H$					10FC$_H$	—	—	—	—	—			+3F0$_H$
253	Reserved	FFFF B1FA$_H$					10FD$_H$	—	—	—	—	—			+3F4$_H$
254	Reserved	FFFF B1FC$_H$					10FE$_H$	—	—	—	—	—			+3F8$_H$
255	Reserved	FFFF B1FE$_H$					10FF$_H$	—	—	—	—	—			+3FC$_H$
256	ICTAUB1I0	FFFF B200$_H$	INTTAUB1I0	Interrupt for TAUB1 channel 0	TAUB1	Edge	1100$_H$	—	—	✓	✓	✓			+400$_H$
257	ICTAUB1I1	FFFF B202$_H$	INTTAUB1I1	Interrupt for TAUB1 channel 1	TAUB1	Edge	1101$_H$	—	—	✓	✓	✓			+404$_H$
258	ICTAUB1I2	FFFF B204$_H$	INTTAUB1I2	Interrupt for TAUB1 channel 2	TAUB1	Edge	1102$_H$	—	—	✓	✓	✓			+408$_H$
259	ICTAUB1I3	FFFF B206$_H$	INTTAUB1I3	Interrupt for TAUB1 channel 3	TAUB1	Edge	1103$_H$	—	—	✓	✓	✓			+40C$_H$
260	ICTAUB1I4	FFFF B208$_H$	INTTAUB1I4	Interrupt for TAUB1 channel 4	TAUB1	Edge	1104$_H$	—	—	✓	✓	✓			+410$_H$
261	ICTAUB1I5	FFFF B20A$_H$	INTTAUB1I5	Interrupt for TAUB1 channel 5	TAUB1	Edge	1105$_H$	—	—	✓	✓	✓			+414$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2							Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
262	ICTAUB1I6	FFFF B20C$_H$	INTTAUB1I6	Interrupt for TAUB1 channel 6	TAUB1	Edge	1106$_H$	—	—	✓	✓	✓	*3	*4	+418$_H$
263	ICTAUB1I7	FFFF B20E$_H$	INTTAUB1I7	Interrupt for TAUB1 channel 7	TAUB1	Edge	1107$_H$	—	—	✓	✓	✓			+41C$_H$
264	ICTAUB1I8	FFFF B210$_H$	INTTAUB1I8	Interrupt for TAUB1 channel 8	TAUB1	Edge	1108$_H$	—	—	✓	✓	✓			+420$_H$
265	ICTAUB1I9	FFFF B212$_H$	INTTAUB1I9	Interrupt for TAUB1 channel 9	TAUB1	Edge	1109$_H$	—	—	✓	✓	✓			+424$_H$
266	ICTAUB1I10	FFFF B214$_H$	INTTAUB1I10	Interrupt for TAUB1 channel 10	TAUB1	Edge	110A$_H$	—	—	✓	✓	✓			+428$_H$
267	ICTAUB1I11	FFFF B216$_H$	INTTAUB1I11	Interrupt for TAUB1 channel 11	TAUB1	Edge	110B$_H$	—	—	✓	✓	✓			+42C$_H$
268	ICTAUB1I12	FFFF B218$_H$	INTTAUB1I12	Interrupt for TAUB1 channel 12	TAUB1	Edge	110C$_H$	—	—	✓	✓	✓			+430$_H$
269	ICTAUB1I13	FFFF B21A$_H$	INTTAUB1I13	Interrupt for TAUB1 channel 13	TAUB1	Edge	110D$_H$	—	—	✓	✓	✓			+434$_H$
270	ICTAUB1I14	FFFF B21C$_H$	INTTAUB1I14	Interrupt for TAUB1 channel 14	TAUB1	Edge	110E$_H$	—	—	✓	✓	✓			+438$_H$
271	ICTAUB1I15	FFFF B21E$_H$	INTTAUB1I15	Interrupt for TAUB1 channel 15	TAUB1	Edge	110F$_H$	—	—	✓	✓	✓			+43C$_H$
272	ICRCAN4ERR	FFFF B220$_H$	INTRCAN4ERR	CAN4 error interrupt	RCFDC0	Level	1110$_H$	✓	✓	✓	✓	✓			+440$_H$
273	ICRCAN4REC	FFFF B222$_H$	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1111$_H$	✓	✓	✓	✓	✓			+444$_H$
274	ICRCAN4TRX	FFFF B224$_H$	INTRCAN4TRX	CAN4 transmit interrupt	RCFDC0	Level	1112$_H$	✓	✓	✓	✓	✓			+448$_H$
275	ICRLIN26	FFFF B226$_H$	INTRLIN26	RLIN26 interrupt	RLIN241	Edge	1113$_H$	—	—	✓	✓	✓			+44C$_H$
276	ICRLIN27	FFFF B228$_H$	INTRLIN27	RLIN27 interrupt	RLIN241	Edge	1114$_H$	—	—	✓	✓	✓			+450$_H$
277	ICTAUJ2I0	FFFF B22A$_H$	INTTAUJ2I0	Interrupt for TAUJ2 channel 0	TAUJ2	Edge	1115$_H$	✓	✓	✓	✓	✓			+454$_H$
278	ICTAUJ2I1	FFFF B22C$_H$	INTTAUJ2I1	Interrupt for TAUJ2 channel 1	TAUJ2	Edge	1116$_H$	✓	✓	✓	✓	✓			+458$_H$
279	ICTAUJ2I2	FFFF B22E$_H$	INTTAUJ2I2	Interrupt for TAUJ2 channel 2	TAUJ2	Edge	1117$_H$	✓	✓	✓	✓	✓			+45C$_H$
280	ICTAUJ2I3	FFFF B230$_H$	INTTAUJ2I3	Interrupt for TAUJ2 channel 3	TAUJ2	Edge	1118$_H$	✓	✓	✓	✓	✓			+460$_H$
281	ICTAUJ3I0	FFFF B232$_H$	INTTAUJ3I0	Interrupt for TAUJ3 channel 0	TAUJ3	Edge	1119$_H$	✓	✓	✓	✓	✓			+464$_H$
282	ICTAUJ3I1	FFFF B234$_H$	INTTAUJ3I1	Interrupt for TAUJ3 channel 1	TAUJ3	Edge	111A$_H$	✓	✓	✓	✓	✓			+468$_H$
283	ICTAUJ3I2	FFFF B236$_H$	INTTAUJ3I2	Interrupt for TAUJ3 channel 2	TAUJ3	Edge	111B$_H$	✓	✓	✓	✓	✓			+46C$_H$
284	ICTAUJ3I3	FFFF B238$_H$	INTTAUJ3I3	Interrupt for TAUJ3 channel 3	TAUJ3	Edge	111C$_H$	✓	✓	✓	✓	✓			+470$_H$
285	ICRLIN28	FFFF B23A$_H$	INTRLIN28	RLIN28 interrupt	RLIN242	Edge	111D$_H$	—	—	✓	✓	✓			+474$_H$
286	ICRLIN29	FFFF B23C$_H$	INTRLIN29	RLIN29 interrupt	RLIN242	Edge	111E$_H$	—	—	✓	✓	✓			+478$_H$
287	ICRCAN5ERR	FFFF B23E$_H$	INTRCAN5ERR	CAN5 error interrupt	RCFDC0	Level	111F$_H$	✓	✓	✓	✓	✓			+47C$_H$
288	ICRCAN5REC	FFFF B240$_H$	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1120$_H$	✓	✓	✓	✓	✓			+480$_H$
289	ICRCAN5TRX	FFFF B242$_H$	INTRCAN5TRX	CAN5 transmit interrupt	RCFDC0	Level	1121$_H$	✓	✓	✓	✓	✓			+484$_H$
290	Reserved	FFFF B244$_H$					1122$_H$	—	—	—	—	—			+488$_H$
291	Reserved	FFFF B246$_H$					1123$_H$	—	—	—	—	—			+48C$_H$
292	Reserved	FFFF B248$_H$					1124$_H$	—	—	—	—	—			+490$_H$
293	Reserved	FFFF B24A$_H$					1125$_H$	—	—	—	—	—			+494$_H$
294	Reserved	FFFF B24C$_H$					1126$_H$	—	—	—	—	—			+498$_H$
295	Reserved	FFFF B24E$_H$					1127$_H$	—	—	—	—	—			+49C$_H$
296	Reserved	FFFF B250$_H$					1128$_H$	—	—	—	—	—			+4A0$_H$
297	Reserved	FFFF B252$_H$					1129$_H$	—	—	—	—	—			+4A4$_H$
298	ICDMA16	FFFF B254$_H$	INTDMA16	DMA20 transfer completion	PDMA0	Edge	112A$_H$	✓	✓	✓	✓	✓			+4A8$_H$
299	ICDMA17	FFFF B256$_H$	INTDMA17	DMA21 transfer completion	PDMA0	Edge	112B$_H$	✓	✓	✓	✓	✓			+4AC$_H$
300	ICDMA18	FFFF B258$_H$	INTDMA18	DMA22 transfer completion	PDMA0	Edge	112C$_H$	✓	✓	✓	✓	✓			+4B0$_H$
301	ICDMA19	FFFF B25A$_H$	INTDMA19	DMA23 transfer completion	PDMA0	Edge	112D$_H$	✓	✓	✓	✓	✓			+4B4$_H$
302	ICDMA20	FFFF B25C$_H$	INTDMA20	DMA24 transfer completion	PDMA0	Edge	112E$_H$	✓	✓	✓	✓	✓			+4B8$_H$
303	ICDMA21	FFFF B25E$_H$	INTDMA21	DMA25 transfer completion	PDMA0	Edge	112F$_H$	✓	✓	✓	✓	✓			+4BC$_H$
304	ICDMA22	FFFF B260$_H$	INTDMA22	DMA26 transfer completion	PDMA0	Edge	1130$_H$	✓	✓	✓	✓	✓			+4C0$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register												Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							RINT = 0	RINT = 1	
305	ICDMA23	FFFF B262$_H$	INTDMA23	DMA27 transfer completion	PDMA0	Edge	1131$_H$	✓	✓	✓	✓	✓	*3	*4	+4C4$_H$
306	ICDMA24	FFFF B264$_H$	INTDMA24	DMA30 transfer completion	PDMA0	Edge	1132$_H$	✓	✓	✓	✓	✓			+4C8$_H$
307	ICDMA25	FFFF B266$_H$	INTDMA25	DMA31 transfer completion	PDMA0	Edge	1133$_H$	✓	✓	✓	✓	✓			+4CC$_H$
308	ICDMA26	FFFF B268$_H$	INTDMA26	DMA32 transfer completion	PDMA0	Edge	1134$_H$	✓	✓	✓	✓	✓			+4D0$_H$
309	ICDMA27	FFFF B26A$_H$	INTDMA27	DMA33 transfer completion	PDMA0	Edge	1135$_H$	✓	✓	✓	✓	✓			+4D4$_H$
310	ICDMA28	FFFF B26C$_H$	INTDMA28	DMA34 transfer completion	PDMA0	Edge	1136$_H$	✓	✓	✓	✓	✓			+4D8$_H$
311	ICDMA29	FFFF B26E$_H$	INTDMA29	DMA35 transfer completion	PDMA0	Edge	1137$_H$	✓	✓	✓	✓	✓			+4DC$_H$
312	ICDMA30	FFFF B270$_H$	INTDMA30	DMA36 transfer completion	PDMA0	Edge	1138$_H$	✓	✓	✓	✓	✓			+4E0$_H$
313	ICDMA31	FFFF B272$_H$	INTDMA31	DMA37 transfer completion	PDMA0	Edge	1139$_H$	✓	✓	✓	✓	✓			+4E4$_H$
314	Reserved	FFFF B274$_H$					113A$_H$	—	—	—	—	—			+4E8$_H$
315	ICETNB0DATA	FFFF B276$_H$	INTETNB0DATA	Data related interrupt	ETNB0	Level	113B$_H$	—	—	✓	✓	✓			+4EC$_H$
316	ICETNB0ERR	FFFF B278$_H$	INTETNB0ERR	Error related interrupt	ETNB0	Level	113C$_H$	—	—	✓	✓	✓			+4F0$_H$
317	ICETNB0MNG	FFFF B27A$_H$	INTETNB0MNG	Management related interrupt	ETNB0	Level	113D$_H$	—	—	✓	✓	✓			+4F4$_H$
318	ICETNB0MAC	FFFF B27C$_H$	INTETNB0MAC	MAC interrupt	ETNB0	Level	113E$_H$	—	—	✓	✓	✓			+4F8$_H$
319	Reserved	FFFF B27E$_H$					113F$_H$	—	—	—	—	—			+4FC$_H$
320	Reserved	FFFF B280$_H$					1140$_H$	—	—	—	—	—			+500$_H$
321	ICRCAN6ERR	FFFF B282$_H$	INTRCAN6ERR	CAN6 error interrupt	RCFDC0	Level	1141$_H$	✓	✓	✓	✓	✓			+504$_H$
322	ICRCAN6REC	FFFF B284$_H$	INTRCAN6REC	CAN6 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1142$_H$	✓	✓	✓	✓	✓			+508$_H$
323	ICRCAN6TRX	FFFF B286$_H$	INTRCAN6TRX	CAN6 transmit interrupt	RCFDC0	Level	1143$_H$	✓	✓	✓	✓	✓			+50C$_H$
324	ICRLIN210	FFFF B288$_H$	INTRLIN210	RLIN210 interrupt	RLIN242	Edge	1144$_H$	—	—	—	✓	✓			+510$_H$
325	ICRLIN211	FFFF B28A$_H$	INTRLIN211	RLIN211 interrupt	RLIN242	Edge	1145$_H$	—	—	—	✓	✓			+514$_H$
326	ICCSIG2IC	FFFF B28C$_H$	INTCSIG2IC	CSIG2 communication status interrupt	CSIG2	Edge	1146$_H$	—	—	✓	✓	✓			+518$_H$
327	ICCSIG2IR	FFFF B28E$_H$	INTCSIG2IR	CSIG2 receive status interrupt	CSIG2	Edge	1147$_H$	—	—	✓	✓	✓			+51C$_H$
328	ICCSIG2IRE	FFFF B290$_H$	INTCSIG2IRE	CSIG2 communication error interrupt	CSIG2	Edge	1148$_H$	—	—	✓	✓	✓			+520$_H$
329	ICCSIG3IC	FFFF B292$_H$	INTCSIG3IC	CSIG3 communication status interrupt	CSIG3	Edge	1149$_H$	—	—	✓	✓	✓			+524$_H$
330	ICCSIG3IR	FFFF B294$_H$	INTCSIG3IR	CSIG3 receive status interrupt	CSIG3	Edge	114A$_H$	—	—	✓	✓	✓			+528$_H$
331	ICCSIG3IRE	FFFF B296$_H$	INTCSIG3IRE	CSIG3 communication error interrupt	CSIG3	Edge	114B$_H$	—	—	✓	✓	✓			+52C$_H$
332	ICRCAN7ERR	FFFF B298$_H$	INTRCAN7ERR	CAN7 error interrupt	RCFDC0	Level	114C$_H$	✓	✓	✓	✓	✓			+530$_H$
333	ICRCAN7REC	FFFF B29A$_H$	INTRCAN7REC	CAN7 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	114D$_H$	✓	✓	✓	✓	✓			+534$_H$
334	ICRCAN7TRX	FFFF B29C$_H$	INTRCAN7TRX	CAN7 transmit interrupt	RCFDC0	Level	114E$_H$	✓	✓	✓	✓	✓			+538$_H$
335	Reserved	FFFF B29E$_H$					114F$_H$	—	—	—	—	—			+53C$_H$
336	Reserved	FFFF B2A0$_H$					1150$_H$	—	—	—	—	—			+540$_H$
337	Reserved	FFFF B2A2$_H$					1151$_H$	—	—	—	—	—			+544$_H$
338	Reserved	FFFF B2A4$_H$					1152$_H$	—	—	—	—	—			+548$_H$
339	Reserved	FFFF B2A6$_H$					1153$_H$	—	—	—	—	—			+54C$_H$
340	Reserved	FFFF B2A8$_H$					1154$_H$	—	—	—	—	—			+550$_H$
341	Reserved	FFFF B2AA$_H$					1155$_H$	—	—	—	—	—			+554$_H$
342	Reserved	FFFF B2AC$_H$					1156$_H$	—	—	—	—	—			+558$_H$
343	Reserved	FFFF B2AE$_H$					1157$_H$	—	—	—	—	—			+55C$_H$
344	Reserved	FFFF B2B0$_H$					1158$_H$	—	—	—	—	—			+560$_H$
345	Reserved	FFFF B2B2$_H$					1159$_H$	—	—	—	—	—			+564$_H$
346	Reserved	FFFF B2B4$_H$					115A$_H$	—	—	—	—	—			+568$_H$
347	ICSENT0SI	FFFF B2B6$_H$	INTSENT0SI	Status interrupt for RSENT0	RSENT0	Level	115B$_H$	—	✓	✓	✓	✓			+56C$_H$
348	ICSENT0RI	FFFF B2B8$_H$	INTSENT0RI	Receive interrupt for RSENT0	RSENT0	Edge	115C$_H$	—	✓	✓	✓	✓			+570$_H$
349	ICSENT1SI	FFFF B2BA$_H$	INTSENT1SI	Status interrupt for RSENT1	RSENT1	Level	115D$_H$	✓	✓	✓	✓	✓			+574$_H$
350	ICSENT1RI	FFFF B2BC$_H$	INTSENT1RI	Receive interrupt for RSENT1	RSENT1	Edge	115E$_H$	✓	✓	✓	✓	✓			+578$_H$

Table 7BC.5 EI Level Maskable Interrupt Sources (RH850/F1KM-S4)

Interrupt			Interrupt Request				Exception Source Code	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register					Detection Type*2							Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit								RINT = 0	RINT = 1	
351	Reserved	FFFF B2BE$_H$					115F$_H$	—	—	—	—	—	*3	*4	+57C$_H$
352	Reserved	FFFF B2C0$_H$					1160$_H$	—	—	—	—	—			+580$_H$
353	Reserved	FFFF B2C2$_H$					1161$_H$	—	—	—	—	—			+584$_H$
354	Reserved	FFFF B2C4$_H$					1162$_H$	—	—	—	—	—			+588$_H$
355	Reserved	FFFF B2C6$_H$					1163$_H$	—	—	—	—	—			+58C$_H$
356	ICDPE	FFFF B2C8$_H$	INTDPE	LPS0 digital port error interrupt	LPS0	Level	1164$_H$	✓	✓	✓	✓	✓			+590$_H$
357	ICAPE	FFFF B2CA$_H$	INTAPE	LPS0 analog port error interrupt	LPS0	Level	1165$_H$	✓	✓	✓	✓	✓			+594$_H$
358	Reserved	FFFF B2CC$_H$					1166$_H$	—	—	—	—	—			+598$_H$
359	Reserved	FFFF B2CE$_H$					1167$_H$	—	—	—	—	—			+59C$_H$
360	ICRLIN36	FFFF B2D0$_H$	INTRLIN36	RLIN36 interrupt	RLIN36	Edge	1168$_H$	—	—	✓	✓	✓			+5A0$_H$
361	ICRLIN36UR0	FFFF B2D2$_H$	INTRLIN36UR0	RLIN36 transmit interrupt	RLIN36	Edge	1169$_H$	—	—	✓	✓	✓			+5A4$_H$
362	ICRLIN36UR1	FFFF B2D4$_H$	INTRLIN36UR1	RLIN36 receive complete interrupt	RLIN36	Edge	116A$_H$	—	—	✓	✓	✓			+5A8$_H$
363	ICRLIN36UR2	FFFF B2D6$_H$	INTRLIN36UR2	RLIN36 status interrupt	RLIN36	Edge	116B$_H$	—	—	✓	✓	✓			+5AC$_H$
364	ICRLIN37	FFFF B2D8$_H$	INTRLIN37	RLIN37 interrupt	RLIN37	Edge	116C$_H$	—	—	✓	✓	✓			+5B0$_H$
365	ICRLIN37UR0	FFFF B2DA$_H$	INTRLIN37UR0	RLIN37 transmit interrupt	RLIN37	Edge	116D$_H$	—	—	✓	✓	✓			+5B4$_H$
366	ICRLIN37UR1	FFFF B2DC$_H$	INTRLIN37UR1	RLIN37 receive complete interrupt	RLIN37	Edge	116E$_H$	—	—	✓	✓	✓			+5B8$_H$
367	ICRLIN37UR2	FFFF B2DE$_H$	INTRLIN37UR2	RLIN37 status interrupt	RLIN37	Edge	116F$_H$	—	—	✓	✓	✓			+5BC$_H$
368	ICP16	FFFF B2E0$_H$	INTP16	External interrupt	Port	Edge	1170$_H$	—	✓	✓	✓	✓			+5C0$_H$
369	ICP17	FFFF B2E2$_H$	INTP17	External interrupt	Port	Edge	1171$_H$	—	✓	✓	✓	✓			+5C4$_H$
370	ICP18	FFFF B2E4$_H$	INTP18	External interrupt	Port	Edge	1172$_H$	—	✓	✓	✓	✓			+5C8$_H$
371	ICP19	FFFF B2E6$_H$	INTP19	External interrupt	Port	Edge	1173$_H$	—	✓	✓	✓	✓			+5CC$_H$
372	ICP20	FFFF B2E8$_H$	INTP20	External interrupt	Port	Edge	1174$_H$	—	✓	✓	✓	✓			+5D0$_H$
373	ICP21	FFFF B2EA$_H$	INTP21	External interrupt	Port	Edge	1175$_H$	—	✓	✓	✓	✓			+5D4$_H$
374	ICP22	FFFF B2EC$_H$	INTP22	External interrupt	Port	Edge	1176$_H$	—	✓	✓	✓	✓			+5D8$_H$
375	ICP23	FFFF B2EE$_H$	INTP23	External interrupt	Port	Edge	1177$_H$	—	✓	✓	✓	✓			+5DC$_H$
376	ICGRZF	FFFF B2F0$_H$	INTGRZF	interrupt for GRZF	GRZF	Edge	1178$_H$	✓	✓	✓	✓	✓			+5E0$_H$

Note 1. Each interrupt is connected to INTC1 channel 8 to 31 and INTC2 channel 32 to 376.

Note 2. This indicates whether an interrupt source is detected at the level or edge. This also affects the value after reset of an EI level interrupt control register. For details, see **Section 7BC.4.2, ICxxx — EI Level Interrupt Control Registers**. For detection at level, an interrupt source is cleared by accessing to the register that retains an interrupt source. The procedure shown in **Section 3BC.3.1, Synchronization of Store Instruction Completion and Subsequent Instruction Execution** are required to reflect the result of the register update to the subsequent instruction.

Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100$_H$ to 1F0$_H$ according to the priority (0 to 15).

Note 4. Irrespective of the priority, offset addresses are uniformly +100$_H$.

Note 5. The table reference method uses a table for reading an exception handler address for each interrupt channel, and it extracts handler address by referencing that table. Table reference position is determined by the following formula.
Exception handler address read position = INTBP register + channel number × 4 bytes

Note 6. The same interrupt source is assigned to different interrupt channels. For details, see **Section 7BC.5.2, SELB_INTC1 — INTC1 Interrupt Select Register**.

Note 7. 32 interrupt sources are assigned to the same interrupt channel. For details, see **Section 37, PWM Output/Diagnostic (PWM-Diag)**.

Note 8. For details, see **Section 7BC.10, Exception Handler Address**.

Note 9. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it in both ways at the same time is not possible.
It is used as FEINT when OSTM0 functions as the TSU (timing supervision unit). It is used as EIINT when OSTM0 functions as anything other than the TSU function.

Note 10. For details on the interrupt source, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register					Detection Type[*2]						Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
0	Reserved	FFFE EA00$_H$					1000$_H$	—	—	—	—	[*3]	[*4]	+000$_H$
1	Reserved	FFFE EA02$_H$					1001$_H$	—	—	—	—			+004$_H$
2	Reserved	FFFE EA04$_H$					1002$_H$	—	—	—	—			+008$_H$
3	Reserved	FFFE EA06$_H$					1003$_H$	—	—	—	—			+00C$_H$
4	Reserved	FFFE EA08$_H$					1004$_H$	—	—	—	—			+010$_H$
5	Reserved	FFFE EA0A$_H$					1005$_H$	—	—	—	—			+014$_H$
6	Reserved	FFFE EA0C$_H$					1006$_H$	—	—	—	—			+018$_H$
7	Reserved	FFFE EA0E$_H$					1007$_H$	—	—	—	—			+01C$_H$
8	ICTAUD0I0	FFFE EA10$_H$	INTTAUD0I0[*6]	Interrupt for CH0 of TAUD0	TAUD0	Edge	1008$_H$	✓	✓	✓	✓			+020$_H$
	ICCSIH2IC_1		INTCSIH2IC_1[*6]	CSIH2 communication status interrupt	CSIH2	Edge		—	—	✓	✓			
9	ICTAUD0I2	FFFE EA12$_H$	INTTAUD0I2[*6]	Interrupt for CH2 of TAUD0	TAUD0	Edge	1009$_H$	✓	✓	✓	✓			+024$_H$
	ICCSIH3IC_1		INTCSIH3IC_1[*6]	CSIH3 communication status interrupt	CSIH3	Edge		—	—	—	✓			
10	ICTAUD0I4	FFFE EA14$_H$	INTTAUD0I4[*6]	Interrupt for CH4 of TAUD0	TAUD0	Edge	100A$_H$	✓	✓	✓	✓			+028$_H$
	ICCSIH2IR_1		INTCSIH2IR_1[*6]	CSIH2 receive status interrupt	CSIH2	Edge		—	—	✓	✓			
11	ICTAUD0I6	FFFE EA16$_H$	INTTAUD0I6[*6]	Interrupt for CH6 of TAUD0	TAUD0	Edge	100B$_H$	✓	✓	✓	✓			+02C$_H$
	ICCSIH2IRE_1		INTCSIH2IRE_1[*6]	CSIH2 communication error interrupt	CSIH2	Edge		—	—	✓	✓			
12	ICTAUD0I8	FFFE EA18$_H$	INTTAUD0I8[*6]	Interrupt for CH8 of TAUD0	TAUD0	Edge	100C$_H$	✓	✓	✓	✓			+030$_H$
	ICCSIH2IJC_1		INTCSIH2IJC_1[*6]	CSIH2 job completion interrupt	CSIH2	Edge		—	—	✓	✓			
13	ICTAUD0I10	FFFE EA1A$_H$	INTTAUD0I10[*6]	Interrupt for CH10 of TAUD0	TAUD0	Edge	100D$_H$	✓	✓	✓	✓			+034$_H$
	ICCSIH3IR_1		INTCSIH3IR_1[*6]	CSIH3 receive status interrupt	CSIH3	Edge		—	—	—	✓			
14	ICTAUD0I12	FFFE EA1C$_H$	INTTAUD0I12[*6]	Interrupt for CH12 of TAUD0	TAUD0	Edge	100E$_H$	✓	✓	✓	✓			+038$_H$
	ICCSIH3IRE_1		INTCSIH3IRE_1[*6]	CSIH3 communication error interrupt	CSIH3	Edge		—	—	—	✓			
15	ICTAUD0I14	FFFE EA1E$_H$	INTTAUD0I14[*6]	Interrupt for CH14 of TAUD0	TAUD0	Edge	100F$_H$	✓	✓	✓	✓			+03C$_H$
	ICCSIH3IJC_1		INTCSIH3IJC_1[*6]	CSIH3 job completion interrupt	CSIH3	Edge		—	—	—	✓			
16	ICTAPA0IPEK0	FFFE EA20$_H$	INTTAPA0IPEK0[*6]	TAPA0 peak interrupt 0	TAPA0	Edge	1010$_H$	✓	✓	✓	✓			+040$_H$
	ICCSIH1IC_1		INTCSIH1IC_1[*6]	CSIH1 communication status interrupt	CSIH1	Edge		—	—	✓	✓			
17	ICTAPA0IVLY0	FFFE EA22$_H$	INTTAPA0IVLY0[*6]	TAPA0 valley interrupt 0	TAPA0	Edge	1011$_H$	✓	✓	✓	✓			+044$_H$
	ICCSIH1IR_1		INTCSIH1IR_1[*6]	CSIH1 receive status interrupt	CSIH1	Edge		—	—	✓	✓			
18	ICADCA0I0	FFFE EA24$_H$	INTADCA0I0	ADCA0 SG1 end interrupt	ADCA0	Edge	1012$_H$	✓	✓	✓	✓			+048$_H$
19	ICADCA0I1	FFFE EA26$_H$	INTADCA0I1	ADCA0 SG2 end interrupt	ADCA0	Edge	1013$_H$	✓	✓	✓	✓			+04C$_H$
20	ICADCA0I2	FFFE EA28$_H$	INTADCA0I2[*6]	ADCA0 SG3 end interrupt	ADCA0	Edge	1014$_H$	✓	✓	✓	✓			+050$_H$
	ICCSIH0IJC_1		INTCSIH0IJC_1[*6]	CSIH0 job completion interrupt	CSIH0	Edge		✓	✓	✓	✓			
21	ICDCUTDI	FFFE EA2A$_H$	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	1015$_H$	✓	✓	✓	✓			+054$_H$
22	ICRCANGERR0	FFFE EA2C$_H$	INTRCANGERR0	CAN global error interrupt	RCFDC0	Level	1016$_H$	✓	✓	✓	✓			+058$_H$
23	ICRCANGRECC0	FFFE EA2E$_H$	INTRCANGRECC0	CAN receive FIFO interrupt	RCFDC0	Level	1017$_H$	✓	✓	✓	✓			+05C$_H$
24	ICRCAN0ERR	FFFE EA30$_H$	INTRCAN0ERR	CAN0 error interrupt	RCFDC0	Level	1018$_H$	✓	✓	✓	✓			+060$_H$
25	ICRCAN0REC	FFFE EA32$_H$	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1019$_H$	✓	✓	✓	✓			+064$_H$
26	ICRCAN0TRX	FFFE EA34$_H$	INTRCAN0TRX	CAN0 transmit interrupt	RCFDC0	Level	101A$_H$	✓	✓	✓	✓			+068$_H$
27	ICCSIG0IC	FFFE EA36$_H$	INTCSIG0IC[*6]	CSIG0 communication status interrupt	CSIG0	Edge	101B$_H$	✓	✓	✓	✓			+06C$_H$
	ICCSIH1IRE_1		INTCSIH1IRE_1[*6]	CSIH1 communication error interrupt	CSIH1	Edge		—	—	✓	✓			
28	ICCSIG0IR	FFFE EA38$_H$	INTCSIG0IR[*6]	CSIG0 receive status interrupt	CSIG0	Edge	101C$_H$	✓	✓	✓	✓			+070$_H$
	ICCSIH1IJC_1		INTCSIH1IJC_1[*6]	CSIH1 job completion interrupt	CSIH1	Edge		—	—	✓	✓			
29	ICCSIH0IC	FFFE EA3A$_H$	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	101D$_H$	✓	✓	✓	✓			+074$_H$
30	ICCSIH0IR	FFFE EA3C$_H$	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	101E$_H$	✓	✓	✓	✓			+078$_H$
31	ICCSIH0IRE	FFFE EA3E$_H$	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	101F$_H$	✓	✓	✓	✓			+07C$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register											Direct Jumping to an Address		Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2						RINT = 0	RINT = 1	
32	ICCSIH0IJC	FFFF B040$_H$	INTCSIH0IJC*6	CSIH0 job completion interrupt	CSIH0	Edge	1020$_H$	✓	✓	✓	✓	*3	*4	+080$_H$
	ICADCA0I2_2		INTADCA0I2_2*6	ADCA0 SG3 end interrupt	ADCA0	Edge		✓	✓	✓	✓			
33	ICRLIN30	FFFF B042$_H$	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1021$_H$	✓	✓	✓	✓			+084$_H$
34	ICRLIN30UR0	FFFF B044$_H$	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	1022$_H$	✓	✓	✓	✓			+088$_H$
35	ICRLIN30UR1	FFFF B046$_H$	INTRLIN30UR1	RLIN30 receive complete interrupt	RLIN30	Edge	1023$_H$	✓	✓	✓	✓			+08C$_H$
36	ICRLIN30UR2	FFFF B048$_H$	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	1024$_H$	✓	✓	✓	✓			+090$_H$
37	ICP0	FFFF B04A$_H$	INTP0	External interrupt	Port	Edge	1025$_H$	✓	✓	✓	✓			+094$_H$
38	ICP1	FFFF B04C$_H$	INTP1	External interrupt	Port	Edge	1026$_H$	✓	✓	✓	✓			+098$_H$
39	ICP2	FFFF B04E$_H$	INTP2	External interrupt	Port	Edge	1027$_H$	✓	✓	✓	✓			+09C$_H$
40	ICWDTA0	FFFF B050$_H$	INTWDTA0	WDTA0 75% interrupt	WDTA0	Edge	1028$_H$	✓	✓	✓	✓			+0A0$_H$
41	ICWDTA1	FFFF B052$_H$	INTWDTA1	WDTA1 75% interrupt	WDTA1	Edge	1029$_H$	✓	✓	✓	✓			+0A4$_H$
42	Reserved	FFFF B054$_H$					102A$_H$	—	—	—	—			+0A8$_H$
43	ICP3	FFFF B056$_H$	INTP3	External interrupt	Port	Edge	102B$_H$	✓	✓	✓	✓			+0AC$_H$
44	ICP4	FFFF B058$_H$	INTP4	External interrupt	Port	Edge	102C$_H$	✓	✓	✓	✓			+0B0$_H$
45	ICP5	FFFF B05A$_H$	INTP5	External interrupt	Port	Edge	102D$_H$	✓	✓	✓	✓			+0B4$_H$
46	ICP10	FFFF B05C$_H$	INTP10	External interrupt	Port	Edge	102E$_H$	✓	✓	✓	✓			+0B8$_H$
47	ICP11	FFFF B05E$_H$	INTP11	External interrupt	Port	Edge	102F$_H$	✓	✓	✓	✓			+0BC$_H$
48	ICTAUD0I1	FFFF B060$_H$	INTTAUD0I1	Interrupt for TAUD0 channel 1	TAUD0	Edge	1030$_H$	✓	✓	✓	✓			+0C0$_H$
49	ICTAUD0I3	FFFF B062$_H$	INTTAUD0I3	Interrupt for TAUD0 channel 3	TAUD0	Edge	1031$_H$	✓	✓	✓	✓			+0C4$_H$
50	ICTAUD0I5	FFFF B064$_H$	INTTAUD0I5	Interrupt for TAUD0 channel 5	TAUD0	Edge	1032$_H$	✓	✓	✓	✓			+0C8$_H$
51	ICTAUD0I7	FFFF B066$_H$	INTTAUD0I7	Interrupt for TAUD0 channel 7	TAUD0	Edge	1033$_H$	✓	✓	✓	✓			+0CC$_H$
52	ICTAUD0I9	FFFF B068$_H$	INTTAUD0I9	Interrupt for TAUD0 channel 9	TAUD0	Edge	1034$_H$	✓	✓	✓	✓			+0D0$_H$
53	ICTAUD0I11	FFFF B06A$_H$	INTTAUD0I11	Interrupt for TAUD0 channel 11	TAUD0	Edge	1035$_H$	✓	✓	✓	✓			+0D4$_H$
54	ICTAUD0I13	FFFF B06C$_H$	INTTAUD0I13	Interrupt for TAUD0 channel 13	TAUD0	Edge	1036$_H$	✓	✓	✓	✓			+0D8$_H$
55	ICTAUD0I15	FFFF B06E$_H$	INTTAUD0I15	Interrupt for TAUD0 channel 15	TAUD0	Edge	1037$_H$	✓	✓	✓	✓			+0DC$_H$
56	ICADCA0ERR	FFFF B070$_H$	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	1038$_H$	✓	✓	✓	✓			+0E0$_H$
57	ICCSIG0IRE	FFFF B072$_H$	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1039$_H$	✓	✓	✓	✓			+0E4$_H$
58	ICRLIN20	FFFF B074$_H$	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	103A$_H$	✓	✓	✓	✓			+0E8$_H$
59	ICRLIN21	FFFF B076$_H$	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	103B$_H$	✓	✓	✓	✓			+0EC$_H$
60	ICDMA0	FFFF B078$_H$	INTDMA0	DMA00 transfer completion	PDMA0	Edge	103C$_H$	✓	✓	✓	✓			+0F0$_H$
61	ICDMA1	FFFF B07A$_H$	INTDMA1	DMA01 transfer completion	PDMA0	Edge	103D$_H$	✓	✓	✓	✓			+0F4$_H$
62	ICDMA2	FFFF B07C$_H$	INTDMA2	DMA02 transfer completion	PDMA0	Edge	103E$_H$	✓	✓	✓	✓			+0F8$_H$
63	ICDMA3	FFFF B07E$_H$	INTDMA3	DMA03 transfer completion	PDMA0	Edge	103F$_H$	✓	✓	✓	✓			+0FC$_H$
64	ICDMA4	FFFF B080$_H$	INTDMA4	DMA04 transfer completion	PDMA0	Edge	1040$_H$	✓	✓	✓	✓			+100$_H$
65	ICDMA5	FFFF B082$_H$	INTDMA5	DMA05 transfer completion	PDMA0	Edge	1041$_H$	✓	✓	✓	✓			+104$_H$
66	ICDMA6	FFFF B084$_H$	INTDMA6	DMA06 transfer completion	PDMA0	Edge	1042$_H$	✓	✓	✓	✓			+108$_H$
67	ICDMA7	FFFF B086$_H$	INTDMA7	DMA07 transfer completion	PDMA0	Edge	1043$_H$	✓	✓	✓	✓			+10C$_H$
68	ICDMA8	FFFF B088$_H$	INTDMA8	DMA10 transfer completion	PDMA0	Edge	1044$_H$	✓	✓	✓	✓			+110$_H$
69	ICDMA9	FFFF B08A$_H$	INTDMA9	DMA11 transfer completion	PDMA0	Edge	1045$_H$	✓	✓	✓	✓			+114$_H$
70	ICDMA10	FFFF B08C$_H$	INTDMA10	DMA12 transfer completion	PDMA0	Edge	1046$_H$	✓	✓	✓	✓			+118$_H$
71	ICDMA11	FFFF B08E$_H$	INTDMA11	DMA13 transfer completion	PDMA0	Edge	1047$_H$	✓	✓	✓	✓			+11C$_H$
72	ICDMA12	FFFF B090$_H$	INTDMA12	DMA14 transfer completion	PDMA0	Edge	1048$_H$	✓	✓	✓	✓			+120$_H$
73	ICDMA13	FFFF B092$_H$	INTDMA13	DMA15 transfer completion	PDMA0	Edge	1049$_H$	✓	✓	✓	✓			+124$_H$
74	ICDMA14	FFFF B094$_H$	INTDMA14	DMA16 transfer completion	PDMA0	Edge	104A$_H$	✓	✓	✓	✓			+128$_H$
75	ICDMA15	FFFF B096$_H$	INTDMA15	DMA17 transfer completion	PDMA0	Edge	104B$_H$	✓	✓	✓	✓			+12C$_H$
76	ICRIIC0TI	FFFF B098$_H$	INTRIIC0TI	RIIC0 transmit data empty interrupt	RIIC0	Edge	104C$_H$	✓	✓	✓	✓			+130$_H$
77	ICRIIC0EE	FFFF B09A$_H$	INTRIIC0EE	RIIC0 receive error/event interrupt	RIIC0	Level	104D$_H$	✓	✓	✓	✓			+134$_H$
78	ICRIIC0RI	FFFF B09C$_H$	INTRIIC0RI	RIIC0 receive complete interrupt	RIIC0	Edge	104E$_H$	✓	✓	✓	✓			+138$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[8]		
Channel No.[1]	Control Register					Detection Type[2]						Direct Jumping to an Address		Reference to a Table[5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
79	ICRIIC0TEI	FFFF B09E$_H$	INTRIIC0TEI	RIIC0 transmit complete interrupt	RIIC0	Level	104F$_H$	✓	✓	✓	✓	[3]	[4]	+13C$_H$
80	ICTAUJ0I0	FFFF B0A0$_H$	INTTAUJ0I0	Interrupt for TAUJ0 channel 0	TAUJ0	Edge	1050$_H$	✓	✓	✓	✓			+140$_H$
81	ICTAUJ0I1	FFFF B0A2$_H$	INTTAUJ0I1	Interrupt for TAUJ0 channel 1	TAUJ0	Edge	1051$_H$	✓	✓	✓	✓			+144$_H$
82	ICTAUJ0I2	FFFF B0A4$_H$	INTTAUJ0I2	Interrupt for TAUJ0 channel 2	TAUJ0	Edge	1052$_H$	✓	✓	✓	✓			+148$_H$
83	ICTAUJ0I3	FFFF B0A6$_H$	INTTAUJ0I3	Interrupt for TAUJ0 channel 3	TAUJ0	Edge	1053$_H$	✓	✓	✓	✓			+14C$_H$
84	ICOSTM0	FFFF B0A8$_H$	INTOSTM0[9]	OSTM0 interrupt	OSTM0	Edge	1054$_H$	✓	✓	✓	✓			+150$_H$
85	ICENCA0IOV	FFFF B0AA$_H$	INTENCA0IOV	ENCA0 overflow interrupt	ENCA0	Edge	1055$_H$	✓	✓	✓	✓			+154$_H$
86	ICENCA0IUD	FFFF B0AC$_H$	INTENCA0IUD	ENCA0 underflow interrupt	ENCA0	Edge	1056$_H$	✓	✓	✓	✓			+158$_H$
87	ICENCA0I0	FFFF B0AE$_H$	INTENCA0I0	ENCA0 match/capture interrupt 0	ENCA0	Edge	1057$_H$	✓	✓	✓	✓			+15C$_H$
88	ICENCA0I1	FFFF B0B0$_H$	INTENCA0I1	ENCA0 match/capture interrupt 1	ENCA0	Edge	1058$_H$	✓	✓	✓	✓			+160$_H$
89	ICENCA0IEC	FFFF B0B2$_H$	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	Edge	1059$_H$	✓	✓	✓	✓			+164$_H$
90	ICKR0	FFFF B0B4$_H$	INTKR0	KR0 key interrupt	KR0	Edge	105A$_H$	✓	✓	✓	✓			+168$_H$
91	ICQFULL	FFFF B0B6$_H$	INTQFULL	PWSA0 queue full interrupt	PWSA0	Edge	105B$_H$	✓	✓	✓	✓			+16C$_H$
92	ICPWGAG00	FFFF B0B8$_H$	INTPWGAG00[7]	PWGA interrupt group 00 (PWGA0 to PWGA31)	PWGA0-31	Level	105C$_H$	✓	✓	✓	✓			+170$_H$
93	ICPWGAG01	FFFF B0BA$_H$	INTPWGAG01[7]	PWGA interrupt group 01 (PWGA32 to PWGA63)	PWGA32-63	Level	105D$_H$	—	—	—	✓			+174$_H$
94	Reserved	FFFF B0BC$_H$					105E$_H$	—	—	—	—			+178$_H$
95	Reserved	FFFF B0BE$_H$					105F$_H$	—	—	—	—			+17C$_H$
96	Reserved	FFFF B0C0$_H$					1060$_H$	—	—	—	—			+180$_H$
97	Reserved	FFFF B0C2$_H$					1061$_H$	—	—	—	—			+184$_H$
98	Reserved	FFFF B0C4$_H$					1062$_H$	—	—	—	—			+188$_H$
99	Reserved	FFFF B0C6$_H$					1063$_H$	—	—	—	—			+18C$_H$
100	Reserved	FFFF B0C8$_H$					1064$_H$	—	—	—	—			+190$_H$
101	Reserved	FFFF B0CA$_H$					1065$_H$	—	—	—	—			+194$_H$
102	Reserved	FFFF B0CC$_H$					1066$_H$	—	—	—	—			+198$_H$
103	Reserved	FFFF B0CE$_H$					1067$_H$	—	—	—	—			+19C$_H$
104	Reserved	FFFF B0D0$_H$					1068$_H$	—	—	—	—			+1A0$_H$
105	Reserved	FFFF B0D2$_H$					1069$_H$	—	—	—	—			+1A4$_H$
106	Reserved	FFFF B0D4$_H$					106A$_H$	—	—	—	—			+1A8$_H$
107	Reserved	FFFF B0D6$_H$					106B$_H$	—	—	—	—			+1AC$_H$
108	Reserved	FFFF B0D8$_H$					106C$_H$	—	—	—	—			+1B0$_H$
109	Reserved	FFFF B0DA$_H$					106D$_H$	—	—	—	—			+1B4$_H$
110	ICFLERR	FFFF B0DC$_H$	INTFLERR[10]	Flash sequencer end error interrupt	FACI	Level	106E$_H$	✓	✓	✓	✓			+1B8$_H$
111	ICFLENDNM	FFFF B0DE$_H$	INTFLENDNM[10]	Flash sequencer end interrupt	FACI	Edge	106F$_H$	✓	✓	✓	✓			+1BC$_H$
112	ICCWEND	FFFF B0E0$_H$	INTCWEND	LPS0 port polling end interrupt	LPS0	Edge	1070$_H$	✓	✓	✓	✓			+1C0$_H$
113	ICRCAN1ERR	FFFF B0E2$_H$	INTRCAN1ERR	CAN1 error interrupt	RCFDC0	Level	1071$_H$	—	✓	✓	✓			+1C4$_H$
114	ICRCAN1REC	FFFF B0E4$_H$	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1072$_H$	—	✓	✓	✓			+1C8$_H$
115	ICRCAN1TRX	FFFF B0E6$_H$	INTRCAN1TRX	CAN1 transmit interrupt	RCFDC0	Level	1073$_H$	—	✓	✓	✓			+1CC$_H$
116	ICCSIH1IC	FFFF B0E8$_H$	INTCSIH1IC[6]	CSIH1 communication status interrupt	CSIH1	Edge	1074$_H$	—	—	✓	✓			+1D0$_H$
	ICTAPA0IPEK0_2		INTTAPA0IPEK0_2[6]	TAPA0 peak interrupt 0	TAPA0	Edge		✓	✓	✓	✓			
117	ICCSIH1IR	FFFF B0EA$_H$	INTCSIH1IR[6]	CSIH1 receive status interrupt	CSIH1	Edge	1075$_H$	—	—	✓	✓			+1D4$_H$
	ICTAPA0IVLY0_2		INTTAPA0IVLY0_2[6]	TAPA0 valley interrupt 0	TAPA0	Edge		✓	✓	✓	✓			
118	ICCSIH1IRE	FFFF B0EC$_H$	INTCSIH1IRE[6]	CSIH1 communication error interrupt	CSIH1	Edge	1076$_H$	—	—	✓	✓			+1D8$_H$
	ICCSIG0IC_2		INTCSIG0IC_2[6]	CSIG0 communication status interrupt	CSIG0	Edge		✓	✓	✓	✓			
119	ICCSIH1IJC	FFFF B0EE$_H$	INTCSIH1IJC[6]	CSIH1 job completion interrupt	CSIH1	Edge	1077$_H$	—	—	✓	✓			+1DC$_H$
	ICCSIG0IR_2		INTCSIG0IR_2[6]	CSIG0 receive status interrupt	CSIG0	Edge		✓	✓	✓	✓			
120	ICRLIN31	FFFF B0F0$_H$	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1078$_H$	—	✓	✓	✓			+1E0$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[8]		
Channel No.[1]	Control Register					Detection Type[2]						Direct Jumping to an Address		Reference to a Table[5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
121	ICRLIN31UR0	FFFF B0F2$_H$	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1079$_H$	—	✓	✓	✓	[3]	[4]	+1E4$_H$
122	ICRLIN31UR1	FFFF B0F4$_H$	INTRLIN31UR1	RLIN31 receive complete interrupt	RLIN31	Edge	107A$_H$	—	✓	✓	✓			+1E8$_H$
123	ICRLIN31UR2	FFFF B0F6$_H$	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	107B$_H$	—	✓	✓	✓			+1EC$_H$
124	Reserved	FFFF B0F8$_H$					107C$_H$	—	—	—	—			+1F0$_H$
125	Reserved	FFFF B0FA$_H$					107D$_H$	—	—	—	—			+1F4$_H$
126	Reserved	FFFF B0FC$_H$					107E$_H$	—	—	—	—			+1F8$_H$
127	Reserved	FFFF B0FE$_H$					107F$_H$	—	—	—	—			+1FC$_H$
128	ICP6	FFFF B100$_H$	INTP6	External interrupt	Port	Edge	1080$_H$	—	—	✓	✓			+200$_H$
129	ICP7	FFFF B102$_H$	INTP7	External interrupt	Port	Edge	1081$_H$	—	—	✓	✓			+204$_H$
130	ICP8	FFFF B104$_H$	INTP8	External interrupt	Port	Edge	1082$_H$	—	—	✓	✓			+208$_H$
131	ICP12	FFFF B106$_H$	INTP12	External interrupt	Port	Edge	1083$_H$	—	—	✓	✓			+20C$_H$
132	ICCSIH2IC	FFFF B108$_H$	INTCSIH2IC[6]	CSIH2 communication status interrupt	CSIH2	Edge	1084$_H$	—	—	✓	✓			+210$_H$
	ICTAUD0I0_2		INTTAUD0I0_2[6]	Interrupt for TAUD0 channel 0	TAUD0	Edge		✓	✓	✓	✓			
133	ICCSIH2IR	FFFF B10A$_H$	INTCSIH2IR[6]	CSIH2 receive status interrupt	CSIH2	Edge	1085$_H$	—	—	✓	✓			+214$_H$
	ICTAUD0I4_2		INTTAUD0I4_2[6]	Interrupt for TAUD0 channel 4	TAUD0	Edge		✓	✓	✓	✓			
134	ICCSIH2IRE	FFFF B10C$_H$	INTCSIH2IRE[6]	CSIH2 communication error interrupt	CSIH2	Edge	1086$_H$	—	—	✓	✓			+218$_H$
	ICTAUD0I6_2		INTTAUD0I6_2[6]	Interrupt for TAUD0 channel 6	TAUD0	Edge		✓	✓	✓	✓			
135	ICCSIH2IJC	FFFF B10E$_H$	INTCSIH2IJC[6]	CSIH2 job completion interrupt	CSIH2	Edge	1087$_H$	—	—	✓	✓			+21C$_H$
	ICTAUD0I8_2		INTTAUD0I8_2[6]	Interrupt for TAUD0 channel 8	TAUD0	Edge		✓	✓	✓	✓			
136	Reserved	FFFF B110$_H$					1088$_H$	—	—	—	—			+220$_H$
137	Reserved	FFFF B112$_H$					1089$_H$	—	—	—	—			+224$_H$
138	Reserved	FFFF B114$_H$					108A$_H$	—	—	—	—			+228$_H$
139	Reserved	FFFF B116$_H$					108B$_H$	—	—	—	—			+22C$_H$
140	Reserved	FFFF B118$_H$					108C$_H$	—	—	—	—			+230$_H$
141	Reserved	FFFF B11A$_H$					108D$_H$	—	—	—	—			+234$_H$
142	ICTAUB0I0	FFFF B11C$_H$	INTTAUB0I0	Interrupt for TAUB0 channel 0	TAUB0	Edge	108E$_H$	—	—	✓	✓			+238$_H$
143	ICTAUB0I1	FFFF B11E$_H$	INTTAUB0I1	Interrupt for TAUB0 channel 1	TAUB0	Edge	108F$_H$	—	—	✓	✓			+23C$_H$
144	ICTAUB0I2	FFFF B120$_H$	INTTAUB0I2	Interrupt for TAUB0 channel 2	TAUB0	Edge	1090$_H$	—	—	✓	✓			+240$_H$
145	ICTAUB0I3	FFFF B122$_H$	INTTAUB0I3	Interrupt for TAUB0 channel 3	TAUB0	Edge	1091$_H$	—	—	✓	✓			+244$_H$
146	ICTAUB0I4	FFFF B124$_H$	INTTAUB0I4	Interrupt for TAUB0 channel 4	TAUB0	Edge	1092$_H$	—	—	✓	✓			+248$_H$
147	ICTAUB0I5	FFFF B126$_H$	INTTAUB0I5	Interrupt for TAUB0 channel 5	TAUB0	Edge	1093$_H$	—	—	✓	✓			+24C$_H$
148	ICTAUB0I6	FFFF B128$_H$	INTTAUB0I6	Interrupt for TAUB0 channel 6	TAUB0	Edge	1094$_H$	—	—	✓	✓			+250$_H$
149	ICTAUB0I7	FFFF B12A$_H$	INTTAUB0I7	Interrupt for TAUB0 channel 7	TAUB0	Edge	1095$_H$	—	—	✓	✓			+254$_H$
150	ICTAUB0I8	FFFF B12C$_H$	INTTAUB0I8	Interrupt for TAUB0 channel 8	TAUB0	Edge	1096$_H$	—	—	✓	✓			+258$_H$
151	ICTAUB0I9	FFFF B12E$_H$	INTTAUB0I9	Interrupt for TAUB0 channel 9	TAUB0	Edge	1097$_H$	—	—	✓	✓			+25C$_H$
152	ICTAUB0I10	FFFF B130$_H$	INTTAUB0I10	Interrupt for TAUB0 channel 10	TAUB0	Edge	1098$_H$	—	—	✓	✓			+260$_H$
153	ICTAUB0I11	FFFF B132$_H$	INTTAUB0I11	Interrupt for TAUB0 channel 11	TAUB0	Edge	1099$_H$	—	—	✓	✓			+264$_H$
154	ICTAUB0I12	FFFF B134$_H$	INTTAUB0I12	Interrupt for TAUB0 channel 12	TAUB0	Edge	109A$_H$	—	—	✓	✓			+268$_H$
155	ICTAUB0I13	FFFF B136$_H$	INTTAUB0I13	Interrupt for TAUB0 channel 13	TAUB0	Edge	109B$_H$	—	—	✓	✓			+26C$_H$
156	ICTAUB0I14	FFFF B138$_H$	INTTAUB0I14	Interrupt for TAUB0 channel 14	TAUB0	Edge	109C$_H$	—	—	✓	✓			+270$_H$
157	ICTAUB0I15	FFFF B13A$_H$	INTTAUB0I15	Interrupt for TAUB0 channel 15	TAUB0	Edge	109D$_H$	—	—	✓	✓			+274$_H$
158	ICCSIH3IC	FFFF B13C$_H$	INTCSIH3IC[6]	CSIH3 communication status interrupt	CSIH3	Edge	109E$_H$	—	—	—	✓			+278$_H$
	ICTAUD0I2_2		INTTAUD0I2_2[6]	Interrupt for TAUD0 channel 2	TAUD0	Edge		✓	✓	✓	✓			
159	ICCSIH3IR	FFFF B13E$_H$	INTCSIH3IR[6]	CSIH3 receive status interrupt	CSIH3	Edge	109F$_H$	—	—	—	✓			+27C$_H$
	ICTAUD0I10_2		INTTAUD0I10_2[6]	Interrupt for TAUD0 channel 10	TAUD0	Edge		✓	✓	✓	✓			
160	ICCSIH3IRE	FFFF B140$_H$	INTCSIH3IRE[6]	CSIH3 communication error interrupt	CSIH3	Edge	10A0$_H$	—	—	—	✓			+280$_H$
	ICTAUD0I12_2		INTTAUD0I12_2[6]	Interrupt for TAUD0 channel 12	TAUD0	Edge		✓	✓	✓	✓			

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register											Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit	Detection Type[*2]						RINT = 0	RINT = 1	
161	ICCSIH3IJC	FFFF B142$_H$	INTCSIH3IJC[*6]	CSIH3 job completion interrupt	CSIH3	Edge	10A1$_H$	—	—	—	✓	[*3]	[*4]	+284$_H$
	ICTAUD0I14_2		INTTAUD0I14_2[*6]	Interrupt for TAUD0 channel 14	TAUD0	Edge		✓	✓	✓	✓			
162	ICRLIN22	FFFF B144$_H$	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	10A2$_H$	—	—	—	✓			+288$_H$
163	Reserved	FFFF B146$_H$					10A3$_H$	—	—	—	—			+28C$_H$
164	ICRLIN32	FFFF B148$_H$	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	10A4$_H$	—	—	✓	✓			+290$_H$
165	ICRLIN32UR0	FFFF B14A$_H$	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	10A5$_H$	—	—	✓	✓			+294$_H$
166	ICRLIN32UR1	FFFF B14C$_H$	INTRLIN32UR1	RLIN32 receive complete interrupt	RLIN32	Edge	10A6$_H$	—	—	✓	✓			+298$_H$
167	ICRLIN32UR2	FFFF B14E$_H$	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	10A7$_H$	—	—	✓	✓			+29C$_H$
168	ICTAUJ1I0	FFFF B150$_H$	INTTAUJ1I0	Interrupt for TAUJ1 channel 0	TAUJ1	Edge	10A8$_H$	✓	✓	✓	✓			+2A0$_H$
169	ICTAUJ1I1	FFFF B152$_H$	INTTAUJ1I1	Interrupt for TAUJ1 channel 1	TAUJ1	Edge	10A9$_H$	✓	✓	✓	✓			+2A4$_H$
170	ICTAUJ1I2	FFFF B154$_H$	INTTAUJ1I2	Interrupt for TAUJ1 channel 2	TAUJ1	Edge	10AA$_H$	✓	✓	✓	✓			+2A8$_H$
171	ICTAUJ1I3	FFFF B156$_H$	INTTAUJ1I3	Interrupt for TAUJ1 channel 3	TAUJ1	Edge	10AB$_H$	✓	✓	✓	✓			+2AC$_H$
172	Reserved	FFFF B158$_H$					10AC$_H$	—	—	—	—			+2B0$_H$
173	Reserved	FFFF B15A$_H$					10AD$_H$	—	—	—	—			+2B4$_H$
174	Reserved	FFFF B15C$_H$					10AE$_H$	—	—	—	—			+2B8$_H$
175	Reserved	FFFF B15E$_H$					10AF$_H$	—	—	—	—			+2BC$_H$
176	Reserved	FFFF B160$_H$					10B0$_H$	—	—	—	—			+2C0$_H$
177	Reserved	FFFF B162$_H$					10B1$_H$	—	—	—	—			+2C4$_H$
178	Reserved	FFFF B164$_H$					10B2$_H$	—	—	—	—			+2C8$_H$
179	Reserved	FFFF B166$_H$					10B3$_H$	—	—	—	—			+2CC$_H$
180	Reserved	FFFF B168$_H$					10B4$_H$	—	—	—	—			+2D0$_H$
181	Reserved	FFFF B16A$_H$					10B5$_H$	—	—	—	—			+2D4$_H$
182	Reserved	FFFF B16C$_H$					10B6$_H$	—	—	—	—			+2D8$_H$
183	Reserved	FFFF B16E$_H$					10B7$_H$	—	—	—	—			+2DC$_H$
184	Reserved	FFFF B170$_H$					10B8$_H$	—	—	—	—			+2E0$_H$
185	Reserved	FFFF B172$_H$					10B9$_H$	—	—	—	—			+2E4$_H$
186	Reserved	FFFF B174$_H$					10BA$_H$	—	—	—	—			+2E8$_H$
187	Reserved	FFFF B176$_H$					10BB$_H$	—	—	—	—			+2EC$_H$
188	Reserved	FFFF B178$_H$					10BC$_H$	—	—	—	—			+2F0$_H$
189	Reserved	FFFF B17A$_H$					10BD$_H$	—	—	—	—			+2F4$_H$
190	Reserved	FFFF B17C$_H$					10BE$_H$	—	—	—	—			+2F8$_H$
191	Reserved	FFFF B17E$_H$					10BF$_H$	—	—	—	—			+2FC$_H$
192	Reserved	FFFF B180$_H$					10C0$_H$	—	—	—	—			+300$_H$
193	Reserved	FFFF B182$_H$					10C1$_H$	—	—	—	—			+304$_H$
194	Reserved	FFFF B184$_H$					10C2$_H$	—	—	—	—			+308$_H$
195	Reserved	FFFF B186$_H$					10C3$_H$	—	—	—	—			+30C$_H$
196	Reserved	FFFF B188$_H$					10C4$_H$	—	—	—	—			+310$_H$
197	Reserved	FFFF B18A$_H$					10C5$_H$	—	—	—	—			+314$_H$
198	Reserved	FFFF B18C$_H$					10C6$_H$	—	—	—	—			+318$_H$
199	Reserved	FFFF B18E$_H$					10C7$_H$	—	—	—	—			+31C$_H$
200	Reserved	FFFF B190$_H$					10C8$_H$	—	—	—	—			+320$_H$
201	Reserved	FFFF B192$_H$					10C9$_H$	—	—	—	—			+324$_H$
202	Reserved	FFFF B194$_H$					10CA$_H$	—	—	—	—			+328$_H$
203	Reserved	FFFF B196$_H$					10CB$_H$	—	—	—	—			+32C$_H$
204	Reserved	FFFF B198$_H$					10CC$_H$	—	—	—	—			+330$_H$
205	Reserved	FFFF B19A$_H$					10CD$_H$	—	—	—	—			+334$_H$
206	ICP13	FFFF B19C$_H$	INTP13	External interrupt	Port	Edge	10CE$_H$	—	—	—	✓			+338$_H$
207	Reserved	FFFF B19E$_H$					10CF$_H$	—	—	—	—			+33C$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[8]		
Channel No.[1]	Control Register					Detection Type[2]						Direct Jumping to an Address		Reference to a Table[5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
208	Reserved	FFFF B1A0$_H$					10D0$_H$	—	—	—	—	[3]	[4]	+340$_H$
209	ICRTCA01S	FFFF B1A2$_H$	INTRTCA01S	RTCA0 1-second interval interrupt	RTCA0	Edge	10D1$_H$	✓	✓	✓	✓			+344$_H$
210	ICRTCA0AL	FFFF B1A4$_H$	INTRTCA0AL	RTCA0 alarm interrupt	RTCA0	Edge	10D2$_H$	✓	✓	✓	✓			+348$_H$
211	ICRTCA0R	FFFF B1A6$_H$	INTRTCA0R	RTCA0 fixed interval interrupt	RTCA0	Edge	10D3$_H$	✓	✓	✓	✓			+34C$_H$
212	Reserved	FFFF B1A8$_H$					10D4$_H$	—	—	—	—			+350$_H$
213	Reserved	FFFF B1AA$_H$					10D5$_H$	—	—	—	—			+354$_H$
214	Reserved	FFFF B1AC$_H$					10D6$_H$	—	—	—	—			+358$_H$
215	Reserved	FFFF B1AE$_H$					10D7$_H$	—	—	—	—			+35C$_H$
216	Reserved	FFFF B1B0$_H$					10D8$_H$	—	—	—	—			+360$_H$
217	ICRCAN2ERR	FFFF B1B2$_H$	INTRCAN2ERR	CAN2 error interrupt	RCFDC0	Level	10D9$_H$	—	✓	✓	✓			+364$_H$
218	ICRCAN2REC	FFFF B1B4$_H$	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DA$_H$	—	✓	✓	✓			+368$_H$
219	ICRCAN2TRX	FFFF B1B6$_H$	INTRCAN2TRX	CAN2 transmit interrupt	RCFDC0	Level	10DB$_H$	—	✓	✓	✓			+36C$_H$
220	ICRCAN3ERR	FFFF B1B8$_H$	INTRCAN3ERR	CAN3 error interrupt	RCFDC0	Level	10DC$_H$	—	—	—	✓			+370$_H$
221	ICRCAN3REC	FFFF B1BA$_H$	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	10DD$_H$	—	—	—	✓			+374$_H$
222	ICRCAN3TRX	FFFF B1BC$_H$	INTRCAN3TRX	CAN3 transmit interrupt	RCFDC0	Level	10DE$_H$	—	—	—	✓			+378$_H$
223	Reserved	FFFF B1BE$_H$					10DF$_H$	—	—	—	—			+37C$_H$
224	Reserved	FFFF B1C0$_H$					10E0$_H$	—	—	—	—			+380$_H$
225	Reserved	FFFF B1C2$_H$					10E1$_H$	—	—	—	—			+384$_H$
226	Reserved	FFFF B1C4$_H$					10E2$_H$	—	—	—	—			+388$_H$
227	Reserved	FFFF B1C6$_H$					10E3$_H$	—	—	—	—			+38C$_H$
228	ICRLIN33	FFFF B1C8$_H$	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10E4$_H$	—	—	—	✓			+390$_H$
229	ICRLIN33UR0	FFFF B1CA$_H$	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10E5$_H$	—	—	—	✓			+394$_H$
230	ICRLIN33UR1	FFFF B1CC$_H$	INTRLIN33UR1	RLIN33 receive complete interrupt	RLIN33	Edge	10E6$_H$	—	—	—	✓			+398$_H$
231	ICRLIN33UR2	FFFF B1CE$_H$	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10E7$_H$	—	—	—	✓			+39C$_H$
232	Reserved	FFFF B1D0$_H$					10E8$_H$	—	—	—	—			+3A0$_H$
233	Reserved	FFFF B1D2$_H$					10E9$_H$	—	—	—	—			+3A4$_H$
234	Reserved	FFFF B1D4$_H$					10EA$_H$	—	—	—	—			+3A8$_H$
235	Reserved	FFFF B1D6$_H$					10EB$_H$	—	—	—	—			+3AC$_H$
236	Reserved	FFFF B1D8$_H$					10EC$_H$	—	—	—	—			+3B0$_H$
237	Reserved	FFFF B1DA$_H$					10ED$_H$	—	—	—	—			+3B4$_H$
238	Reserved	FFFF B1DC$_H$					10EE$_H$	—	—	—	—			+3B8$_H$
239	Reserved	FFFF B1DE$_H$					10EF$_H$	—	—	—	—			+3BC$_H$
240	ICRIIC1TI	FFFF B1E0$_H$	INTRIIC1TI	RIIC1 transmit data empty interrupt	RIIC1	Edge	10F0$_H$	✓	✓	✓	✓			+3C0$_H$
241	ICRIIC1EE	FFFF B1E2$_H$	INTRIIC1EE	RIIC1 receive error/event interrupt	RIIC1	Level	10F1$_H$	✓	✓	✓	✓			+3C4$_H$
242	ICRIIC1RI	FFFF B1E4$_H$	INTRIIC1RI	RIIC1 receive complete interrupt	RIIC1	Edge	10F2$_H$	✓	✓	✓	✓			+3C8$_H$
243	ICRIIC1TEI	FFFF B1E6$_H$	INTRIIC1TEI	RIIC1 transmit complete interrupt	RIIC1	Level	10F3$_H$	✓	✓	✓	✓			+3CC$_H$
244	Reserved	FFFF B1E8$_H$					10F4$_H$	—	—	—	—			+3D0$_H$
245	Reserved	FFFF B1EA$_H$					10F5$_H$	—	—	—	—			+3D4$_H$
246	Reserved	FFFF B1EC$_H$					10F6$_H$	—	—	—	—			+3D8$_H$
247	Reserved	FFFF B1EE$_H$					10F7$_H$	—	—	—	—			+3DC$_H$
248	Reserved	FFFF B1F0$_H$					10F8$_H$	—	—	—	—			+3E0$_H$
249	Reserved	FFFF B1F2$_H$					10F9$_H$	—	—	—	—			+3E4$_H$
250	Reserved	FFFF B1F4$_H$					10FA$_H$	—	—	—	—			+3E8$_H$
251	Reserved	FFFF B1F6$_H$					10FB$_H$	—	—	—	—			+3EC$_H$
252	Reserved	FFFF B1F8$_H$					10FC$_H$	—	—	—	—			+3F0$_H$
253	Reserved	FFFF B1FA$_H$					10FD$_H$	—	—	—	—			+3F4$_H$
254	Reserved	FFFF B1FC$_H$					10FE$_H$	—	—	—	—			+3F8$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[*8]		
Channel No.[*1]	Control Register											Direct Jumping to an Address		Reference to a Table[*5]
	Name	Address	Name	Source	Unit	Detection Type[*2]						RINT = 0	RINT = 1	
255	Reserved	FFFF B1FE$_{H}$					10FF$_{H}$	—	—	—	—	[*3]	[*4]	+3FC$_{H}$
256	Reserved	FFFF B200$_{H}$					1100$_{H}$	—	—	—	—			+400$_{H}$
257	Reserved	FFFF B202$_{H}$					1101$_{H}$	—	—	—	—			+404$_{H}$
258	Reserved	FFFF B204$_{H}$					1102$_{H}$	—	—	—	—			+408$_{H}$
259	Reserved	FFFF B206$_{H}$					1103$_{H}$	—	—	—	—			+40C$_{H}$
260	Reserved	FFFF B208$_{H}$					1104$_{H}$	—	—	—	—			+410$_{H}$
261	Reserved	FFFF B20A$_{H}$					1105$_{H}$	—	—	—	—			+414$_{H}$
262	Reserved	FFFF B20C$_{H}$					1106$_{H}$	—	—	—	—			+418$_{H}$
263	Reserved	FFFF B20E$_{H}$					1107$_{H}$	—	—	—	—			+41C$_{H}$
264	Reserved	FFFF B210$_{H}$					1108$_{H}$	—	—	—	—			+420$_{H}$
265	Reserved	FFFF B212$_{H}$					1109$_{H}$	—	—	—	—			+424$_{H}$
266	Reserved	FFFF B214$_{H}$					110A$_{H}$	—	—	—	—			+428$_{H}$
267	Reserved	FFFF B216$_{H}$					110B$_{H}$	—	—	—	—			+42C$_{H}$
268	Reserved	FFFF B218$_{H}$					110C$_{H}$	—	—	—	—			+430$_{H}$
269	Reserved	FFFF B21A$_{H}$					110D$_{H}$	—	—	—	—			+434$_{H}$
270	Reserved	FFFF B21C$_{H}$					110E$_{H}$	—	—	—	—			+438$_{H}$
271	Reserved	FFFF B21E$_{H}$					110F$_{H}$	—	—	—	—			+43C$_{H}$
272	ICRCAN4ERR	FFFF B220$_{H}$	INTRCAN4ERR	CAN4 error interrupt	RCFDC0	Level	1110$_{H}$	—	—	—	✓			+440$_{H}$
273	ICRCAN4REC	FFFF B222$_{H}$	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1111$_{H}$	—	—	—	✓			+444$_{H}$
274	ICRCAN4TRX	FFFF B224$_{H}$	INTRCAN4TRX	CAN4 transmit interrupt	RCFDC0	Level	1112$_{H}$	—	—	—	✓			+448$_{H}$
275	Reserved	FFFF B226$_{H}$					1113$_{H}$	—	—	—	—			+44C$_{H}$
276	Reserved	FFFF B228$_{H}$					1114$_{H}$	—	—	—	—			+450$_{H}$
277	ICTAUJ2I0	FFFF B22A$_{H}$	INTTAUJ2I0	Interrupt for TAUJ2 channel 0	TAUJ2	Edge	1115$_{H}$	✓	✓	✓	✓			+454$_{H}$
278	ICTAUJ2I1	FFFF B22C$_{H}$	INTTAUJ2I1	Interrupt for TAUJ2 channel 1	TAUJ2	Edge	1116$_{H}$	✓	✓	✓	✓			+458$_{H}$
279	ICTAUJ2I2	FFFF B22E$_{H}$	INTTAUJ2I2	Interrupt for TAUJ2 channel 2	TAUJ2	Edge	1117$_{H}$	✓	✓	✓	✓			+45C$_{H}$
280	ICTAUJ2I3	FFFF B230$_{H}$	INTTAUJ2I3	Interrupt for TAUJ2 channel 3	TAUJ2	Edge	1118$_{H}$	✓	✓	✓	✓			+460$_{H}$
281	ICTAUJ3I0	FFFF B232$_{H}$	INTTAUJ3I0	Interrupt for TAUJ3 channel 0	TAUJ3	Edge	1119$_{H}$	✓	✓	✓	✓			+464$_{H}$
282	ICTAUJ3I1	FFFF B234$_{H}$	INTTAUJ3I1	Interrupt for TAUJ3 channel 1	TAUJ3	Edge	111A$_{H}$	✓	✓	✓	✓			+468$_{H}$
283	ICTAUJ3I2	FFFF B236$_{H}$	INTTAUJ3I2	Interrupt for TAUJ3 channel 2	TAUJ3	Edge	111B$_{H}$	✓	✓	✓	✓			+46C$_{H}$
284	ICTAUJ3I3	FFFF B238$_{H}$	INTTAUJ3I3	Interrupt for TAUJ3 channel 3	TAUJ3	Edge	111C$_{H}$	✓	✓	✓	✓			+470$_{H}$
285	Reserved	FFFF B23A$_{H}$					111D$_{H}$	—	—	—	—			+474$_{H}$
286	Reserved	FFFF B23C$_{H}$					111E$_{H}$	—	—	—	—			+478$_{H}$
287	ICRCAN5ERR	FFFF B23E$_{H}$	INTRCAN5ERR	CAN5 error interrupt	RCFDC0	Level	111F$_{H}$	—	—	—	✓			+47C$_{H}$
288	ICRCAN5REC	FFFF B240$_{H}$	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RCFDC0	Level	1120$_{H}$	—	—	—	✓			+480$_{H}$
289	ICRCAN5TRX	FFFF B242$_{H}$	INTRCAN5TRX	CAN5 transmit interrupt	RCFDC0	Level	1121$_{H}$	—	—	—	✓			+484$_{H}$
290	Reserved	FFFF B244$_{H}$					1122$_{H}$	—	—	—	—			+488$_{H}$
291	Reserved	FFFF B246$_{H}$					1123$_{H}$	—	—	—	—			+48C$_{H}$
292	Reserved	FFFF B248$_{H}$					1124$_{H}$	—	—	—	—			+490$_{H}$
293	Reserved	FFFF B24A$_{H}$					1125$_{H}$	—	—	—	—			+494$_{H}$
294	Reserved	FFFF B24C$_{H}$					1126$_{H}$	—	—	—	—			+498$_{H}$
295	Reserved	FFFF B24E$_{H}$					1127$_{H}$	—	—	—	—			+49C$_{H}$
296	Reserved	FFFF B250$_{H}$					1128$_{H}$	—	—	—	—			+4A0$_{H}$
297	Reserved	FFFF B252$_{H}$					1129$_{H}$	—	—	—	—			+4A4$_{H}$
298	Reserved	FFFF B254$_{H}$					112A$_{H}$	—	—	—	—			+4A8$_{H}$
299	Reserved	FFFF B256$_{H}$					112B$_{H}$	—	—	—	—			+4AC$_{H}$
300	Reserved	FFFF B258$_{H}$					112C$_{H}$	—	—	—	—			+4B0$_{H}$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)*8		
Channel No.*1	Control Register Name	Control Register Address	Name	Source	Unit	Detection Type*2						Direct Jumping to an Address RINT = 0	Direct Jumping to an Address RINT = 1	Reference to a Table*5
301	Reserved	FFFF B25A$_H$					112D$_H$	—	—	—	—	*3	*4	+4B4$_H$
302	Reserved	FFFF B25C$_H$					112E$_H$	—	—	—	—			+4B8$_H$
303	Reserved	FFFF B25E$_H$					112F$_H$	—	—	—	—			+4BC$_H$
304	Reserved	FFFF B260$_H$					1130$_H$	—	—	—	—			+4C0$_H$
305	Reserved	FFFF B262$_H$					1131$_H$	—	—	—	—			+4C4$_H$
306	Reserved	FFFF B264$_H$					1132$_H$	—	—	—	—			+4C8$_H$
307	Reserved	FFFF B266$_H$					1133$_H$	—	—	—	—			+4CC$_H$
308	Reserved	FFFF B268$_H$					1134$_H$	—	—	—	—			+4D0$_H$
309	Reserved	FFFF B26A$_H$					1135$_H$	—	—	—	—			+4D4$_H$
310	Reserved	FFFF B26C$_H$					1136$_H$	—	—	—	—			+4D8$_H$
311	Reserved	FFFF B26E$_H$					1137$_H$	—	—	—	—			+4DC$_H$
312	Reserved	FFFF B270$_H$					1138$_H$	—	—	—	—			+4E0$_H$
313	Reserved	FFFF B272$_H$					1139$_H$	—	—	—	—			+4E4$_H$
314	Reserved	FFFF B274$_H$					113A$_H$	—	—	—	—			+4E8$_H$
315	Reserved	FFFF B276$_H$					113B$_H$	—	—	—	—			+4EC$_H$
316	Reserved	FFFF B278$_H$					113C$_H$	—	—	—	—			+4F0$_H$
317	Reserved	FFFF B27A$_H$					113D$_H$	—	—	—	—			+4F4$_H$
318	Reserved	FFFF B27C$_H$					113E$_H$	—	—	—	—			+4F8$_H$
319	Reserved	FFFF B27E$_H$					113F$_H$	—	—	—	—			+4FC$_H$
320	Reserved	FFFF B280$_H$					1140$_H$	—	—	—	—			+500$_H$
321	Reserved	FFFF B282$_H$					1141$_H$	—	—	—	—			+504$_H$
322	Reserved	FFFF B284$_H$					1142$_H$	—	—	—	—			+508$_H$
323	Reserved	FFFF B286$_H$					1143$_H$	—	—	—	—			+50C$_H$
324	Reserved	FFFF B288$_H$					1144$_H$	—	—	—	—			+510$_H$
325	Reserved	FFFF B28A$_H$					1145$_H$	—	—	—	—			+514$_H$
326	Reserved	FFFF B28C$_H$					1146$_H$	—	—	—	—			+518$_H$
327	Reserved	FFFF B28E$_H$					1147$_H$	—	—	—	—			+51C$_H$
328	Reserved	FFFF B290$_H$					1148$_H$	—	—	—	—			+520$_H$
329	Reserved	FFFF B292$_H$					1149$_H$	—	—	—	—			+524$_H$
330	Reserved	FFFF B294$_H$					114A$_H$	—	—	—	—			+528$_H$
331	Reserved	FFFF B296$_H$					114B$_H$	—	—	—	—			+52C$_H$
332	Reserved	FFFF B298$_H$					114C$_H$	—	—	—	—			+530$_H$
333	Reserved	FFFF B29A$_H$					114D$_H$	—	—	—	—			+534$_H$
334	Reserved	FFFF B29C$_H$					114E$_H$	—	—	—	—			+538$_H$
335	Reserved	FFFF B29E$_H$					114F$_H$	—	—	—	—			+53C$_H$
336	Reserved	FFFF B2A0$_H$					1150$_H$	—	—	—	—			+540$_H$
337	Reserved	FFFF B2A2$_H$					1151$_H$	—	—	—	—			+544$_H$
338	Reserved	FFFF B2A4$_H$					1152$_H$	—	—	—	—			+548$_H$
339	Reserved	FFFF B2A6$_H$					1153$_H$	—	—	—	—			+54C$_H$
340	Reserved	FFFF B2A8$_H$					1154$_H$	—	—	—	—			+550$_H$
341	Reserved	FFFF B2AA$_H$					1155$_H$	—	—	—	—			+554$_H$
342	Reserved	FFFF B2AC$_H$					1156$_H$	—	—	—	—			+558$_H$
343	Reserved	FFFF B2AE$_H$					1157$_H$	—	—	—	—			+55C$_H$
344	Reserved	FFFF B2B0$_H$					1158$_H$	—	—	—	—			+560$_H$
345	Reserved	FFFF B2B2$_H$					1159$_H$	—	—	—	—			+564$_H$
346	Reserved	FFFF B2B4$_H$					115A$_H$	—	—	—	—			+568$_H$
347	ICSENT0SI	FFFF B2B6$_H$	INTSENT0SI	Status interrupt for RSENT0	RSENT0	Level	115B$_H$	✓	✓	✓	✓			+56C$_H$
348	ICSENT0RI	FFFF B2B8$_H$	INTSENT0RI	Receive interrupt for RSENT0	RSENT0	Edge	115C$_H$	✓	✓	✓	✓			+570$_H$

Table 7BC.6 EI Level Maskable Interrupt Sources (RH850/F1KM-S1)

Interrupt			Interrupt Request				Exception Source Code	48 Pins	64 Pins	80 Pins	100 Pins	Handler Address (Offset)[8]		
Channel No.[1]	Control Register					Detection Type[2]						Direct Jumping to an Address		Reference to a Table[5]
	Name	Address	Name	Source	Unit							RINT = 0	RINT = 1	
349	ICSENT1SI	FFFF B2BA$_H$	INTSENT1SI	Status interrupt for RSENT1	RSENT1	Level	115D$_H$	✓	✓	✓	✓	[3]	[4]	+574$_H$
350	ICSENT1RI	FFFF B2BC$_H$	INTSENT1RI	Receive interrupt for RSENT1	RSENT1	Edge	115E$_H$	✓	✓	✓	✓			+578$_H$
351	Reserved	FFFF B2BE$_H$					115F$_H$	—	—	—	—			+57C$_H$
352	Reserved	FFFF B2C0$_H$					1160$_H$	—	—	—	—			+580$_H$
353	Reserved	FFFF B2C2$_H$					1161$_H$	—	—	—	—			+584$_H$
354	Reserved	FFFF B2C4$_H$					1162$_H$	—	—	—	—			+588$_H$
355	Reserved	FFFF B2C6$_H$					1163$_H$	—	—	—	—			+58C$_H$
356	ICDPE	FFFF B2C8$_H$	INTDPE	LPS0 digital port error interrupt	LPS0	Level	1164$_H$	✓	✓	✓	✓			+590$_H$
357	ICAPE	FFFF B2CA$_H$	INTAPE	LPS0 analog port error interrupt	LPS0	Level	1165$_H$	✓	✓	✓	✓			+594$_H$

Note 1. Each interrupt is connected to INTC1 channel 8 to 31 and INTC2 channel 32 to 357.

Note 2. This indicates whether an interrupt source is detected at the level or edge. This also affects the value after reset of an EI level interrupt control register. For details, see **Section 7BC.4.2, ICxxx — EI Level Interrupt Control Registers**. For detection at level, an interrupt source is cleared by accessing to the register that retains an interrupt source. The procedure shown in **Section 3BC.3.1, Synchronization of Store Instruction Completion and Subsequent Instruction Execution** are required to reflect the result of the register update to the subsequent instruction.

Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100$_H$ to 1F0$_H$ according to the priority (0 to 15).

Note 4. Irrespective of the priority, offset addresses are uniformly +100$_H$.

Note 5. The table reference method uses a table for reading an exception handler address for each interrupt channel, and it extracts handler address by referencing that table. Table reference position is determined by the following formula.
Exception handler address read position = INTBP register + channel number × 4 bytes

Note 6. The same interrupt source is assigned to different interrupt channels. For details, see **Section 7BC.5.2, SELB_INTC1 — INTC1 Interrupt Select Register**.

Note 7. 32 interrupt sources are assigned to the same interrupt channel. For details, see **Section 37, PWM Output/Diagnostic (PWM-Diag)**.

Note 8. For details, see **Section 7BC.10, Exception Handler Address**.

Note 9. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it in both ways at the same time is not possible.
It is used as FEINT when OSTM0 functions as the TSU (timing supervision unit). It is used as EIINT when OSTM0 functions as anything other than the TSU function.

Note 10. For details on the interrupt source, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

7BC.2.2 FE Level Non-Maskable Interrupt Sources

7BC.2.2.1 List of Registers

The following table lists the FE Level Non-Maskable Interrupt Sources registers.

Table 7BC.7 List of Registers

Module Name	Register Name	Symbol	Address
ECON_NMI	FENMI factor register	WDTNMIF	FFC0 0000_H
	WDTNMI factor clear register	WDTNMIFC	FFC0 0008_H

7BC.2.2.2 WDTNMIF — FENMI Factor Register

This register contains information about which source has generated the FE level non-maskable interrupt (FENMI). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1 NMIF	WDTA0 NMIF	TNMIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7BC.8 WDTNMIF Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	WDTA1NMIF	WDTA1NMI flag 0: No WDTA1NMI occurred 1: WDTA1NMI has occurred
1	WDTA0NMIF	WDTA0NMI flag 0: No WDTA0NMI occurred 1: WDTA0NMI has occurred
0	TNMIF	Input signal flag from the NMI pin 0: No TNMI occurred 1: TNMI has occurred

7BC.2.2.3 WDTNMIFC — WDTNMI Factor Clear Register

This register clears the FE level non-maskable interrupt flags of the WDTNMIF register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0008$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1 NMIFC	WDTA0 NMIFC	TNMIF C
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Table 7BC.9 WDTNMIFC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	WDTA1NMIFC	WDTA1NMIF flag clear 0: — 1: Clear
1	WDTA0NMIFC	WDTA0NMIF flag clear 0: — 1: Clear
0	TNMIFC	TNMIF flag clear 0: — 1: Clear

7BC.2.3 FE Level Maskable Interrupt Sources

7BC.2.3.1 List of Registers

The following table lists the FE Level Maskable Interrupt Sources registers.

Table 7BC.10 List of Registers

Module Name	Register Name	Symbol	Address
ECON_FEINT	FEINT factor register	FEINTF	FFC0 0100_H
	FEINT factor mask register	FEINTFMSK	FFC0 0104_H
	FEINT factor clear register	FEINTFC	FFC0 0108_H

7BC.2.3.2 FEINTF — FEINT Factor Register

This register contains information about which source has generated the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0100_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIF	—	—	—	—	—	—	—	ECCETHFEIF	—	OSTM4FEIF	OSTM3FEIF	OSTM2FEIF	OSTM1FEIF	GUARDFEIF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIF	OSTM0FEIF	ECCRAMFEIF	ECCFLI0FEIF	—	ECCCSIH3FEIF	ECCCSIH2FEIF	ECCCSIH1FEIF	ECCCSIH0FEIF	ECCCNFDRAMFEIF	ECCFLRAMFEIF	ECCEEP0FEIF	—	—	—	LVILFEIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7BC.11 FEINTF Register Contents

Bit Position	Bit Name	Function
31	DMAFEIF	INTDMAERR interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
30 to 24	Reserved	When read, the value after reset is returned.
23	ECCETHFEIF*1	RH850/F1KM-S4: INTECCETH interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.
22	Reserved	When read, the value after reset is returned.
21	OSTM4FEIF	RH850/F1KM-S4: INTOSTM4_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.
20	OSTM3FEIF	RH850/F1KM-S4: INTOSTM3_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.

Table 7BC.11 FEINTF Register Contents

Bit Position	Bit Name	Function
19	OSTM2FEIF	RH850/F1KM-S4: INTOSTM2_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.
18	OSTM1FEIF	RH850/F1KM-S4: INTOSTM1_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.
17	GUARDFEIF	INTGUARD interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
16	Reserved	When read, the value after reset is returned.
15	LVIHFEIF	INTLVIH interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
14	OSTM0 FEIF	INTOSTM0_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
13	ECCRAM FEIF	INTECCRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
12	ECCFLI0 FEIF	INTECCFLI0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
11	Reserved	When read, the value after reset is returned.
10	ECCCSIH3 FEIF[*2]	INTECCCSIH3 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
9	ECCCSIH2 FEIF[*2]	INTECCCSIH2 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
8	ECCCSIH1 FEIF[*2]	INTECCCSIH1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
7	ECCCSIH0 FEIF	INTECCCSIH0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
6	ECCCNFDRAMFEIF	INTECCCNFDRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Table 7BC.11 FEINTF Register Contents

Bit Position	Bit Name	Function
5	ECCFLRAM FEIF	RH850/F1KM-S4: INTECCFLRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred RH850/F1KM-S1: When read, the value after reset is returned.
4	ECCEEP0 FEIF	INTECCEEP0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
3 to 1	Reserved	When read, the value after reset is returned.
0	LVILFEIF	INTLVIL interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Note 1. For the supported products, see **Table 7BC.3, FE Level Maskable Interrupt Requests (RH850/F1KM-S4)**.

Note 2. For the supported products, see **Table 7BC.4, FE Level Maskable Interrupt Requests (RH850/F1KM-S1)**.

7BC.2.3.3 FEINTFMSK — FEINT Factor Mask Register

This register masks the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFC0 0104_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIFMSK	—	—	—	—	—	—	—	ECCETHFEIFMSK	—	OSTM4FEIFMSK	OSTM3FEIFMSK	OSTM2FEIFMSK	OSTM1FEIFMSK	GUARDFEIFMSK	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIFMSK	OSTM0FEIFMSK	ECCRAMFEIFMSK	ECCFLI0FEIFMSK	—	ECCCSIH3FEIFMSK	ECCCSIH2FEIFMSK	ECCCSIH1FEIFMSK	ECCCSIH0FEIFMSK	ECCCNFDRAMFEIFMSK	ECCFLRAMFEIFMSK	ECCEEP0FEIFMSK	—	—	—	LVILFEIFMSK
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 7BC.12 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
31	DMAFEIFMSK	INTDMAERR interrupt mask 0: Not masked 1: Masked
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	ECCETH FEIFMSK*[1]	RH850/F1KM-S4: INTECCETH interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	OSTM4 FEIFMSK	RH850/F1KM-S4: INTOSTM4_FE interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
20	OSTM3 FEIFMSK	RH850/F1KM-S4: INTOSTM3_FE interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.

Table 7BC.12 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
19	OSTM2 FEIFMSK	RH850/F1KM-S4: INTOSTM2_FE interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
18	OSTM1 FEIFMSK	RH850/F1KM-S4: INTOSTM1_FE interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
17	GUARD FEIFMSK	INTGUARD interrupt mask 0: Not masked 1: Masked
16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	LVIHFEIFMSK	INTLVIH interrupt mask 0: Not masked 1: Masked
14	OSTM0FEI FMSK	INTOSTM0_FE interrupt mask 0: Not masked 1: Masked
13	ECCRAMFEI FMSK	INTECCRAM interrupt mask 0: Not masked 1: Masked
12	ECCFLI0 FEIFMSK	INTECCFLI0 interrupt mask 0: Not masked 1: Masked
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECCCSIH3 FEIFMSK*2	INTECCCSIH3 interrupt mask 0: Not masked 1: Masked
9	ECCCSIH2 FEIFMSK*2	INTECCCSIH2 interrupt mask 0: Not masked 1: Masked
8	ECCCSIH1 FEIFMSK*2	INTECCCSIH1 interrupt mask 0: Not masked 1: Masked
7	ECCCSIH0 FEIFMSK	INTECCCSIH0 interrupt mask 0: Not masked 1: Masked
6	ECCCNFDRAM FEIFMSK	INTECCCNFDRAM interrupt mask 0: Not masked 1: Masked
5	ECCFLRAM FEIFMSK	RH850/F1KM-S4: INTECCFLRAM interrupt mask 0: Not masked 1: Masked RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.

Table 7BC.12 FEINTFMSK Register Contents

Bit Position	Bit Name	Function
4	ECCEEP0 FEIFMSK	INTECCEEP0 interrupt mask 0: Not masked 1: Masked
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVILFEIFMSK	INTLVIL interrupt mask 0: Not masked 1: Masked

Note 1. For the supported products, see **Table 7BC.3, FE Level Maskable Interrupt Requests (RH850/F1KM-S4)**.

Note 2. For the supported products, see **Table 7BC.4, FE Level Maskable Interrupt Requests (RH850/F1KM-S1)**.

7BC.2.3.4 FEINTFC — FEINT Factor Clear Register

This register clears the bits of the FEINT factor register (FEINTF).

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0108_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAFEIFC	—	—	—	—	—	—	—	ECCETHFEIFC	—	OSTM4FEIFC	OSTM3FEIFC	OSTM2FEIFC	OSTM1FEIFC	GUARDFEIFC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	W	R	W	W	W	W	W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIFC	OSTM0FEIFC	ECCRAMFEIFC	ECCFLI0FEIFC	—	ECCCSIH3FEIFC	ECCCSIH2FEIFC	ECCCSIH1FEIFC	ECCCSIH0FEIFC	ECCCNFDRAMFEIFC	ECCFLRAMFEIFC	ECCEEP0FEIFC	—	—	—	LVILFEIFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	R	W	W	W	W	W	W	W	R	R	R	W

Table 7BC.13 FEINTFC Register Contents

Bit Position	Bit Name	Function
31	DMAFEIFC	DMAFEIF flag clear 0: — 1: Clear
30 to 24	Reserved	When writing, write the value after reset.
23	ECCETHFEIFC*1	RH850/F1KM-S4: ECCETHFEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.
22	Reserved	When writing, write the value after reset.
21	OSTM4FEIFC	RH850/F1KM-S4: OSTM4FEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.
20	OSTM3FEIFC	RH850/F1KM-S4: OSTM3FEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.
19	OSTM2FEIFC	RH850/F1KM-S4: OSTM2FEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.

Table 7BC.13 FEINTFC Register Contents

Bit Position	Bit Name	Function
18	OSTM1FEIFC	RH850/F1KM-S4: OSTM1FEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.
17	GUARDFEIFC	GUARDFEIF flag clear 0: — 1: Clear
16	Reserved	When writing, write the value after reset.
15	LVIHFEIFC	LVIHFEIF flag clear 0: — 1: Clear
14	OSTM0 FEIFC	OSTM0FEIF flag clear 0: — 1: Clear
13	ECCRAM FEIFC	ECCRAMFEIF flag clear 0: — 1: Clear
12	ECCFLI0FEIFC	ECCFLI0FEIF flag clear 0: — 1: Clear
11	Reserved	When writing, write the value after reset.
10	ECCCSIH3 FEIFC*2	ECCCSIH3FEIF flag clear 0: — 1: Clear
9	ECCCSIH2 FEIFC*2	ECCCSIH2FEIF flag clear 0: — 1: Clear
8	ECCCSIH1 FEIFC*2	ECCCSIH1FEIF flag clear 0: — 1: Clear
7	ECCCSIH0 FEIFC	ECCCSIH0FEIF flag clear 0: — 1: Clear
6	ECCCNFDRAM FEIFC	ECCCNFDRAMFEIF flag clear 0: — 1: Clear
5	ECCFLRAM FEIFC	RH850/F1KM-S4: ECCFLRAMFEIF flag clear 0: — 1: Clear RH850/F1KM-S1: When writing, write the value after reset.
4	ECCEEP0 FEIFC	ECCEEP0FEIF flag clear 0: — 1: Clear

Table 7BC.13 FEINTFC Register Contents

Bit Position	Bit Name	Function
3 to 1	Reserved	When writing, write the value after reset.
0	LVILFEIFC	LVILFEIF flag clear 0: — 1: Clear

Note 1. For the supported products, see **Table 7BC.3, FE Level Maskable Interrupt Requests (RH850/F1KM-S4)**.

Note 2. For the supported products, see **Table 7BC.4, FE Level Maskable Interrupt Requests (RH850/F1KM-S1)**.

7BC.3 Edge/Level Detection

External interrupts (TNMI and INTPm) can be specified to be generated when a rising edge, falling edge, rising or falling edge, low level, or high level is detected at an external interrupt pin.

The following registers are used to specify the edge and level of each interrupt:

Table 7BC.14 External Interrupt Edge/Level Detection Registers (RH850/F1KM-S4)

Interrupt	Register
TNMI	FCLA0CTL0_NMI
INTP0	FCLA0CTL0_INTPL
INTP1	FCLA0CTL1_INTPL
INTP2	FCLA0CTL2_INTPL
INTP3	FCLA0CTL3_INTPL
INTP4	FCLA0CTL4_INTPL
INTP5	FCLA0CTL5_INTPL
INTP6	FCLA0CTL6_INTPL
INTP7	FCLA0CTL7_INTPL
INTP8	FCLA0CTL0_INTPH
INTP9	FCLA0CTL1_INTPH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12	FCLA0CTL4_INTPH
INTP13	FCLA0CTL5_INTPH
INTP14*1	FCLA0CTL6_INTPH
INTP15*1	FCLA0CTL7_INTPH
INTP16*1	FCLA0CTL0_INTPU
INTP17*1	FCLA0CTL1_INTPU
INTP18*1	FCLA0CTL2_INTPU
INTP19*1	FCLA0CTL3_INTPU
INTP20*1	FCLA0CTL4_INTPU
INTP21*1	FCLA0CTL5_INTPU
INTP22*1	FCLA0CTL6_INTPU
INTP23*1	FCLA0CTL7_INTPU

Note 1. For the supported products, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**.

See **Section 2B, Pin Function of RH850/F1KM-S4** for details of these registers.

Table 7BC.15 External Interrupt Edge/Level Detection Registers (RH850/F1KM-S1)

Interrupt	Register
TNMI	FCLA0CTL0_NMI
INTP0	FCLA0CTL0_INTPL
INTP1	FCLA0CTL1_INTPL
INTP2	FCLA0CTL2_INTPL
INTP3	FCLA0CTL3_INTPL
INTP4	FCLA0CTL4_INTPL
INTP5	FCLA0CTL5_INTPL
INTP6*[1]	FCLA0CTL6_INTPL
INTP7*[1]	FCLA0CTL7_INTPL
INTP8*[1]	FCLA0CTL0_INTPH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12*[1]	FCLA0CTL4_INTPH
INTP13*[1]	FCLA0CTL5_INTPH

Note 1. For the supported products, see **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

See **Section 2C, Pin Function of RH850/F1KM-S1** for details of these registers.

7BC.4 Interrupt Controller Control Registers

Writing to the ICxxx, IMRm (m = 0 to 11), FNC, and FIC registers is enabled only in supervisor mode (PSW.UM = 0).

7BC.4.1 List of Registers

The following table lists the Interrupt Controller Control registers.

Table 7BC.16 List of Registers

Module Name	Register Name	Symbol	Address
INTC1	EI level interrupt control registers	ICxxx*[1]	See **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**
	EI level interrupt mask registers	IMR0	FFFE EAF0$_H$
INTC2	EI level interrupt control registers	ICxxx*[2]	See **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**
	EI level interrupt mask registers	IMRm	FFFF B400$_H$ + (04$_H$ × m)
INTC1	FE level NMI status register	FNC	FFFE EA78$_H$
	FE level maskable interrupt status register	FIC	FFFE EA7A$_H$

Note 1. Channel No.0 to 31.

Note 2. Channel No.32 or more.

7BC.4.2 ICxxx — EI Level Interrupt Control Registers

One of these registers is assigned to each EI level maskable interrupt (EIINT) channel and is used to set the conditions for controlling that channel. This register is initialized by any reset. For each source, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

CAUTION

If 0 is written to the RFxxx bit immediately after a peripheral module generates the corresponding interrupt request in edge detection mode (before an interrupt is accepted by the CPU), the request may be lost.

Writing 1 to the RFxxx bit immediately after an interrupt is accepted by the CPU may lead to re-issuing of the request.

Writing method to these registers contains the use of bit manipulation instructions (set1, clr1, and not1).

For bit-manipulation instructions, see also **Section 3BC.3.3, Access to Registers by Using Bit-Manipulation Instructions**.

Executing a bit-manipulation instruction to the lower bytes including the MKxxx bit has no effect on the RFxxx bit.

Access:	ICxxx can be read or written in 16-bit units. ICxxxH and ICxxxL can be read or written in 8- or 1-bit units. Access to bits 14, 13, 11 to 8, 5, and 4 by using a SET1, CLR1, or NOT1 instruction is prohibited.
Address:	See **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.
Value after reset:	$008F_H$ (edge detection), $808F_H$ (level detection)*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTxxx	—	—	RFxxx	—	—	—	—	MKxxx	TBxxx	—	—	P3xxx	P2xxx	P1xxx	P0xxx
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Note 1. The value after reset differs depending on the detection type of a given interrupt (edge detection: 0, level detection: 1). For details, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

Table 7BC.17 ICxxx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CTxxx</td><td>This bit indicates the type of interrupt detection. This bit is read only.
0: Edge detection
1: Level detection
When writing in 8-bit or 16-bit units, write the value after reset.</td></tr>
<tr><td>14, 13</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>12</td><td>RFxxx</td><td>This is an interrupt request flag.
The RFxxx bit can be written from a program. Setting the RFxxx bit to 1 generates an EI level maskable interrupt n (EIINTn), just as when an interrupt request is acknowledged.
0: No interrupt request is made.
1: Interrupt request is made.
<table>
<tr><th>Input Interface</th><th>Operation</th></tr>
<tr><td>Edge detection
(CTxxx = 0)</td><td>This bit is automatically cleared when an interrupt request is acknowledged by the CPU core. It can be set and cleared by software.</td></tr>
<tr><td>Level detection
(CTxxx = 1)</td><td>This bit cannot be set or cleared by software. It can only be read.
It is not cleared when an interrupt request is acknowledged by the CPU core.</td></tr>
</table>
</td></tr>
</table>

Table 7BC.17 ICxxx Register Contents

Bit Position	Bit Name	Function
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MKxxx	This is the interrupt request mask bit. When the MKxxx bit is set, interrupt requests from the channel are masked and are not issued to the CPU core. The interrupt pending status is not reflected in the ICSR.PMEI bit for any channels that are masked. When the interrupt request from the channel is masked with MKxxx = 1, the RFxxx still reflects the interrupt request for the channel and can be polled in software. When the MKxxx bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the MKxxx bit is also reflected in the corresponding IMRm register. 0: Enables interrupt processing 1: Disables interrupt processing
6	TBxxx	This bit is used to select the way to determine the interrupt vector. 0: Direct jumping to an address determined from the level of priority 1: Table reference For details on the way to determine the interrupt vector, see *the RH850G3KH User's Manual: Software*.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	P3xxx to P0xxx	These bits specify the interrupt priority as one of 16 levels, with 0 as the highest and 15 as the lowest. When multiple EI level-interrupt requests are made simultaneously, the interrupt from the source with the highest priority setting in these bits is selected and conveyed to the CPU core for servicing first. When P3xxx to P0xxx bits specify the same priority level for simultaneously occurring interrupt requests, the source with the lower channel number takes priority.

CAUTION

Do not access ICxxx registers of interrupt channels listed as “Reserved” in **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)** and of the channels which are not incorporated in the product.

7BC.4.3 IMRm — EI Level Interrupt Mask Registers (m = 0 to 11)

These registers are a collection of the MKxxx bits of the ICxxx registers. Each bit of IMRm reflects the setting of the corresponding MKxxx bit. The setting for IMRm is also reflected in the corresponding MKxxx bit. This register is initialized by any reset.

Access: IMRm can be read or written in 32-bit units.
IMRmH and IMRmL can be read or written in 16-bit units.
IMRmHH, IMRmHL, IMRmLH, and IMRmLL can be read or written in 8- or 1-bit units.

Address: IMR0: FFFE EAF0$_H$
IMR0L: FFFE EAF0$_{H,}$
IMR0H: FFFE EAF2$_H$
IMR0LH: FFFE EAF1$_{H,}$
IMR0HL: FFFE EAF2$_{H,}$
IMR0HH: FFFE EAF3$_H$
IMRm (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m)
IMRmL (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m)
IMRmH (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m) + 2$_{H,}$
IMRmLL (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m)
IMRmLH (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m) + 1$_{H,}$
IMRmHL (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m) + 2$_{H,}$
IMRmHH (m = 1 to 11): FFFF B400$_H$ + (04$_H$ × m) + 3$_H$

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMRmEIMK (m × 32 + 31)	IMRmEIMK (m × 32 + 30)	IMRmEIMK (m × 32 + 29)	IMRmEIMK (m × 32 + 28)	IMRmEIMK (m × 32 + 27)	IMRmEIMK (m × 32 + 26)	IMRmEIMK (m × 32 + 25)	IMRmEIMK (m × 32 + 24)	IMRmEIMK (m × 32 + 23)	IMRmEIMK (m × 32 + 22)	IMRmEIMK (m × 32 + 21)	IMRmEIMK (m × 32 + 20)	IMRmEIMK (m × 32 + 19)	IMRmEIMK (m × 32 + 18)	IMRmEIMK (m × 32 + 17)	IMRmEIMK (m × 32 + 16)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMRmEIMK (m × 32 + 15)	IMRmEIMK (m × 32 + 14)	IMRmEIMK (m × 32 + 13)	IMRmEIMK (m × 32 + 12)	IMRmEIMK (m × 32 + 11)	IMRmEIMK (m × 32 + 10)	IMRmEIMK (m × 32 + 9)	IMRmEIMK (m × 32 + 8)	IMRmEIMK (m × 32 + 7)	IMRmEIMK (m × 32 + 6)	IMRmEIMK (m × 32 + 5)	IMRmEIMK (m × 32 + 4)	IMRmEIMK (m × 32 + 3)	IMRmEIMK (m × 32 + 2)	IMRmEIMK (m × 32 + 1)	IMRmEIMK (m × 32 + 0)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7BC.18 IMRm Register Contents

Bit Position	Bit Name	Function
31 to 0	IMRmEIMK (m × 32 + 31) to IMRmEIMK (m × 32 + 0)	These are interrupt mask bits for EI level maskable interrupt (EIINT) channels 0 to 376. 0: Enables interrupt servicing 1: Disables interrupt servicing

CAUTION

MKxxx bits which correspond to channels listed as “Reserved” in **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)** and to channels which are not incorporated in the product must be set to “1”.

7BC.4.4 FNC — FE Level NMI Status Register

This register indicates the status of an FE level non-maskable interrupt (FENMI).

This register is initialized by any reset.

Access: FNC is a read-only register that can be read in 16-bit units.
FNCH is a read-only register that can be read in 8- or 1-bit units.

Address: FNC: FFFE EA78$_H$
FNCH: FFFE EA79$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FNRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7BC.19 FNC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FNRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit is automatically cleared when an FE level NMI interrupt request is acknowledged by the CPU core.
11 to 0	Reserved	When read, the value after reset is returned.

7BC.4.5 FIC — FE Level Maskable Interrupt Status Register

This register indicates the status of an FE level maskable interrupt (FEINT).

This register is initialized by any reset.

Access:	FIC is a read-only register that can be read in 16-bit units. FICH is a read-only register that can be read in 8- or 1-bit units.
Address:	FIC: FFFE EA7A$_{H}$ FICH: FFFE EA7B$_{H}$
Value after reset:	8000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FIRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7BC.20 FIC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FIRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit cannot be set or cleared by software. It can only be read.
11 to 0	Reserved	When read, the value after reset is returned.

7BC.5 EI Level Maskable Interrupt Select Register

The following registers are used to select an EI level maskable interrupt.

7BC.5.1 List of Registers

The following table lists the EI Level Maskable Interrupt Select register.

Table 7BC.21 List of Register

Module Name	Register Name	Symbol	Address
SL_INTC	INTC1 interrupt select register	SELB_INTC1	FFC0 1000_H

7BC.5.2 SELB_INTC1 — INTC1 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled. This register is initialized by any reset.

NOTE

The channel described in each bit setting indicates the channel of an interrupt and the priority. For details on channels, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

Access:	This register can be read or written in 16-bit units.
Address:	FFC0 1000_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SELB_INTC1_12	SELB_INTC1_11	SELB_INTC1_10	SELB_INTC1_9	SELB_INTC1_8	SELB_INTC1_7	SELB_INTC1_6	SELB_INTC1_5	SELB_INTC1_4	SELB_INTC1_3	SELB_INTC1_2	SELB_INTC1_1	SELB_INTC1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7BC.22 SELB_INTC1 Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SELB_INTC1_12	Interrupt channel selection 0: INTADCA0I2 (Channel 20) INTCSIH0IJC (Channel 32) 1: INTCSIH0IJC_1 (Channel 20) INTADCA0I2_2 (Channel 32)
11	SELB_INTC1_11	Interrupt channel selection 0: INTTAUD0I14 (Channel 15) INTCSIH3IJC (Channel 161) 1: INTCSIH3IJC_1 (Channel 15) INTTAUD0I14_2 (Channel 161)
10	SELB_INTC1_10	Interrupt channel selection 0: INTTAUD0I12 (Channel 14) INTCSIH3IRE (Channel 160) 1: INTCSIH3IRE_1 (Channel 14) INTTAUD0I12_2 (Channel 160)
9	SELB_INTC1_9	Interrupt channel selection 0: INTTAUD0I10 (Channel 13) INTCSIH3IR (Channel 159) 1: INTCSIH3IR_1 (Channel 13) INTTAUD0I10_2 (Channel 159)
8	SELB_INTC1_8	Interrupt channel selection 0: INTTAUD0I2 (Channel 9) INTCSIH3IC (Channel 158) 1: INTCSIH3IC_1 (Channel 9) INTTAUD0I2_2 (Channel 158)

Table 7BC.22 SELB_INTC1 Register Contents

Bit Position	Bit Name	Function
7	SELB_INTC1_7	Interrupt channel selection 0: INTTAUD0I8 (Channel 12) INTCSIH2IJC (Channel 135) 1: INTCSIH2IJC_1 (Channel 12) INTTAUD0I8_2 (Channel 135)
6	SELB_INTC1_6	Interrupt channel selection 0: INTTAUD0I6 (Channel 11) INTCSIH2IRE (Channel 134) 1: INTCSIH2IRE_1 (Channel 11) INTTAUD0I6_2 (Channel 134)
5	SELB_INTC1_5	Interrupt channel selection 0: INTTAUD0I4 (Channel 10) INTCSIH2IR (Channel 133) 1: INTCSIH2IR_1 (Channel 10) INTTAUD0I4_2 (Channel 133)
4	SELB_INTC1_4	Interrupt channel selection 0: INTTAUD0I0 (Channel 8) INTCSIH2IC (Channel 132) 1: INTCSIH2IC_1 (Channel 8) INTTAUD0I0_2 (Channel 132)
3	SELB_INTC1_3	Interrupt channel selection 0: INTCSIG0IR (Channel 28) INTCSIH1IJC (Channel 119) 1: INTCSIH1IJC_1 (Channel 28) INTCSIG0IR_2 (Channel 119)
2	SELB_INTC1_2	Interrupt channel selection 0: INTCSIG0IC (Channel 27) INTCSIH1IRE (Channel 118) 1: INTCSIH1IRE_1 (Channel 27) INTCSIG0IC_2 (Channel 118)
1	SELB_INTC1_1	Interrupt channel selection 0: INTTAPA0IVLY0 (Channel 17) INTCSIH1IR (Channel 117) 1: INTCSIH1IR_1 (Channel 17) INTTAPA0IVLY0_2 (Channel 117)
0	SELB_INTC1_0	Interrupt channel selection 0: INTTAPA0IPEK0 (Channel 16) INTCSIH1IC (Channel 116) 1: INTCSIH1IC_1 (Channel 16) INTTAPA0IPEK0_2 (Channel 116)

NOTE

For the supported products, see **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)**, **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)**.

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB_INTC1.

7BC.6 Interrupt Function System Registers

See **Table 3BC.31, Interrupt Function System Registers**.

7BC.6.1 FPIPR — FPI Exception Interrupt Priority

See **Table 3BC.32, FPIPR Register Contents**.

7BC.6.2 ISPR — Priority of Interrupt being Serviced

See **Table 3BC.33, ISPR Register Contents**.

7BC.6.3 PMR — Interrupt Priority Masking

See **Table 3BC.34, PMR Register Contents**.

7BC.6.4 ICSR — Interrupt Control Status

See **Table 3BC.35, ICSR Register Contents**.

7BC.6.5 INTCFG — Interrupt Function Setting

See **Table 3BC.36, INTCFG Register Contents**.

7BC.7 Operation when Acknowledging an Interrupt

Check whether each interrupt that is reported during instruction execution is acknowledged according to the priority. The procedure for acknowledging each interrupt is shown below.

(1) Check whether the acknowledgment conditions are satisfied and whether interrupts are acknowledged according to their priority.

(2) Calculate the exception handler address according to the current PSW value.*[1]

(3) For FE-level non-maskable/maskable interrupts, the following processing is performed:

- Save the PC to the FEPC.
- Save the PSW to the FEPSW.
- Store the exception source code in the FEIC.
- Update the PSW.*[2]
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

(4) For EI level exceptions, the following processing is performed:

- Save the PC to the EIPC.
- Save the PSW to the EIPSW.
- Store the exception source code in the EIIC.
- Update the PSW.*[2]
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

Note 1. For details, see **Section 7BC.10, Exception Handler Address**.

Note 2. For the values to be updated, see *Table 4.1 Exception Cause List* in *the RH850G3KH User's Manual: Software*.

The following figure shows steps (1) to (4).

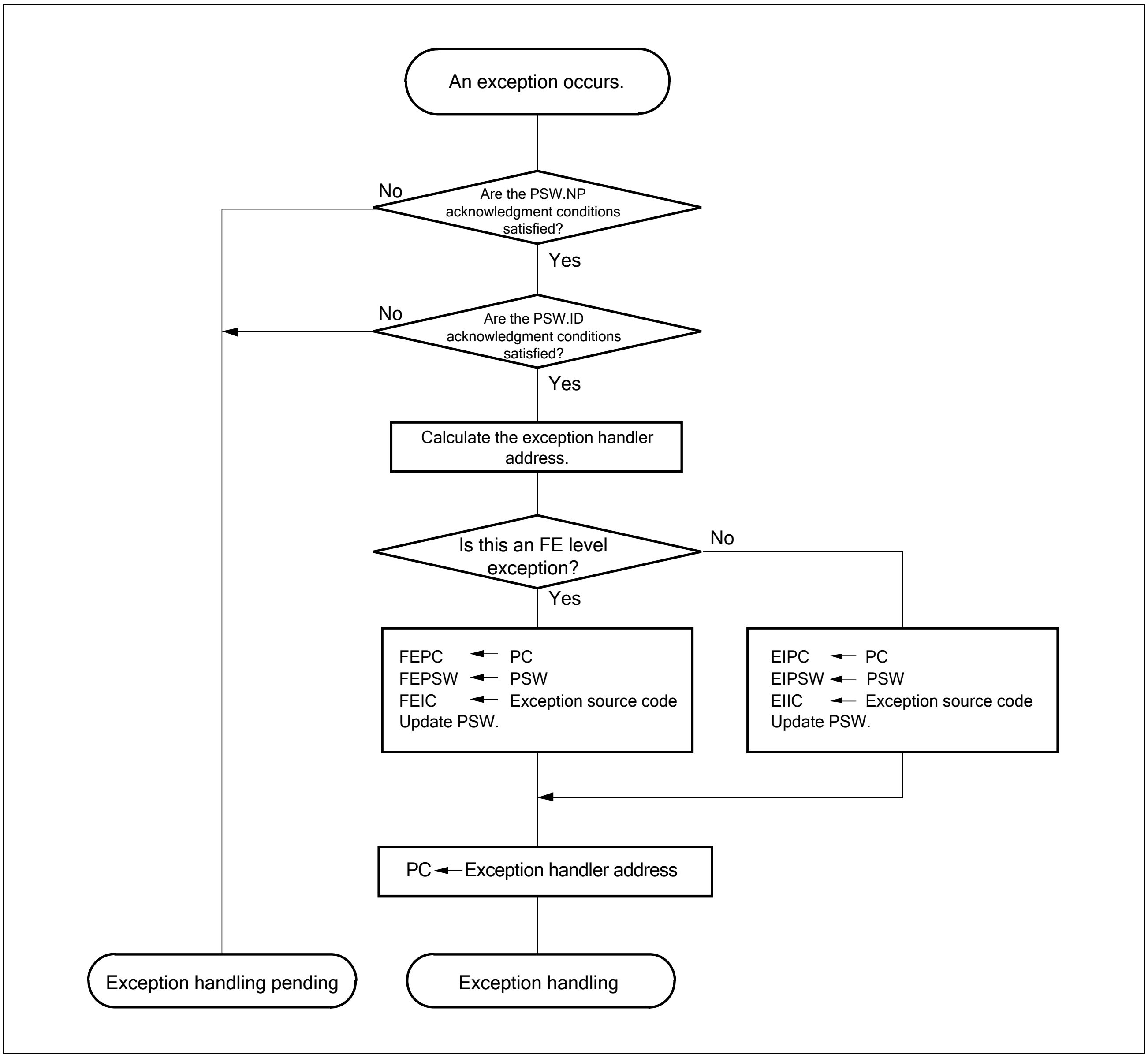


Figure 7BC.2 Operation when Acknowledging an Interrupt

7BC.7.1 Exception Source Codes for Different Types of SYSERR Exceptions

The following table lists exception source codes for the different types of SYSERR exceptions.

Table 7BC.23 Exception Source Codes for Different Types of SYSERR Exceptions

Exception Source Code	Source of SYSERR Generation
11_H	• Detection of an error during the fetching of an instruction from the code flash memory area
13_H	• Detection of an error during the fetching of an instruction from the local, global or retention RAM areas
14_H	• Detection of an error during access to data in the code flash area • Detection of an error during read access to a module via the system interconnect or PBUS The exception source code reports an occurrence of a system error factor corresponding to VCIF bit of SEGFLAG register*1
16_H	• Detection of an error during access to data in the own local RAM areas The exception source code reports an occurrence of a system error factor corresponding to TCMF bit of SEGFLAG register*1
18_H	• Detection of an IPG error The exception source code reports an occurrence of a system error factor corresponding to VCIF, IPGF bit of SEGFLAG register*1
19_H	• Detection of an error during write access to a module via the PBUS The exception source code reports an occurrence of a system error factor corresponding to APIF bit of SEGFLAG register*1
$1A_H$	• Detection of an error inside system interconnect The exception source code reports an occurrence of a system error factor corresponding to VCSF bit of SEGFLAG register*1

Note 1. See **Section 3BC.2.3.3, System Error Generator Function (SEG)** for details.

7BC.8 Return from Interrupts

To return from interrupt handling, execute the return instruction (EIRET or FERET) corresponding to each relevant interrupt level.

When a context has been saved in a stack and the like, the context must be restored before executing the return instruction.

The EIRET instruction is used to return from the EI level maskable interrupt handling and the FERET instruction is used to return from FE-level maskable interrupt handling.

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes its control to the return PC address:

(1) When returning from the service routine for an EI-level exception, the PC and PSW values on return are loaded from the EIPC and EIPSW registers.
When returning from the service routine for an FE-level exception, the PC and PSW values on return are loaded from the FEPC and FEPSW registers.

(2) Control is passed to the addresses indicated by the return PC and PSW that were loaded.

(3) When EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.

The flows for returning from exception handling using the EIRET and FERET instructions are shown below.

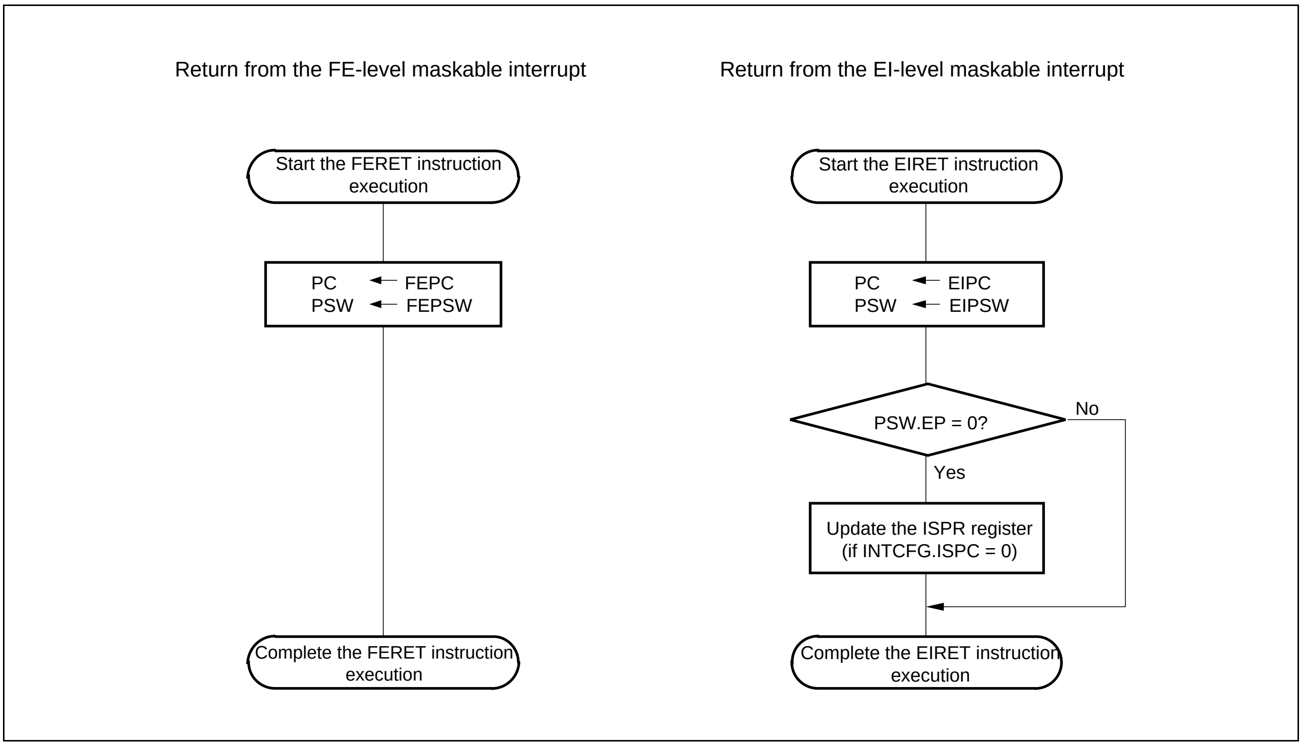


Figure 7BC.3 Flow of Return from Interrupts

7BC.9 Interrupt Operation

7BC.9.1 Interrupt Mask Function of EI Level Maskable Interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by the following register settings.

Table 7BC.24 Operation of the MKxxx Bit

ICxxx.MKxxx	Operation
1	Masks interrupt
0	Enables interrupt

The ICxxx.MKxxx bits can also be read and written via the corresponding IMRmEIMKn bits of the IMRm registers. The interrupt mask state is reflected in both the ICxxx registers and the IMRm registers.

[Operation example]

(1) When a 1 is written to an IMRm.IMRmEIMKn bit, interrupts are prohibited for the corresponding channel.

(2) When the corresponding ICxxx.MKxxx bit is read, 1 is returned.

CAUTION

If the MKxxx bit is set to 0 while an interrupt request is pending (RFxxx = 1), the interrupt service routine will be executed at that time (subject to the rules of interrupt prioritization). Even if an interrupt request is issued in software by setting the RFxxx bit to 1, the interrupt will not occur as long as the interrupt is masked with MKxxx = 1.
To cancel an interrupt request that is pending, clear the corresponding RFxxx bit in software.

7BC.9.2 Interrupt Priority Level Judgment

When FE level non-maskable interrupts (FENMI), FE level maskable interrupts (FEINT), and EI level maskable interrupts (EIINT) are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is processed. Exceptions occurred at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

FENMI > FEINT > EIINT

See *the RH850G3KH User's Manual: Software* for other exceptions.

For EIINT (INTn) interrupts, the interrupt priority can be set independently for each interrupt source. Specify the interrupt priority with the bits P3xxx to P0xxx. The interrupt priority levels can be set from 0 to 15: 0 is the highest and 15 is the lowest. Among multiple EIINT (INTn) interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

Table 7BC.25 Example of EIINT (INTn) Interrupt Priority Level Settings and Priority Levels during Operation

EIINT (INTn)	ICxxx.P[3:0]xxx Setting	Priority Level During Operation
INT0	3	10
INT1	4	11
INT2	0	1
INT3	0	2
INT4	1	3
INT5	2	6
INT6	2	7
INT7	1	4
INT8	1	5
INT9	2	8
INT10	2	9

The interrupt controller executes multiple interrupt handling when another interrupt request is acknowledged while an interrupt processing is being executed. When multiple EIINT (INTn) interrupts occur at the same time, the interrupt to be acknowledged is determined by the following procedure.

7BC.9.2.1 Comparison with the Priority Level of the Interrupt Currently being Handled

Interrupts with the same or lower priority level as the interrupt currently being handled are held pending.

The priority level of the interrupt currently being handled is stored in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being handled proceed to the next priority judgment stage.

7BC.9.2.2 Masking through Priority Mask Register (PMR)

Only interrupts enabled by the PMR register proceed to the next priority judgment stage.

For the PMR register, see **Table 3BC.34, PMR Register Contents**, or *the RH850G3KH User's Manual: Software*.

7BC.9.2.3 The Requested Interrupt Source with the Highest Priority Level is Selected

When interrupts are requested simultaneously from multiple sources, the interrupt set the highest priority by ICxxx.P[3:0]xxx bits takes priority.

When there are multiple highest priority interrupts, the lowest interrupt channel number is selected.

7BC.9.2.4 Interrupt Suspended by CPU

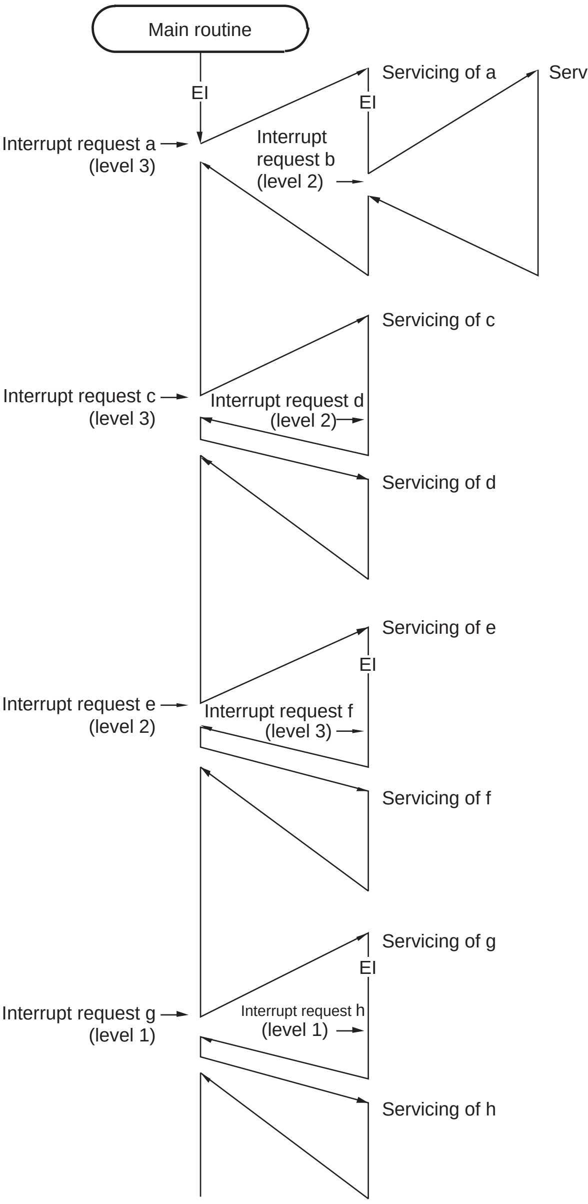
Interrupt acknowledgment is held according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT, FEINT and FENMI interrupts are performed even while interrupt acknowledgment is pending, and the interrupt with the highest priority is selected when the acknowledgment condition is satisfied.

Example

An EIINT interrupt with the priority level 5 has already been requested and interrupt generation is pending because the value of the PSW.ID bit is 1. If a subsequent EIINT interrupt with the priority level 3 is requested and the PSW.ID bit is cleared to 0, the latter EIINT interrupt (with the priority level 3) will be generated.

Figure 7BC.4, Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1) shows an example of multiple interrupt handling when another interrupt request is acknowledged while interrupt processing is being executed.

When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, the ID flag should be cleared to 0 to execute multiple interrupt handling. Specifically, execute the EI instruction and the like in an interrupt handling program to enable the interrupt.



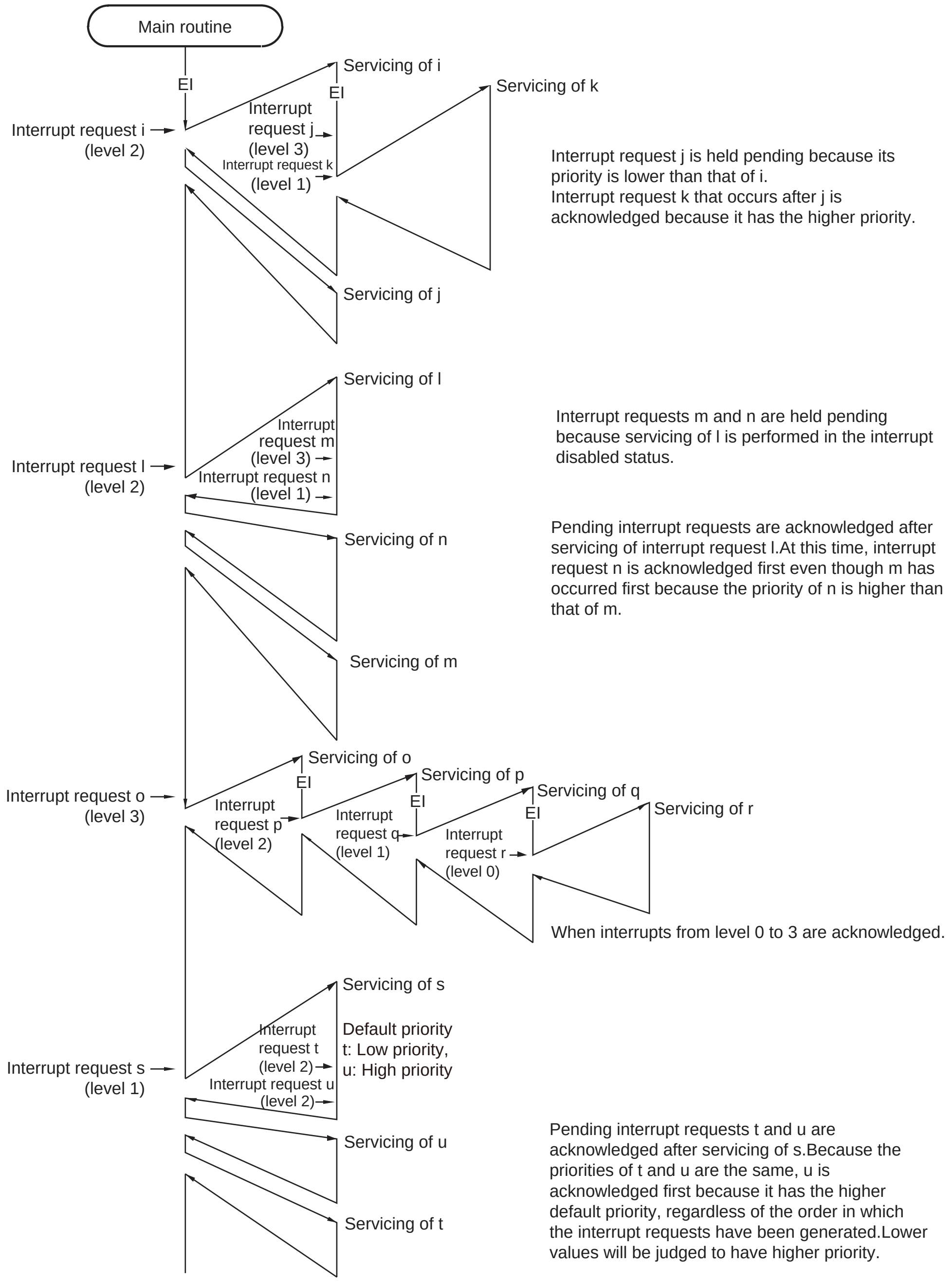
Interrupt request b is acknowledged because the priority of b is higher than that of a and interrupts are enabled.

Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.

Interrupt request f is held pending even if interrupts are enabled because its priority of f is lower than that of e.

Interrupt request h is held pending even if interrupts are enabled because its priority level is the same as that of g.

Figure 7BC.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1)



Interrupt request j is held pending because its priority is lower than that of i. Interrupt request k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because servicing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after servicing of interrupt request l.At this time, interrupt request n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

When interrupts from level 0 to 3 are acknowledged.

Pending interrupt requests t and u are acknowledged after servicing of s.Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.Lower values will be judged to have higher priority.

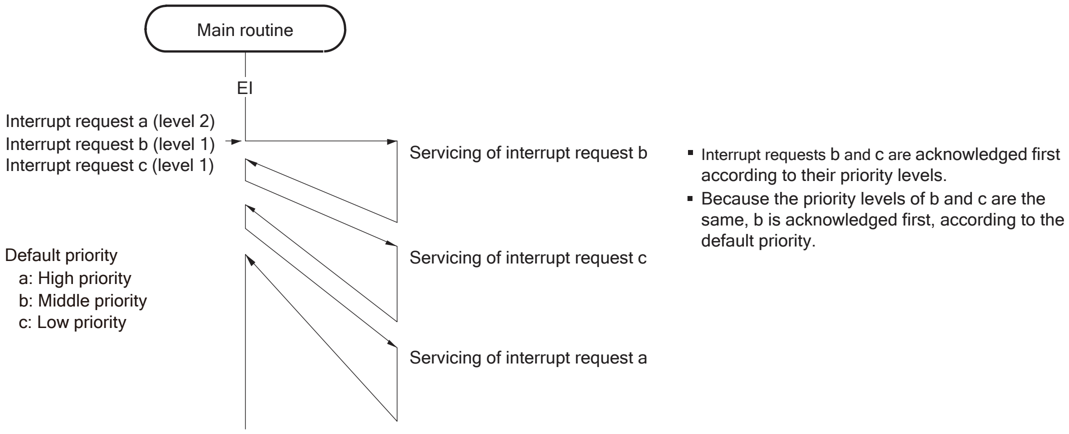
Note 1. “a” to “u” in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7BC.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (2)

CAUTION

To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.



Note 1. "a" to "c" in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7BC.5 Example of Servicing Simultaneously Generated Interrupt Requests

7BC.9.3 Interrupt Request Acknowledgement Conditions and the Priority

See *the RH850G3KH User's Manual: Software*.

7BC.9.4 Exception Priority of Interrupts and the Priority Mask

See *the RH850G3KH User's Manual: Software*.

7BC.9.5 Interrupt Priority Mask

See *the RH850G3KH User's Manual: Software*.

7BC.9.6 Priority Mask Function

The priority mask function prohibits all EIINT interrupts of the specified interrupt priority level.

The interrupt priority levels to be masked are specified in the PMR register. Masking and acknowledgment can be set for each priority level.

The following operations are possible using this function:

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

Table 7BC.26 Operation of the PMR.PMm Bit

PMR.PMm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note: m = 0 to 15

The presence of EIINT interrupts held pending with this function can be checked with **Section 7BC.9.7, Exception Management**.

For details on the PMR register, see **Table 3BC.34, PMR Register Contents**, or *the RH850G3KH User's Manual: Software*.

7BC.9.7 Exception Management

Pending interrupts can be checked in the RH850/F1KM. For details, see *the RH850G3KH User's Manual: Software*.

7BC.10 Exception Handler Address

In the RH850/F1KM, the exception handler address from which the handler is executed after a reset is input or when an exception or interrupt is acknowledged can be changed according to a setting.

The exception handler address for resets and exceptions (including interrupts) is determined with the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, the RBASE register, and the EBASE register. For interrupts, the direct vector method and table reference method can be selected for each channel. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory.

CAUTION

The exception handler address of EIINT (INTn) selected using the direct vector method differs from that of the V850E2 core products. In the V850E2 core products, a different exception handler address is individually assigned to each interrupt channel (EIINT (INTn)). In the RH850/F1KM, one exception handler address is assigned to each interrupt priority. Consequently, interrupts that have the same priority level branch to the same exception handler.

7BC.10.1 Direct Vector Method

The CPU uses the result of adding the offset shown in **Table 7BC.27, Selection of Base Register/Offset Address** to the base address indicated by the RBASE or EBASE register as the exception handler address.

Select whether the RBASE or EBASE register is used as the base address by using the PSW.EBV bit*[1]. When the PSW.EBV bit is set to 1, the value of the EBASE register is used as the base address. When the PSW.EBV bit is cleared to 0, the value of the RBASE register is used as the base address.

For reset input the RBASE register is always used for reference.

In addition, user interrupts see the RINT bit of the corresponding base register, and reduce the offset address according to the bit status. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, all user interrupts are handled using an offset of 100_H. If the bit is cleared to 0, the offset address is determined according to **Table 7BC.27, Selection of Base Register/Offset Address**.

Note 1. Exception acknowledgment itself may sometimes update the status of the PSW.EBV bit. In this case, the base register is selected based on the updated value.

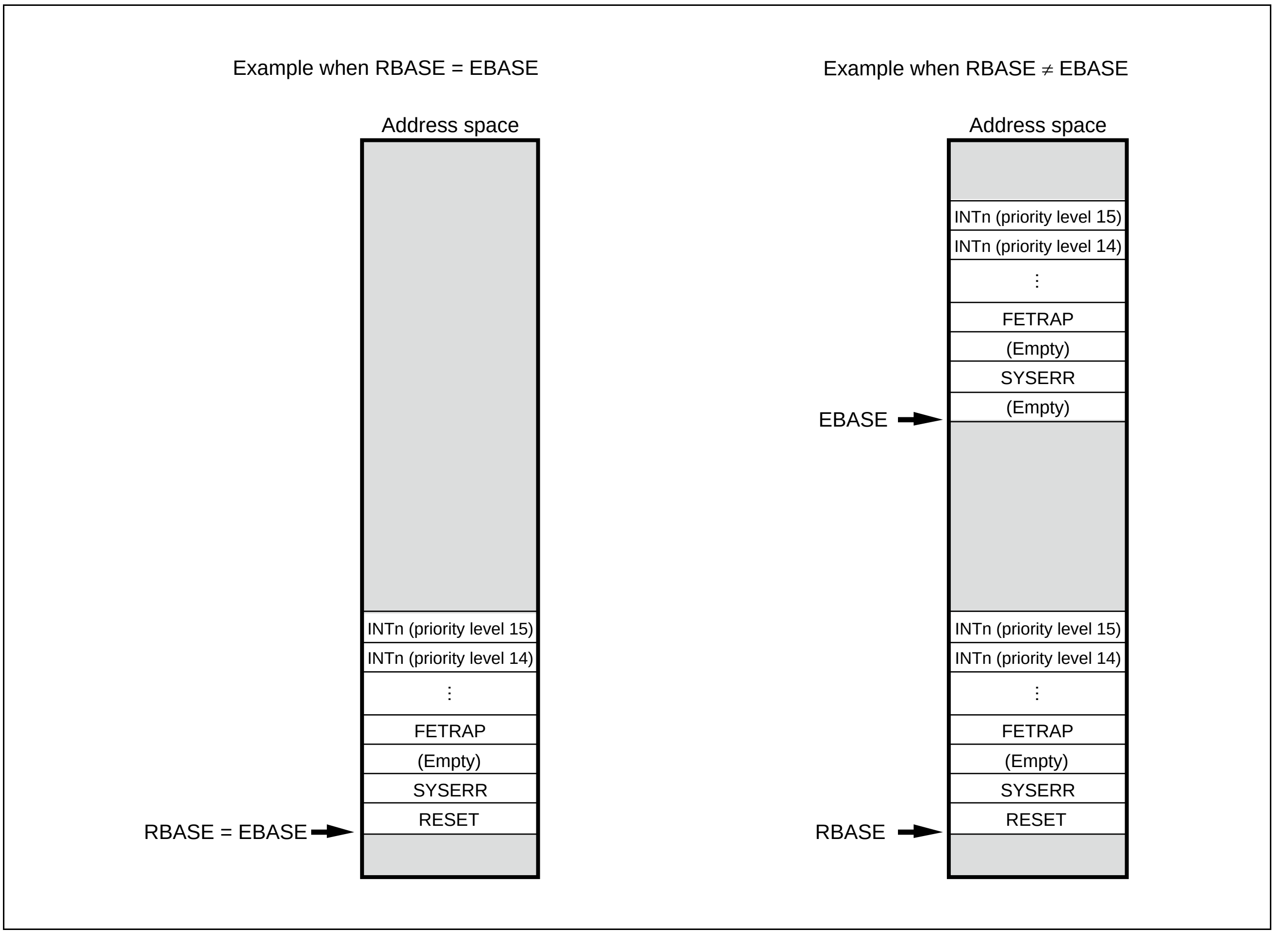


Figure 7BC.6 Direct Vector Method

The table below shows how base register selection and offset address reduction function for each exception to determine the exception handler address. The value of the PSW bit determines the exception handler address on the basis of the value updated by the acknowledgment of an exception.

Table 7BC.27 Selection of Base Register/Offset Address

Exception/Interrupt	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
	Base Register		Offset Address	
RESET	RBASE	N.A.	000_H	000_H
SYSERR		EBASE	010_H	010_H
Reserved			020_H	020_H
FETRAP			030_H	030_H
TRAP0			040_H	040_H
TRAP1			050_H	050_H
RIE			060_H	060_H
FPP/FPI			070_H	070_H
UCPOP			080_H	080_H
MIP/MDP			090_H	090_H
PIE			$0A0_H$	$0A0_H$
MAE			$0C0_H$	$0C0_H$
Reserved			$0D0_H$	$0D0_H$
FENMI			$0E0_H$	$0E0_H$
FEINT			$0F0_H$	$0F0_H$
INTn (Priority level 0)			100_H	100_H
INTn (Priority level 1)			110_H	
INTn (Priority level 2)			120_H	
INTn (Priority level 3)			130_H	
INTn (Priority level 4)			140_H	
INTn (Priority level 5)			150_H	
INTn (Priority level 6)			160_H	
INTn (Priority level 7)			170_H	
INTn (Priority level 8)			180_H	
INTn (Priority level 9)			190_H	
INTn (Priority level 10)			$1A0_H$	
INTn (Priority level 11)			$1B0_H$	
INTn (Priority level 12)			$1C0_H$	
INTn (Priority level 13)			$1D0_H$	
INTn (Priority level 14)			$1E0_H$	
INTn (Priority level 15)			$1F0_H$	

Base register selection is used to execute exception handling for resets and some hardware errors by using the programs in a relatively reliable area such as ROM instead of the areas that are easily affected by software errors such as RAM and cache area. The user interrupt offset address reduction function is used to reduce the memory occupation size required by the exception handler for specific system-internal operating modes. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

7BC.10.2 Table Reference Method

With the direct vector method, there is one user-interrupt exception handler for each interrupt priority level, and interrupt channels that indicate multiple interrupts with the same priority branch to the same interrupt handler, but some users might want to use different code areas for each interrupt handler from the beginning.

The RH850/F1KM uses the table reference method for interrupts that assume the above usage.

If the table reference method is specified as the interrupt channel vector selection method in the interrupt controller and the like, the method for determining the exception handler address when an interrupt request corresponding to that interrupt channel is acknowledged differs as follows:

<1> In any of the following cases, the exception handler address is determined by using the direct vector method:

  - When PSW.EBV = 0 and RBASE.RINT = 1
  - When PSW.EBV = 1 and EBASE.RINT = 1
  - When the interrupt channel setting is not the table reference method

<2> In cases other than <1>, calculate the table reference position.
Exception handler address read position = INTBP register + channel number × 4 bytes

<3> Read word data starting at the interrupt handler address read position calculated in <2>.

<4> Use the word data read in <3> as the exception handler address.

Table 7BC.28, Exception Handler Address Expansion shows the exception handler address read positions corresponding to each interrupt channel and
Figure 7BC.7, Table Reference Method shows an overview of the allocation in memory.

Table 7BC.28 Exception Handler Address Expansion

Type of Interrupt	Exception Handler Address Read Position
EI level maskable interrupt channel 0	INTBP register value + 0 × 4
EI level maskable interrupt channel 1	INTBP register value + 1 × 4
EI level maskable interrupt channel 2	INTBP register value + 2 × 4
:	:
EI level maskable interrupt channel 375	INTBP register value + 375 × 4
EI level maskable interrupt channel 376	INTBP register value + 376 × 4

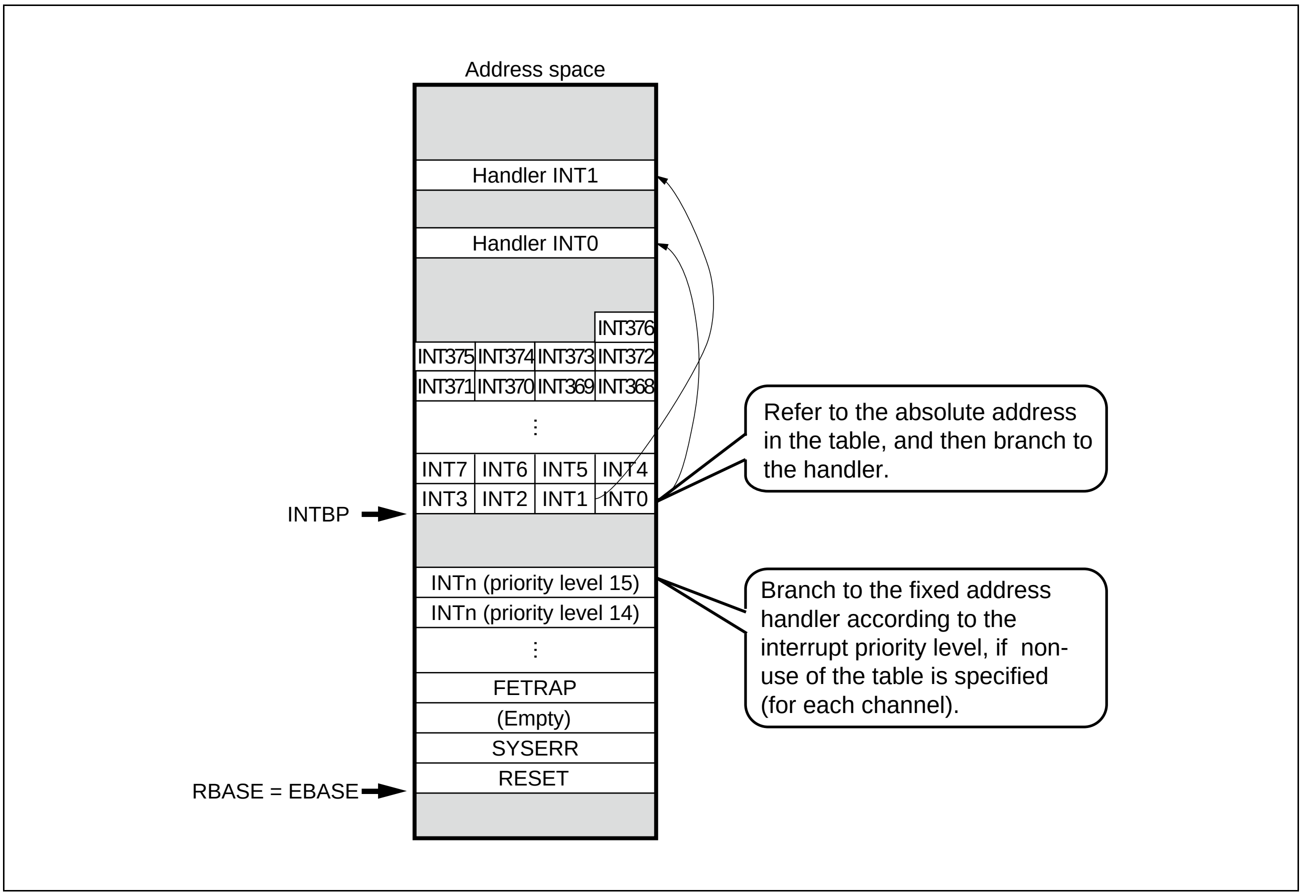


Figure 7BC.7 Table Reference Method

Section 8 DMA Controller

This section contains a generic description of the DMA controller (DMA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of channels, register base addresses, etc.

The remainder of the section describes the functions and registers of DMA.

8.1 Features of RH850/F1KH, RH850/F1KM DMA Controller

8.1.1 Number of Channels

This microcontroller has the following number of DMA channels.

Table 8.1 Number of Channels (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Channels	64 ch (8 ch × 4 × 2)	64 ch (8 ch × 4 × 2)	64 ch (8 ch × 4 × 2)

Table 8.2 Number of Channels (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of channels	32 ch (8 ch × 4)	32 ch (8 ch × 4)	32 ch (8 ch × 4)	32 ch (8 ch × 4)	32 ch (8 ch × 4)

Table 8.3 Number of Channels (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of channels	16 ch (8 ch × 2)	16 ch (8 ch × 2)	16 ch (8 ch × 2)	16 ch (8 ch × 2)

Table 8.4 Indices (RH850/F1KH-D8)

Index	Description
n	In this section, the DMA units are identified with an index “n” for example, as PDMAn (n = 0, 1).
y	In this section, the DMA groups are identified with an index “y” for example, as DMACy (y = 0 to 3).
m	In this section, the DMA channels are identified with an index “m”. For example, PDMAnDSAm (m = 0 to 31) is the DMA source address register.
i	In this section, the DMAC channels are identified with an index “i”. For example, PDMAnDMyiCM (i = 0 to 7) is the DMAC channel master.

Table 8.5 Indices (RH850/F1KM-S4)

Index	Description
n	In this section, the DMA units are identified with an index “n” for example, as PDMAn (n = 0).
y	In this section, the DMA groups are identified with an index “y” for example, as DMACy (y = 0 to 3).
m	In this section, the DMA channels are identified with an index “m”. For example, PDMAnDSAm (m = 0 to 31) is the DMA source address register.
i	In this section, the DMAC channels are identified with an index “i”. For example, PDMAnDMyiCM (i = 0 to 7) is the DMAC channel master.

Table 8.6 Indices (RH850/F1KM-S1)

Index	Description
n	In this section, the DMA units are identified with an index “n” for example, as PDMAn (n = 0).
y	In this section, the DMA groups are identified with an index “y” for example, as DMACy (y = 0, 1).
m	In this section, the DMA channels are identified with an index “m”. For example, PDMAnDSAm (m = 0 to 15) is the DMA source address register.
i	In this section, the DMAC channels are identified with an index “i”. For example, PDMAnDMyiCM (i = 0 to 7) is the DMAC channel master.

8.1.2 Register Base Addresses

The DMA base addresses are listed in the following table.

The DMA register addresses are given as offsets from the base addresses.

Table 8.7 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<DMA0_base>	FFFF 8000_H
<DMA1_base>	FFFF 9000_H

Table 8.8 Register Base Address (RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<DMA0_base>	FFFF 8000_H

8.1.3 Interrupt Requests

The DMA interrupt requests are listed in the following table.

Table 8.9 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number
PDMA0		
INTDMA0	PDMA0 DMA00 transfer completion	60
INTDMA1	PDMA0 DMA01 transfer completion	61
INTDMA2	PDMA0 DMA02 transfer completion	62
INTDMA3	PDMA0 DMA03 transfer completion	63
INTDMA4	PDMA0 DMA04 transfer completion	64
INTDMA5	PDMA0 DMA05 transfer completion	65
INTDMA6	PDMA0 DMA06 transfer completion	66
INTDMA7	PDMA0 DMA07 transfer completion	67
INTDMA8	PDMA0 DMA10 transfer completion	68
INTDMA9	PDMA0 DMA11 transfer completion	69
INTDMA10	PDMA0 DMA12 transfer completion	70
INTDMA11	PDMA0 DMA13 transfer completion	71
INTDMA12	PDMA0 DMA14 transfer completion	72
INTDMA13	PDMA0 DMA15 transfer completion	73
INTDMA14	PDMA0 DMA16 transfer completion	74
INTDMA15	PDMA0 DMA17 transfer completion	75
INTDMA16	PDMA0 DMA20 transfer completion	298
INTDMA17	PDMA0 DMA21 transfer completion	299
INTDMA18	PDMA0 DMA22 transfer completion	300
INTDMA19	PDMA0 DMA23 transfer completion	301
INTDMA20	PDMA0 DMA24 transfer completion	302
INTDMA21	PDMA0 DMA25 transfer completion	303
INTDMA22	PDMA0 DMA26 transfer completion	304
INTDMA23	PDMA0 DMA27 transfer completion	305
INTDMA24	PDMA0 DMA30 transfer completion	306
INTDMA25	PDMA0 DMA31 transfer completion	307
INTDMA26	PDMA0 DMA32 transfer completion	308
INTDMA27	PDMA0 DMA33 transfer completion	309
INTDMA28	PDMA0 DMA34 transfer completion	310
INTDMA29	PDMA0 DMA35 transfer completion	311
INTDMA30	PDMA0 DMA36 transfer completion	312
INTDMA31	PDMA0 DMA37 transfer completion	313

Table 8.9 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number
PDMA1		
INTDMA32	PDMA1 DMA00 transfer completion	101
INTDMA33	PDMA1 DMA01 transfer completion	102
INTDMA34	PDMA1 DMA02 transfer completion	103
INTDMA35	PDMA1 DMA03 transfer completion	106
INTDMA36	PDMA1 DMA04 transfer completion	188
INTDMA37	PDMA1 DMA05 transfer completion	189
INTDMA38	PDMA1 DMA06 transfer completion	190
INTDMA39	PDMA1 DMA07 transfer completion	191
INTDMA40	PDMA1 DMA10 transfer completion	192
INTDMA41	PDMA1 DMA11 transfer completion	193
INTDMA42	PDMA1 DMA12 transfer completion	194
INTDMA43	PDMA1 DMA13 transfer completion	195
INTDMA44	PDMA1 DMA14 transfer completion	196
INTDMA45	PDMA1 DMA15 transfer completion	197
INTDMA46	PDMA1 DMA16 transfer completion	198
INTDMA47	PDMA1 DMA17 transfer completion	199
INTDMA48	PDMA1 DMA20 transfer completion	200
INTDMA49	PDMA1 DMA21 transfer completion	201
INTDMA50	PDMA1 DMA22 transfer completion	202
INTDMA51	PDMA1 DMA23 transfer completion	203
INTDMA52	PDMA1 DMA24 transfer completion	204
INTDMA53	PDMA1 DMA25 transfer completion	216
INTDMA54	PDMA1 DMA26 transfer completion	290
INTDMA55	PDMA1 DMA27 transfer completion	291
INTDMA56	PDMA1 DMA30 transfer completion	292
INTDMA57	PDMA1 DMA31 transfer completion	293
INTDMA58	PDMA1 DMA32 transfer completion	294
INTDMA59	PDMA1 DMA33 transfer completion	295
INTDMA60	PDMA1 DMA34 transfer completion	296
INTDMA61	PDMA1 DMA35 transfer completion	297
INTDMA62	PDMA1 DMA36 transfer completion	355
INTDMA63	PDMA1 DMA37 transfer completion	381

Table 8.10 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number
PDMA0		
INTDMA0	PDMA0 DMA00 transfer completion	60
INTDMA1	PDMA0 DMA01 transfer completion	61
INTDMA2	PDMA0 DMA02 transfer completion	62
INTDMA3	PDMA0 DMA03 transfer completion	63
INTDMA4	PDMA0 DMA04 transfer completion	64
INTDMA5	PDMA0 DMA05 transfer completion	65
INTDMA6	PDMA0 DMA06 transfer completion	66
INTDMA7	PDMA0 DMA07 transfer completion	67
INTDMA8	PDMA0 DMA10 transfer completion	68
INTDMA9	PDMA0 DMA11 transfer completion	69
INTDMA10	PDMA0 DMA12 transfer completion	70
INTDMA11	PDMA0 DMA13 transfer completion	71
INTDMA12	PDMA0 DMA14 transfer completion	72
INTDMA13	PDMA0 DMA15 transfer completion	73
INTDMA14	PDMA0 DMA16 transfer completion	74
INTDMA15	PDMA0 DMA17 transfer completion	75
INTDMA16	PDMA0 DMA20 transfer completion	298
INTDMA17	PDMA0 DMA21 transfer completion	299
INTDMA18	PDMA0 DMA22 transfer completion	300
INTDMA19	PDMA0 DMA23 transfer completion	301
INTDMA20	PDMA0 DMA24 transfer completion	302
INTDMA21	PDMA0 DMA25 transfer completion	303
INTDMA22	PDMA0 DMA26 transfer completion	304
INTDMA23	PDMA0 DMA27 transfer completion	305
INTDMA24	PDMA0 DMA30 transfer completion	306
INTDMA25	PDMA0 DMA31 transfer completion	307
INTDMA26	PDMA0 DMA32 transfer completion	308
INTDMA27	PDMA0 DMA33 transfer completion	309
INTDMA28	PDMA0 DMA34 transfer completion	310
INTDMA29	PDMA0 DMA35 transfer completion	311
INTDMA30	PDMA0 DMA36 transfer completion	312
INTDMA31	PDMA0 DMA37 transfer completion	313

Table 8.11 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number
PDMA0		
INTDMA0	PDMA0 DMA00 transfer completion	60
INTDMA1	PDMA0 DMA01 transfer completion	61
INTDMA2	PDMA0 DMA02 transfer completion	62
INTDMA3	PDMA0 DMA03 transfer completion	63
INTDMA4	PDMA0 DMA04 transfer completion	64
INTDMA5	PDMA0 DMA05 transfer completion	65
INTDMA6	PDMA0 DMA06 transfer completion	66
INTDMA7	PDMA0 DMA07 transfer completion	67
INTDMA8	PDMA0 DMA10 transfer completion	68
INTDMA9	PDMA0 DMA11 transfer completion	69
INTDMA10	PDMA0 DMA12 transfer completion	70
INTDMA11	PDMA0 DMA13 transfer completion	71
INTDMA12	PDMA0 DMA14 transfer completion	72
INTDMA13	PDMA0 DMA15 transfer completion	73
INTDMA14	PDMA0 DMA16 transfer completion	74
INTDMA15	PDMA0 DMA17 transfer completion	75

Table 8.12 Interrupt Request (FE Level Non-Maskable Interrupts) (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Name
INTDMAERR	DMA transfer error interrupt	INTDMAERR

8.1.4 DMA Trigger Factors

DMA trigger factors can be selected by setting the PDMAnDTFRm.REQSEL[6:0] bits.

The following table lists all DMA trigger factors which can be selected by the PDMAnDTFRm register.

Table 8.13 DMA Trigger Factor (RH850/F1KH-D8)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KH-D8 176 Pins	F1KH-D8 233 Pins	F1KH-D8 272 Pins	F1KH-D8 324 Pins
DMACTRG[0]	TAUD0REQSEL0*[1]	✓	✓	—	✓
DMACTRG[1]	TAUD0REQSEL1*[1]	✓	✓	—	✓
DMACTRG[2]	TAUD0REQSEL2*[1]	✓	✓	—	✓
DMACTRG[3]	TAUD0REQSEL3*[1]	✓	✓	—	✓
DMACTRG[4]	INTADCA0I0	✓	✓	—	✓
DMACTRG[5]	INTADCA0I1	✓	✓	—	✓
DMACTRG[6]	INTADCA0I2	✓	✓	—	✓
DMACTRG[7]	ADC_CONV_END0	✓	✓	—	✓
DMACTRG[8]	INTCSIG0IC	✓	✓	—	✓
DMACTRG[9]	INTCSIG0IR	✓	✓	—	✓
DMACTRG[10]	INTRLIN30UR0	✓	✓	—	✓
DMACTRG[11]	INTRLIN30UR1	✓	✓	—	✓
DMACTRG[12]	INTP0	✓	✓	—	✓
DMACTRG[13]	INTP2	✓	✓	—	✓
DMACTRG[14]	INTP4	✓	✓	—	✓
DMACTRG[15]	RSCANFDRF12	✓	✓	—	✓
DMACTRG[16]	INTRLIN37UR0	✓	✓	—	✓
DMACTRG[17]	TAUD0REQSEL4*[1]	✓	✓	—	✓
DMACTRG[18]	TAUD0REQSEL5*[1]	✓	✓	—	✓
DMACTRG[19]	INTRIIC0TI	✓	✓	—	✓
DMACTRG[20]	INTRIIC0RI	✓	✓	—	✓
DMACTRG[21]	INTTAUJ0I0	✓	✓	—	✓
DMACTRG[22]	INTTAUJ0I3	✓	✓	—	✓
DMACTRG[23]	RSCANFDCF0	✓	✓	—	✓
DMACTRG[24]	RSCANFDCF1	✓	✓	—	✓
DMACTRG[25]	RSCANFDRF13	✓	✓	—	✓
DMACTRG[26]	RSCANFDCF2	✓	✓	—	✓
DMACTRG[27]	RSCANFDCF3	✓	✓	—	✓
DMACTRG[28]	INTCSIH1IC	✓	✓	—	✓
DMACTRG[29]	INTCSIH1IR	✓	✓	—	✓
DMACTRG[30]	INTCSIH1IJC	✓	✓	—	✓
DMACTRG[31]	INTP6	✓	✓	—	✓
DMACTRG[32]	INTP8	✓	✓	—	✓
DMACTRG[33]	TAUB0REQSEL0*[1]	✓	✓	—	✓
DMACTRG[34]	TAUB0REQSEL1*[1]	✓	✓	—	✓
DMACTRG[35]	TAUB0REQSEL2*[1]	✓	✓	—	✓
DMACTRG[36]	TAUB0REQSEL3*[1]	✓	✓	—	✓
DMACTRG[37]	TAUB0REQSEL4*[1]	✓	✓	—	✓
DMACTRG[38]	TAUB0REQSEL5*[1]	✓	✓	—	✓
DMACTRG[39]	INTCSIG4IC	✓	✓	—	✓

Table 8.13 DMA Trigger Factor (RH850/F1KH-D8)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KH-D8 176 Pins	F1KH-D8 233 Pins	F1KH-D8 272 Pins	F1KH-D8 324 Pins
DMACTRG[40]	INTCSIG4IR	✓	✓	—	✓
DMACTRG[41]	INTCSIH3IC	✓	✓	—	✓
DMACTRG[42]	INTCSIH3IR	✓	✓	—	✓
DMACTRG[43]	INTCSIH3IJC	✓	✓	—	✓
DMACTRG[44]	INTRLIN32UR0	✓	✓	—	✓
DMACTRG[45]	INTRLIN32UR1	✓	✓	—	✓
DMACTRG[46]	INTTAUJ1I0	✓	✓	—	✓
DMACTRG[47]	INTTAUJ1I2	✓	✓	—	✓
DMACTRG[48]	RSCANFDCF4	✓	✓	—	✓
DMACTRG[49]	RSCANFDCF5	✓	✓	—	✓
DMACTRG[50]	INTRLIN34UR0	✓	✓	—	✓
DMACTRG[51]	INTRLIN34UR1	✓	✓	—	✓
DMACTRG[52]	TAUB1REQSEL0*1	✓	✓	—	✓
DMACTRG[53]	TAUB1REQSEL1*1	✓	✓	—	✓
DMACTRG[54]	TAUB1REQSEL2*1	✓	✓	—	✓
DMACTRG[55]	TAUB1REQSEL3*1	✓	✓	—	✓
DMACTRG[56]	TAUB1REQSEL4*1	✓	✓	—	✓
DMACTRG[57]	TAUB1REQSEL5*1 RSCANFDRF15*1	✓	✓	—	✓
DMACTRG[58]	INTCSIH4IC	✓	✓	—	✓
DMACTRG[59]	INTCSIH4IR	✓	✓	—	✓
DMACTRG[60]	RSCANFDRF0	✓	✓	—	✓
DMACTRG[61]	RSCANFDRF1	✓	✓	—	✓
DMACTRG[62]	RSCANFDRF2	✓	✓	—	✓
DMACTRG[63]	RSCANFDRF3	✓	✓	—	✓
DMACTRG[64]	RSCANFDCF6	✓	✓	—	✓
DMACTRG[65]	RSCANFDCF7	✓	✓	—	✓
DMACTRG[66]	INTCSIG1IC	✓	✓	—	✓
DMACTRG[67]	INTCSIG1IR	✓	✓	—	✓
DMACTRG[68]	RSCANFDRF4	✓	✓	—	✓
DMACTRG[69]	RSCANFDRF5	✓	✓	—	✓
DMACTRG[70]	INTCSIH0IC	✓	✓	—	✓
DMACTRG[71]	INTCSIH0IR	✓	✓	—	✓
DMACTRG[72]	INTCSIH0IJC	✓	✓	—	✓
DMACTRG[73]	INTP1	✓	✓	—	✓
DMACTRG[74]	INTP3	✓	✓	—	✓
DMACTRG[75]	INTP5	✓	✓	—	✓
DMACTRG[76]	INTCSIH4IJC	✓	✓	—	✓
DMACTRG[77]	INTRLIN37UR1	✓	✓	—	✓
DMACTRG[78]	INTCSIG2IC	✓	✓	—	✓
DMACTRG[79]	INTCSIG2IR	✓	✓	—	✓
DMACTRG[80]	INTTAUJ0I1	✓	✓	—	✓
DMACTRG[81]	INTTAUJ0I2	✓	✓	—	✓
DMACTRG[82]	RSCANFDRF6	✓	✓	—	✓
DMACTRG[83]	RSCANFDRF7	✓	✓	—	✓

Table 8.13 DMA Trigger Factor (RH850/F1KH-D8)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KH-D8 176 Pins	F1KH-D8 233 Pins	F1KH-D8 272 Pins	F1KH-D8 324 Pins
DMACTRG[84]	RSCANFDRF14	✓	✓	—	✓
DMACTRG[85]	INTDMAFL	✓	✓	—	✓
DMACTRG[86]	INTRLIN31UR0	✓	✓	—	✓
DMACTRG[87]	INTRLIN31UR1	✓	✓	—	✓
DMACTRG[88]	INTP7	✓	✓	—	✓
DMACTRG[89]	INTCSIH2IC	✓	✓	—	✓
DMACTRG[90]	INTCSIH2IR	✓	✓	—	✓
DMACTRG[91]	INTCSIH2IJC	✓	✓	—	✓
DMACTRG[92]	RSCANFDCF8	—	—	—	✓
DMACTRG[93]	RSCANFDCF9	—	—	—	✓
DMACTRG[94]	RSCANFDCF10	—	—	—	✓
DMACTRG[95]	RSCANFDCF11	—	—	—	✓
DMACTRG[96]	RSCANFDRF8	✓	✓	—	✓
DMACTRG[97]	RSCANFDRF9	✓	✓	—	✓
DMACTRG[98]	RSCANFDRF10	✓	✓	—	✓
DMACTRG[99]	RSCANFDRF11	✓	✓	—	✓
DMACTRG[100]	INTTAUJ1I1	✓	✓	—	✓
DMACTRG[101]	INTTAUJ1I3	✓	✓	—	✓
DMACTRG[102]	INTP9	✓	✓	—	✓
DMACTRG[103]	INTTAUJ2I0	✓	✓	—	✓
DMACTRG[104]	INTTAUJ2I1	✓	✓	—	✓
DMACTRG[105]	INTTAUJ2I2	✓	✓	—	✓
DMACTRG[106]	INTTAUJ2I3	✓	✓	—	✓
DMACTRG[107]	INTTAUJ3I0	✓	✓	—	✓
DMACTRG[108]	INTTAUJ3I1	✓	✓	—	✓
DMACTRG[109]	INTTAUJ3I2	✓	✓	—	✓
DMACTRG[110]	INTTAUJ3I3	✓	✓	—	✓
DMACTRG[111]	INTRLIN33UR0	✓	✓	—	✓
DMACTRG[112]	INTRLIN33UR1	✓	✓	—	✓
DMACTRG[113]	INTRIIC1TI	✓	✓	—	✓
DMACTRG[114]	INTRIIC1RI	✓	✓	—	✓
DMACTRG[115]	INTADCA1I0	✓	✓	—	✓
DMACTRG[116]	INTADCA1I1	✓	✓	—	✓
DMACTRG[117]	INTADCA1I2	✓	✓	—	✓
DMACTRG[118]	ADC_CONV_END1	✓	✓	—	✓
DMACTRG[119]	INTRLIN36UR0	✓	✓	—	✓
DMACTRG[120]	INTRLIN36UR1	✓	✓	—	✓
DMACTRG[121]	INTRLIN35UR0	✓	✓	—	✓
DMACTRG[122]	INTRLIN35UR1	✓	✓	—	✓
DMACTRG[123]	INTSENT0RI	✓	✓	—	✓
DMACTRG[124]	INTSENT1RI	✓	✓	—	✓
DMACTRG[125]	INTCSIG3IC	✓	✓	—	✓
DMACTRG[126]	INTCSIG3IR	✓	✓	—	✓
DMACTRG[127]	DMAMMCA0	—	—	—	✓

Note 1. For details, see **Section 8.10.2, DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1 — DMA Trigger Factor Select Register**.

Note 2. For details INTDMAFL of the function, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Table 8.14 DMA Trigger Factor (RH850/F1KM-S4)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S4 100 Pins	F1KM-S4 144 Pins	F1KM-S4 176 Pins	F1KM-S4 233 Pins	F1KM-S4 272 Pins
DMACTRG[0]	TAUD0REQSEL0*1	✓	✓	✓	✓	✓
DMACTRG[1]	TAUD0REQSEL1*1	✓	✓	✓	✓	✓
DMACTRG[2]	TAUD0REQSEL2*1	✓	✓	✓	✓	✓
DMACTRG[3]	TAUD0REQSEL3*1	✓	✓	✓	✓	✓
DMACTRG[4]	INTADCA0I0	✓	✓	✓	✓	✓
DMACTRG[5]	INTADCA0I1	✓	✓	✓	✓	✓
DMACTRG[6]	INTADCA0I2	✓	✓	✓	✓	✓
DMACTRG[7]	ADC_CONV_END0	✓	✓	✓	✓	✓
DMACTRG[8]	INTCSIG0IC	✓	✓	✓	✓	✓
DMACTRG[9]	INTCSIG0IR	✓	✓	✓	✓	✓
DMACTRG[10]	INTRLIN30UR0	✓	✓	✓	✓	✓
DMACTRG[11]	INTRLIN30UR1	✓	✓	✓	✓	✓
DMACTRG[12]	INTP0	✓	✓	✓	✓	✓
DMACTRG[13]	INTP2	✓	✓	✓	✓	✓
DMACTRG[14]	INTP4	✓	✓	✓	✓	✓
DMACTRG[15]	Setting prohibited	—	—	—	—	—
DMACTRG[16]	INTRLIN37UR0	—	—	✓	✓	✓
DMACTRG[17]	TAUD0REQSEL4*1	✓	✓	✓	✓	✓
DMACTRG[18]	TAUD0REQSEL5*1	✓	✓	✓	✓	✓
DMACTRG[19]	INTRIIC0TI	✓	✓	✓	✓	✓
DMACTRG[20]	INTRIIC0RI	✓	✓	✓	✓	✓
DMACTRG[21]	INTTAUJ0I0	✓	✓	✓	✓	✓
DMACTRG[22]	INTTAUJ0I3	✓	✓	✓	✓	✓
DMACTRG[23]	RSCANFDCF0	✓	✓	✓	✓	✓
DMACTRG[24]	RSCANFDCF1	✓	✓	✓	✓	✓
DMACTRG[25]	Setting prohibited	—	—	—	—	—
DMACTRG[26]	RSCANFDCF2	✓	✓	✓	✓	✓
DMACTRG[27]	RSCANFDCF3	✓	✓	✓	✓	✓
DMACTRG[28]	INTCSIH1IC	✓	✓	✓	✓	✓
DMACTRG[29]	INTCSIH1IR	✓	✓	✓	✓	✓
DMACTRG[30]	INTCSIH1IJC	✓	✓	✓	✓	✓
DMACTRG[31]	INTP6	✓	✓	✓	✓	✓
DMACTRG[32]	INTP8	✓	✓	✓	✓	✓
DMACTRG[33]	TAUB0REQSEL0*1	✓	✓	✓	✓	✓
DMACTRG[34]	TAUB0REQSEL1*1	✓	✓	✓	✓	✓
DMACTRG[35]	TAUB0REQSEL2*1	✓	✓	✓	✓	✓
DMACTRG[36]	TAUB0REQSEL3*1	✓	✓	✓	✓	✓
DMACTRG[37]	TAUB0REQSEL4*1	✓	✓	✓	✓	✓
DMACTRG[38]	TAUB0REQSEL5*1	✓	✓	✓	✓	✓
DMACTRG[39]	Setting prohibited	—	—	—	—	—
DMACTRG[40]	Setting prohibited	—	—	—	—	—
DMACTRG[41]	INTCSIH3IC	✓	✓	✓	✓	✓
DMACTRG[42]	INTCSIH3IR	✓	✓	✓	✓	✓

Table 8.14 DMA Trigger Factor (RH850/F1KM-S4)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S4 100 Pins	F1KM-S4 144 Pins	F1KM-S4 176 Pins	F1KM-S4 233 Pins	F1KM-S4 272 Pins
DMACTRG[43]	INTCSIH3IJC	✓	✓	✓	✓	✓
DMACTRG[44]	INTRLIN32UR0	✓	✓	✓	✓	✓
DMACTRG[45]	INTRLIN32UR1	✓	✓	✓	✓	✓
DMACTRG[46]	INTTAUJ1I0	✓	✓	✓	✓	✓
DMACTRG[47]	INTTAUJ1I2	✓	✓	✓	✓	✓
DMACTRG[48]	RSCANFDCF4	✓	✓	✓	✓	✓
DMACTRG[49]	RSCANFDCF5	✓	✓	✓	✓	✓
DMACTRG[50]	INTRLIN34UR0	—	✓	✓	✓	✓
DMACTRG[51]	INTRLIN34UR1	—	✓	✓	✓	✓
DMACTRG[52]	TAUB1REQSEL0*1	—	—	✓	✓	✓
DMACTRG[53]	TAUB1REQSEL1*1	—	—	✓	✓	✓
DMACTRG[54]	TAUB1REQSEL2*1	—	—	✓	✓	✓
DMACTRG[55]	TAUB1REQSEL3*1	—	—	✓	✓	✓
DMACTRG[56]	TAUB1REQSEL4*1	—	—	✓	✓	✓
DMACTRG[57]	TAUB1REQSEL5*1	—	—	✓	✓	✓
DMACTRG[58]	Setting prohibited	—	—	—	—	—
DMACTRG[59]	Setting prohibited	—	—	—	—	—
DMACTRG[60]	RSCANFDRF0	✓	✓	✓	✓	✓
DMACTRG[61]	RSCANFDRF1	✓	✓	✓	✓	✓
DMACTRG[62]	RSCANFDRF2	✓	✓	✓	✓	✓
DMACTRG[63]	RSCANFDRF3	✓	✓	✓	✓	✓
DMACTRG[64]	RSCANFDCF6	✓	✓	✓	✓	✓
DMACTRG[65]	RSCANFDCF7	✓	✓	✓	✓	✓
DMACTRG[66]	INTCSIG1IC	—	✓	✓	✓	✓
DMACTRG[67]	INTCSIG1IR	—	✓	✓	✓	✓
DMACTRG[68]	RSCANFDRF4	✓	✓	✓	✓	✓
DMACTRG[69]	RSCANFDRF5	✓	✓	✓	✓	✓
DMACTRG[70]	INTCSIH0IC	✓	✓	✓	✓	✓
DMACTRG[71]	INTCSIH0IR	✓	✓	✓	✓	✓
DMACTRG[72]	INTCSIH0IJC	✓	✓	✓	✓	✓
DMACTRG[73]	INTP1	✓	✓	✓	✓	✓
DMACTRG[74]	INTP3	✓	✓	✓	✓	✓
DMACTRG[75]	INTP5	✓	✓	✓	✓	✓
DMACTRG[76]	Setting prohibited	—	—	—	—	—
DMACTRG[77]	INTRLIN37UR1	—	—	✓	✓	✓
DMACTRG[78]	INTCSIG2IC	—	—	✓	✓	✓
DMACTRG[79]	INTCSIG2IR	—	—	✓	✓	✓
DMACTRG[80]	INTTAUJ0I1	✓	✓	✓	✓	✓
DMACTRG[81]	INTTAUJ0I2	✓	✓	✓	✓	✓
DMACTRG[82]	RSCANFDRF6	✓	✓	✓	✓	✓
DMACTRG[83]	RSCANFDRF7	✓	✓	✓	✓	✓
DMACTRG[84]	Setting prohibited	—	—	—	—	—
DMACTRG[85]	INTDMAFL*2	✓	✓	✓	✓	✓
DMACTRG[86]	INTRLIN31UR0	✓	✓	✓	✓	✓

Table 8.14 DMA Trigger Factor (RH850/F1KM-S4)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S4 100 Pins	F1KM-S4 144 Pins	F1KM-S4 176 Pins	F1KM-S4 233 Pins	F1KM-S4 272 Pins
DMACTRG[87]	INTRLIN31UR1	✓	✓	✓	✓	✓
DMACTRG[88]	INTP7	✓	✓	✓	✓	✓
DMACTRG[89]	INTCSIH2IC	✓	✓	✓	✓	✓
DMACTRG[90]	INTCSIH2IR	✓	✓	✓	✓	✓
DMACTRG[91]	INTCSIH2IJC	✓	✓	✓	✓	✓
DMACTRG[92]	Setting prohibited	—	—	—	—	—
DMACTRG[93]	Setting prohibited	—	—	—	—	—
DMACTRG[94]	Setting prohibited	—	—	—	—	—
DMACTRG[95]	Setting prohibited	—	—	—	—	—
DMACTRG[96]	Setting prohibited	—	—	—	—	—
DMACTRG[97]	Setting prohibited	—	—	—	—	—
DMACTRG[98]	Setting prohibited	—	—	—	—	—
DMACTRG[99]	Setting prohibited	—	—	—	—	—
DMACTRG[100]	INTTAUJ1I1	✓	✓	✓	✓	✓
DMACTRG[101]	INTTAUJ1I3	✓	✓	✓	✓	✓
DMACTRG[102]	INTP9	✓	✓	✓	✓	✓
DMACTRG[103]	INTTAUJ2I0	✓	✓	✓	✓	✓
DMACTRG[104]	INTTAUJ2I1	✓	✓	✓	✓	✓
DMACTRG[105]	INTTAUJ2I2	✓	✓	✓	✓	✓
DMACTRG[106]	INTTAUJ2I3	✓	✓	✓	✓	✓
DMACTRG[107]	INTTAUJ3I0	✓	✓	✓	✓	✓
DMACTRG[108]	INTTAUJ3I1	✓	✓	✓	✓	✓
DMACTRG[109]	INTTAUJ3I2	✓	✓	✓	✓	✓
DMACTRG[110]	INTTAUJ3I3	✓	✓	✓	✓	✓
DMACTRG[111]	INTRLIN33UR0	—	✓	✓	✓	✓
DMACTRG[112]	INTRLIN33UR1	—	✓	✓	✓	✓
DMACTRG[113]	INTRIIC1TI	✓	✓	✓	✓	✓
DMACTRG[114]	INTRIIC1RI	✓	✓	✓	✓	✓
DMACTRG[115]	INTADCA1I0	—	✓	✓	✓	✓
DMACTRG[116]	INTADCA1I1	—	✓	✓	✓	✓
DMACTRG[117]	INTADCA1I2	—	✓	✓	✓	✓
DMACTRG[118]	ADC_CONV_END1	—	✓	✓	✓	✓
DMACTRG[119]	INTRLIN36UR0	—	—	✓	✓	✓
DMACTRG[120]	INTRLIN36UR1	—	—	✓	✓	✓
DMACTRG[121]	INTRLIN35UR0	—	✓	✓	✓	✓
DMACTRG[122]	INTRLIN35UR1	—	✓	✓	✓	✓
DMACTRG[123]	INTSENT0RI	—	✓	✓	✓	✓
DMACTRG[124]	INTSENT1RI	✓	✓	✓	✓	✓
DMACTRG[125]	INTCSIG3IC	—	—	✓	✓	✓
DMACTRG[126]	INTCSIG3IR	—	—	✓	✓	✓
DMACTRG[127]	Setting prohibited	—	—	—	—	—

Note 1. For details, see **Section 8.10.2, DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1 — DMA Trigger Factor Select Register**.

Note 2. For details INTDMAFL of the function, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Table 8.15 DMA Trigger Factor (RH850/F1KM-S1)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S1 48 Pins	F1KM-S1 64 Pins	F1KM-S1 80 Pins	F1KM-S1 100 Pins
DMACTRG[0]	TAUD0REQSEL0*[1]	✓	✓	✓	✓
DMACTRG[1]	TAUD0REQSEL1*[1]	✓	✓	✓	✓
DMACTRG[2]	TAUD0REQSEL2*[1]	✓	✓	✓	✓
DMACTRG[3]	TAUD0REQSEL3*[1]	✓	✓	✓	✓
DMACTRG[4]	INTADCA0I0	✓	✓	✓	✓
DMACTRG[5]	INTADCA0I1	✓	✓	✓	✓
DMACTRG[6]	INTADCA0I2	✓	✓	✓	✓
DMACTRG[7]	ADC_CONV_END0	✓	✓	✓	✓
DMACTRG[8]	INTCSIG0IC	✓	✓	✓	✓
DMACTRG[9]	INTCSIG0IR	✓	✓	✓	✓
DMACTRG[10]	INTRLIN30UR0	✓	✓	✓	✓
DMACTRG[11]	INTRLIN30UR1	✓	✓	✓	✓
DMACTRG[12]	INTP0	✓	✓	✓	✓
DMACTRG[13]	INTP2	✓	✓	✓	✓
DMACTRG[14]	INTP4	✓	✓	✓	✓
DMACTRG[15]	Setting prohibited	—	—	—	—
DMACTRG[16]	Setting prohibited	—	—	—	—
DMACTRG[17]	TAUD0REQSEL4*[1]	✓	✓	✓	✓
DMACTRG[18]	TAUD0REQSEL5*[1]	✓	✓	✓	✓
DMACTRG[19]	INTRIIC0TI	✓	✓	✓	✓
DMACTRG[20]	INTRIIC0RI	✓	✓	✓	✓
DMACTRG[21]	INTTAUJ0I0	✓	✓	✓	✓
DMACTRG[22]	INTTAUJ0I3	✓	✓	✓	✓
DMACTRG[23]	RSCANFDCF0	✓	✓	✓	✓
DMACTRG[24]	RSCANFDCF1	—	✓	✓	✓
DMACTRG[25]	Setting prohibited	—	—	—	—
DMACTRG[26]	RSCANFDCF2	—	✓	✓	✓
DMACTRG[27]	RSCANFDCF3	—	—	—	✓
DMACTRG[28]	INTCSIH1IC	—	—	✓	✓
DMACTRG[29]	INTCSIH1IR	—	—	✓	✓
DMACTRG[30]	INTCSIH1IJC	—	—	✓	✓
DMACTRG[31]	INTP6	—	—	✓	✓
DMACTRG[32]	INTP8	—	—	✓	✓
DMACTRG[33]	TAUB0REQSEL0*[1]	—	—	✓	✓
DMACTRG[34]	TAUB0REQSEL1*[1]	—	—	✓	✓
DMACTRG[35]	TAUB0REQSEL2*[1]	—	—	✓	✓
DMACTRG[36]	TAUB0REQSEL3*[1]	—	—	✓	✓
DMACTRG[37]	TAUB0REQSEL4*[1]	—	—	✓	✓
DMACTRG[38]	TAUB0REQSEL5*[1]	—	—	✓	✓
DMACTRG[39]	Setting prohibited	—	—	—	—
DMACTRG[40]	Setting prohibited	—	—	—	—
DMACTRG[41]	INTCSIH3IC	—	—	—	✓
DMACTRG[42]	INTCSIH3IR	—	—	—	✓
DMACTRG[43]	INTCSIH3IJC	—	—	—	✓
DMACTRG[44]	INTRLIN32UR0	—	—	✓	✓

Table 8.15 DMA Trigger Factor (RH850/F1KM-S1)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S1 48 Pins	F1KM-S1 64 Pins	F1KM-S1 80 Pins	F1KM-S1 100 Pins
DMACTRG[45]	INTRLIN32UR1	—	—	✓	✓
DMACTRG[46]	INTTAUJ1I0	✓	✓	✓	✓
DMACTRG[47]	INTTAUJ1I2	✓	✓	✓	✓
DMACTRG[48]	RSCANFDCF4	—	—	—	✓
DMACTRG[49]	RSCANFDCF5	—	—	—	✓
DMACTRG[50]	Setting prohibited	—	—	—	—
DMACTRG[51]	Setting prohibited	—	—	—	—
DMACTRG[52]	Setting prohibited	—	—	—	—
DMACTRG[53]	Setting prohibited	—	—	—	—
DMACTRG[54]	Setting prohibited	—	—	—	—
DMACTRG[55]	Setting prohibited	—	—	—	—
DMACTRG[56]	Setting prohibited	—	—	—	—
DMACTRG[57]	Setting prohibited	—	—	—	—
DMACTRG[58]	Setting prohibited	—	—	—	—
DMACTRG[59]	Setting prohibited	—	—	—	—
DMACTRG[60]	RSCANFDRF0	✓	✓	✓	✓
DMACTRG[61]	RSCANFDRF1	—	✓	✓	✓
DMACTRG[62]	RSCANFDRF2	—	✓	✓	✓
DMACTRG[63]	RSCANFDRF3	—	—	—	✓
DMACTRG[64]	Setting prohibited	—	—	—	—
DMACTRG[65]	Setting prohibited	—	—	—	—
DMACTRG[66]	Setting prohibited	—	—	—	—
DMACTRG[67]	Setting prohibited	—	—	—	—
DMACTRG[68]	RSCANFDRF4	—	—	—	✓
DMACTRG[69]	RSCANFDRF5	—	—	—	✓
DMACTRG[70]	INTCSIH0IC	✓	✓	✓	✓
DMACTRG[71]	INTCSIH0IR	✓	✓	✓	✓
DMACTRG[72]	INTCSIH0IJC	✓	✓	✓	✓
DMACTRG[73]	INTP1	✓	✓	✓	✓
DMACTRG[74]	INTP3	✓	✓	✓	✓
DMACTRG[75]	INTP5	✓	✓	✓	✓
DMACTRG[76]	Setting prohibited	—	—	—	—
DMACTRG[77]	Setting prohibited	—	—	—	—
DMACTRG[78]	Setting prohibited	—	—	—	—
DMACTRG[79]	Setting prohibited	—	—	—	—
DMACTRG[80]	INTTAUJ0I1	✓	✓	✓	✓
DMACTRG[81]	INTTAUJ0I2	✓	✓	✓	✓
DMACTRG[82]	Setting prohibited	—	—	—	—
DMACTRG[83]	Setting prohibited	—	—	—	—
DMACTRG[84]	Setting prohibited	—	—	—	—
DMACTRG[85]	INTDMAFL*2	✓	✓	✓	✓
DMACTRG[86]	INTRLIN31UR0	—	✓	✓	✓
DMACTRG[87]	INTRLIN31UR1	—	✓	✓	✓
DMACTRG[88]	INTP7	—	—	✓	✓
DMACTRG[89]	INTCSIH2IC	—	—	✓	✓

Table 8.15 DMA Trigger Factor (RH850/F1KM-S1)

DMA Trigger Number PDMAnDTFRm.REQSEL[6:0]	DMA Trigger Factor	F1KM-S1 48 Pins	F1KM-S1 64 Pins	F1KM-S1 80 Pins	F1KM-S1 100 Pins
DMACTRG[90]	INTCSIH2IR	—	—	✓	✓
DMACTRG[91]	INTCSIH2IJC	—	—	✓	✓
DMACTRG[92]	Setting prohibited	—	—	—	—
DMACTRG[93]	Setting prohibited	—	—	—	—
DMACTRG[94]	Setting prohibited	—	—	—	—
DMACTRG[95]	Setting prohibited	—	—	—	—
DMACTRG[96]	Setting prohibited	—	—	—	—
DMACTRG[97]	Setting prohibited	—	—	—	—
DMACTRG[98]	Setting prohibited	—	—	—	—
DMACTRG[99]	Setting prohibited	—	—	—	—
DMACTRG[100]	INTTAUJ1I1	✓	✓	✓	✓
DMACTRG[101]	INTTAUJ1I3	✓	✓	✓	✓
DMACTRG[102]	Setting prohibited	—	—	—	—
DMACTRG[103]	INTTAUJ2I0	✓	✓	✓	✓
DMACTRG[104]	INTTAUJ2I1	✓	✓	✓	✓
DMACTRG[105]	INTTAUJ2I2	✓	✓	✓	✓
DMACTRG[106]	INTTAUJ2I3	✓	✓	✓	✓
DMACTRG[107]	INTTAUJ3I0	✓	✓	✓	✓
DMACTRG[108]	INTTAUJ3I1	✓	✓	✓	✓
DMACTRG[109]	INTTAUJ3I2	✓	✓	✓	✓
DMACTRG[110]	INTTAUJ3I3	✓	✓	✓	✓
DMACTRG[111]	INTRLIN33UR0	—	—	—	✓
DMACTRG[112]	INTRLIN33UR1	—	—	—	✓
DMACTRG[113]	INTRIIC1TI	✓	✓	✓	✓
DMACTRG[114]	INTRIIC1RI	✓	✓	✓	✓
DMACTRG[115]	Setting prohibited	—	—	—	—
DMACTRG[116]	Setting prohibited	—	—	—	—
DMACTRG[117]	Setting prohibited	—	—	—	—
DMACTRG[118]	Setting prohibited	—	—	—	—
DMACTRG[119]	Setting prohibited	—	—	—	—
DMACTRG[120]	Setting prohibited	—	—	—	—
DMACTRG[121]	Setting prohibited	—	—	—	—
DMACTRG[122]	Setting prohibited	—	—	—	—
DMACTRG[123]	INTSENT0RI	✓	✓	✓	✓
DMACTRG[124]	INTSENT1RI	✓	✓	✓	✓
DMACTRG[125]	Setting prohibited	—	—	—	—
DMACTRG[126]	Setting prohibited	—	—	—	—
DMACTRG[127]	Setting prohibited	—	—	—	—

Note 1. For details, see **Section 8.10.2, DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1 — DMA Trigger Factor Select Register**.

Note 2. For details INTDMAFL of the function, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

8.2 Overview

8.2.1 Overview

Direct memory access (DMA) is used to access data without intervention of the CPU.

DMA includes a DMA transfer module called DMAC. A DMAC includes registers for storing transfer information.

- RH850/F1KH-D8

  DMA has two 8-channel × 4-group DMAC modules. 2 DMA modules operate respectively as different master. This chapter describes the function of one DMAC unit.

- RH850/F1KM-S4

  DMA has one 8-channel × 4-group DMAC modules.

- RH850/F1KM-S1

  DMA has one 8-channel × 2-group DMAC modules.

In this manual, DTFR denotes the function to select among hardware DMA transfer sources for a DMAC and retain the DMA transfer request. The DTFR can handle 128 types of hardware DMA transfer sources.

The address space that can be used for DMA transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DMA, see **Section 4A, Address Space of RH850/F1KH-D8**, **Section 4B, Address Space of RH850/F1KM-S4** and **Section 4C, Address Space of RH850/F1KM-S1**.

CAUTION

DMA can be used after PEG access permission setting.

- RH850/F1KH-D8:
  When accessing from the DMA to the local RAM in the PE, the PEGSP register, PEGGnMK register (n = 0 to 3), PEGGnBA register (n = 0 to 3) and PDMAnDMyiCM register (yi = 00 to 07, 10 to 17, 20 to 27, and 30 to 37) must be set.
- RH850/F1KM-S4:
  When accessing from the DMA to the local RAM in the PE, the PEGSP register, PEGGnMK register (n = 0 to 3), PEGGnBA register (n = 0 to 3) and PDMAnDMyiCM register (yi = 00 to 07, 10 to 17, 20 to 27, and 30 to 37) must be set.
- RH850/F1KM-S1:
  When accessing from the DMA to the local RAM and the retention RAM in the PE, the PEGSP register, PEGGnMK register (n = 0 to 3), PEGGnBA register (n = 0 to 3) and PDMAnDMyiCM register (yi = 00 to 07, 10 to 17) must be set.

8.2.2 Term Definition

Table 8.16, List of Term Definitions shows the terms used in this section.

Table 8.16 List of Term Definitions

Term	Description
DMA transfer	A general term for data transfer carried out by DMA.
DMA cycle	A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DMA cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle.
Hardware DMA transfer source	A trigger for a DMA transfer request sent by an internal peripheral device.
Hardware DMA transfer request	A DMA transfer request generated by a hardware DMA transfer source.
Software DMA transfer request	A DMA transfer request generated by software writing to a register.
DMA transfer request	A trigger to start DMA transfer with the DMAC.
Transfer information	The information required for DMA transfer, including the source address, destination address, transfer data size, and transfer count.
Single transfer	DMA transfer consisting of one DMA cycle started by one DMA transfer request.
Block transfer 1	DMA transfer consisting of the number of DMA cycles specified by the transfer count in the transfer information, started by one DMA transfer request.
Block transfer 2	DMA transfer consisting of the number of DMA cycles specified by the address reload count in the transfer information, started by one DMA transfer request.
Block transfer	A general term for both block transfer 1 and block transfer 2.
Last transfer	The DMA cycle carried out when the transfer count in the transfer information is 1.
Address reload transfer	The DMA cycle carried out when the address reload count in the transfer information is 1 if the reload function 2 is used.
Suspension	An action of pausing DMA transfer during block transfer. You can resume DMA transfer after suspension.
Resume	An action of resuming suspended DMA transfer.
Transfer abort	An action of aborting DMA transfer in the middle. You cannot resume DMA transfer after that.

8.3 DMA Function

8.3.1 Basic Operation of DMA Transfer

8.3.1.1 Transfer Mode

DMA has three transfer modes.

Single Transfer

One DMA cycle is executed when a DMA transfer request is acknowledged.

Block Transfer 1

The number of DMA cycles specified in the transfer count register are executed when a DMA transfer request is acknowledged.

Block Transfer 2

The number of DMA cycles specified by the address reload count are executed when a DMA transfer request is acknowledged. If the value specified in the address reload count is larger than the value in the transfer count register, the number of DMA cycles specified in the transfer count register are executed.

8.3.1.2 Executing a DMA Cycle

DMA always executes a write cycle after a read cycle is complete.

For example, if the transfer data size is 128 bits, a write cycle is executed after a read cycle for the 128-bit data is finished. A write cycle never starts in the middle of a read cycle.

8.3.1.3 Updating Transfer Information

When a DMA cycle is executed, DMA updates transfer information as follows:

Source Address and Destination Address

Transfer information will be updated as described in **Table 8.17, Updating the Source Addresses and the Destination Addresses** according to the settings in the transfer control register such as the count directions of source address and destination address and transfer data size.

Table 8.17 Updating the Source Addresses and the Destination Addresses

Count Direction	Transfer Data Size	Address after Update
Increment	8 bits	(address before update) + 0000 0001_H
	16 bits	(address before update) + 0000 0002_H
	32 bits	(address before update) + 0000 0004_H
	64 bits	(address before update) + 0000 0008_H
	128 bits	(address before update) + 0000 0010_H
Decrement	8 bits	(address before update) – 0000 0001_H
	16 bits	(address before update) – 0000 0002_H
	32 bits	(address before update) – 0000 0004_H
	64 bits	(address before update) – 0000 0008_H
	128 bits	(address before update) – 0000 0010_H
Fixed	—	Same as the address before update.

When you use the reload function, a special update rule is applied other than the one described in **Table 8.17, Updating the Source Addresses and the Destination Addresses** for the last transfer and the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

Transfer Count/Address Reload Count

The transfer count is decremented by one for every DMA cycle.

The address reload count is decremented by one for every DMA cycle when the reload function 2 or block transfer 2 is used. When the reload function 2 or block transfer 2 is not used, it is not updated.

When you use the reload function, a special update rule is applied for the last transfer and the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

Other transfer information

Other transfer information is not updated during execution of a DMA cycle.

8.3.1.4 Last Transfer and Address Reload Transfer

The last transfer means a DMA cycle executed when the value in the transfer count register, which shows the remaining number of transfers, is one. The last transfer differs in operation compared to other DMA cycles as follows.

- The transfer completion flag (PDMAnDCSTm.TC) is set when the last transfer is complete.
  The channel operation enable (PDMAnDCENm.DTE) bit is cleared when the last transfer is complete (when continuous transfer is disabled).
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When the reload function 1 is enabled, the reload function 1 is executed at the timing of the last transfer. For details, see **Section 8.3.3, Reload Function**.

The address reload transfer means a DMA cycle executed when the reload function 2 is enabled and the address reload count is one. The reload function 2 is executed during the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

8.3.1.5 Transfer Completion Interrupt Output

DMA can output a transfer completion interrupt to external devices.

Transfer Completion Interrupt Output

When the transfer completion interrupt output enable (PDMAnDTCTm.TCE) is set in the transfer control register, a DMAC requests a transfer completion interrupt when the last transfer is complete.

Figure 8.1, Transfer Completion Interrupt shows the operation of the transfer completion interrupt.

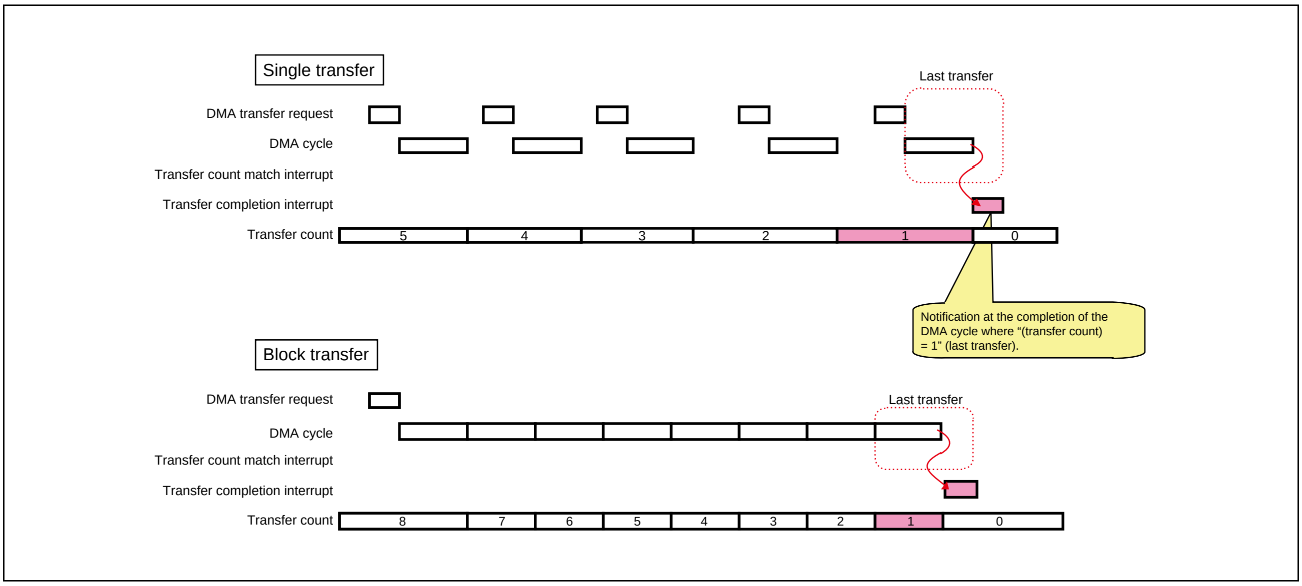


Figure 8.1 Transfer Completion Interrupt

8.3.1.6 Continuous Transfer

If the continuous transfer is not used, a DMAC sets the transfer completion flag (PDMAnDCSTm.TC) and clears the channel operation enable (PDMAnDCENm.DTE) bit when the last transfer is complete. In this case, a DMA transfer request is not accepted when the request is generated after the completion of the last transfer.

If the continuous transfer is used, the channel operation enable (PDMAnDCENm.DTE) bit is not cleared when the last transfer is complete, and a DMA transfer request can be accepted even when the transfer completion flag is set. If DMA is used for a case where a specified number of DMA transfers are executed repetitively, software overhead associated with clearing the transfer completion flag and setting the channel operation enable bit after the completion of the last transfer can be reduced by using the continuous transfer.

The continuous transfer is enabled by setting the continuous transfer enable (PDMAnDTCTm.MLE) in the DMAC transfer control register.

The continuous transfer is designed to work with the reload function 1. The continuous transfer function itself does not update the source address register, destination address register, and transfer count register. If, after the last transfer is complete, you want to restore the source address register, destination address register, and transfer count register to the state before the DMA transfer starts, use the reload function 1 and set the values of the source address register, destination address register, and the transfer count register before the DMA transfer starts to the reload source address register, reload destination address register, and reload transfer count register respectively.

Figure 8.2, Operation of Continuous Transfer by a DMAC shows an operation of continuous transfer by a DMAC.

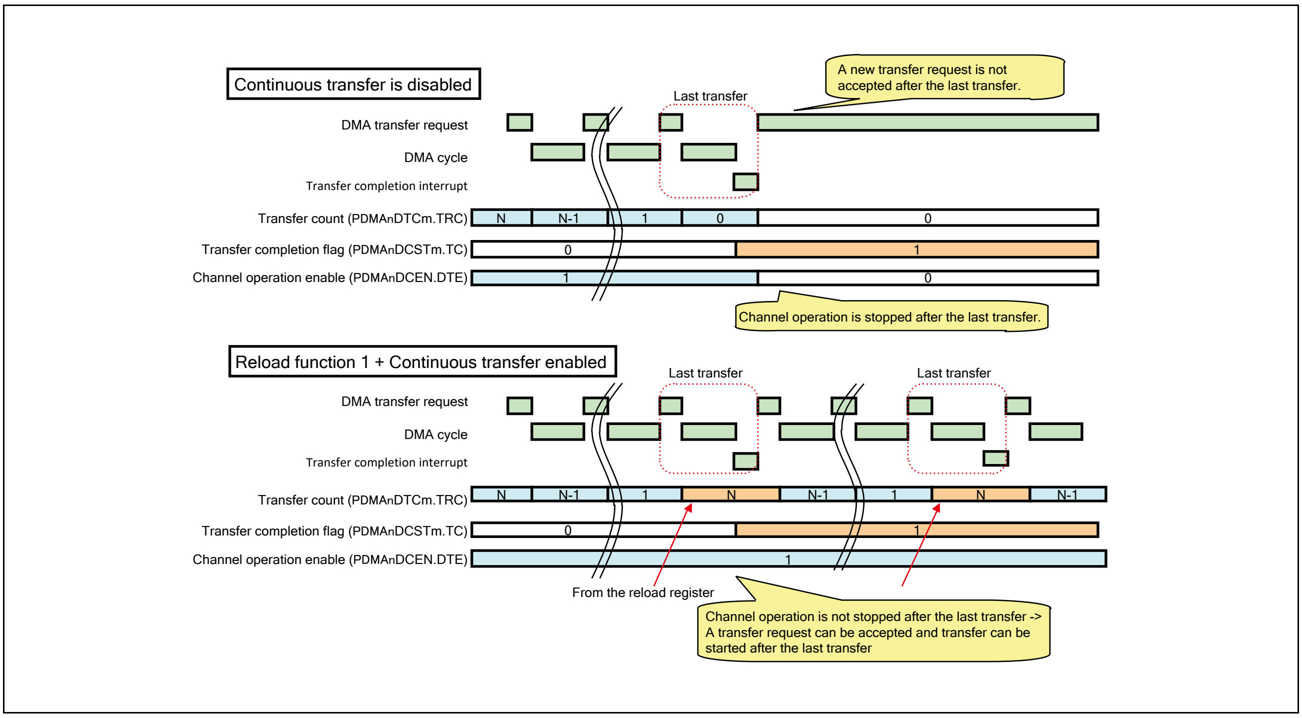


Figure 8.2 Operation of Continuous Transfer by a DMAC

8.3.2 Channel Priority Order

This subsection explains arbitration between multiple DMA channels.

8.3.2.1 DMAC Channel Arbitration

A DMAC select one channel out of eight channels with arbitration. Arbitration is done according to the fixed priority order. The priority order is “channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7” for DMAC0, and “channel 8 > channel 9 > channel 10 > channel 11 > channel 12 > channel 13 > channel 14 > channel 15” for DMAC1, “channel 16 > channel 17 > channel 18 > channel 19 > channel 20 > channel 21 > channel 22 > channel 23” for DMAC2, and “channel 24> channel 25 > channel 26 > channel 27 > channel 28 > channel 29 > channel 30 > channel 31” for DMAC3.

Arbitration is done for every DMA cycle. No arbitration occurs between the read and write of a DMA cycle.

If, at the timing when one DMA cycle completes in the middle of a block transfer of a channel, there is a DMA transfer request from a channel with a higher priority, a DMA cycle of the channel with the higher priority will be executed as the result of arbitration.

If a DMAC executes the block transfer 1 or block transfer 2, DMAC channel arbitration is done for every DMA cycle, and possibly a DMA cycle of another DMAC channel with a higher priority may interrupt.

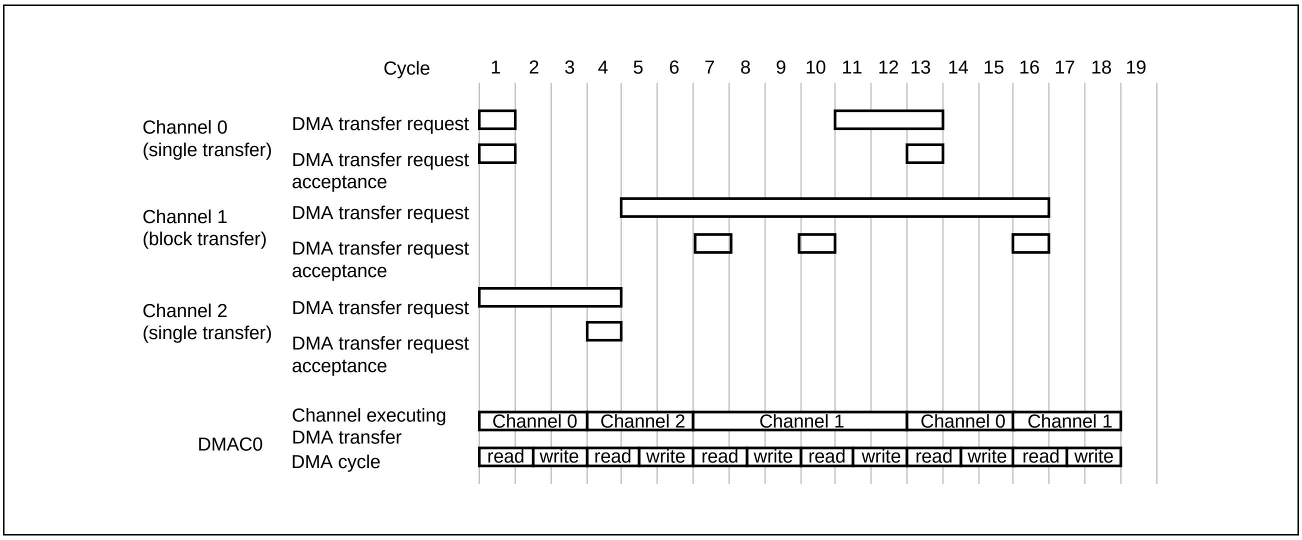


Figure 8.3 DMAC Channel Arbitration

Cycle numbers shown in **Figure 8.3, DMAC Channel Arbitration** are for explanation purpose only. They do not indicate an actual number of cycles necessary for executing DMA transfer.

In **Figure 8.3, DMAC Channel Arbitration**, DMA transfer requests for channels 0 and 2 are generated at Cycle 1. As a result of arbitration, a DMA cycle for channel 0 starts because its priority is higher. At Cycle 4, a DMA cycle for channel 2 starts. At Cycle 5, a DMA transfer request for channel 1 is generated. However, since the DMA cycle for channel 2 is still ongoing, no arbitration is done at this point. At Cycle 7, a DMA cycle for channel 1 starts. Because channel 1 uses block transfer, this DMA cycle continues at Cycle 10 where there are no other DMA transfer requests from other channel. At Cycle 11, a DMA transfer request for channel 0 is generated. However, since the DMA cycle for channel 1 is still ongoing, no arbitration is done at this point. At Cycle 12, the DMA cycle for channel 1 is complete. At Cycle 13, a DMA cycle for channel 0 starts as a result of arbitration between DMA channels 0 and 1.

It should be noted that, even though a block transfer of channel 1 has been already started, a DMA cycle of not channel 1 but channel 0 is executed at Cycle 13 because the priority of the latter is higher. At Cycle 15, the DMA cycle for channel 0 is complete. At Cycle 16, a DMA cycle for channel 1 starts again. At Cycle 18, the last DMA cycle of the block transfer of channel 1 is complete.

8.3.2.2 Interface Arbitration

DMAC0, DMAC1, DMAC2, and DMAC3 work independently and execute DMA transfer.

8.3.3 Reload Function

8.3.3.1 Overview of the Reload Function

The reload function updates a portion of transfer information, more specifically, the source address, destination address, transfer count, and address reload count, to the predefined values during DMA transfer.

The reload function has two types of functions: reload function 1 and reload function 2.

8.3.3.2 Operation of Reload Function 1

When the reload function 1 is enabled, actions described in **Table 8.18, Operation of Reload Function 1** are executed at the timing of the last transfer according to the reload function 1 setting.

Table 8.18 Operation of Reload Function 1

Reload Function 1 Setting	Register	Action at the Last Transfer
00 (Reload function 1 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Transfer count	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 1 enabled. Reloading source address and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If the reload function 2 is disable: Not reloaded. If the reload function 2 is enabled: The reload address reload count is copied to this.
10 (Reload function 1 enabled. Reloading destination address and transfer count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If the reload function 2 is disable: Not reloaded. If the reload function 2 is enabled: The reload address reload count is copied to this.
11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If the reload function 2 is disable: Not reloaded. If the reload function 2 is enabled: The reload address reload count is copied to this.

Figure 8.4, Operation of Reload Function 1 shows an operation of the reload function 1.

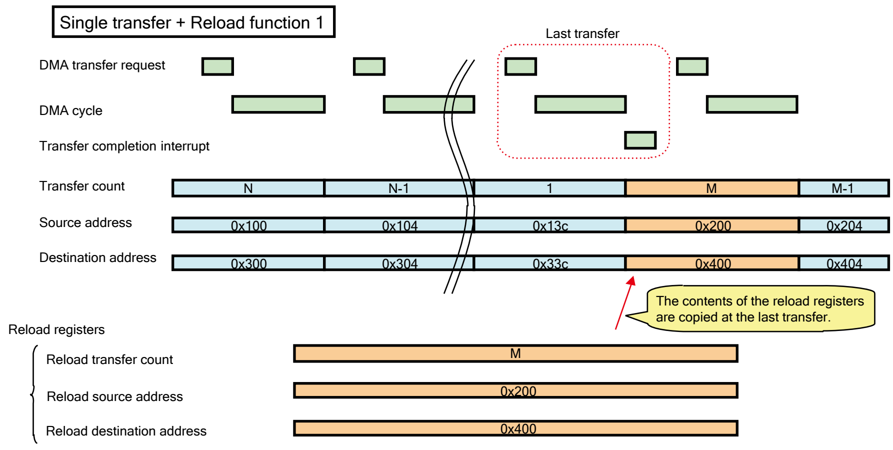


Figure 8.4 Operation of Reload Function 1

8.3.3.3 Reload Function 2

When the reload function 2 is enabled, actions described in **Table 8.19, Operation of Reload Function 2** are executed at the timing of the address reload transfer according to the reload function 2 setting.

Table 8.19 Operation of Reload Function 2

Reload Function 2 Setting	Register	Action at the Address Reload Transfer
00 (Reload function 2 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 2 enabled. Reloading source address and address reload count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Address reload count	The reload address reload count is copied to this.
10 (Reload function 2 enabled. Reloading destination address and address reload count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.
11 (Reload function 2 enabled. Reloading source address and destination address and address reload count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.

Figure 8.5, Operation of Reload Function 2 shows an operation of the reload function 2.

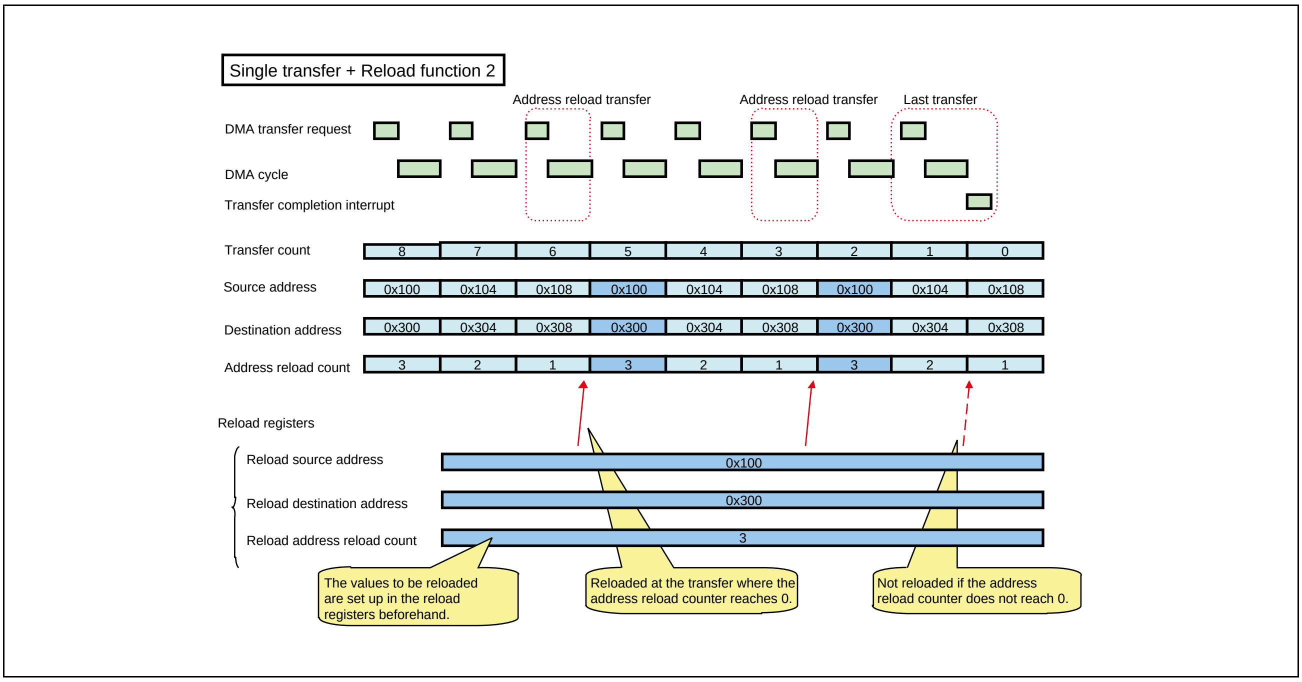


Figure 8.5 Operation of Reload Function 2

Figure 8.6, Operation when Combining the Reload Function 1 and the Reload Function 2 shows an operation when both the reload function 1 and the reload function 2 are used simultaneously.

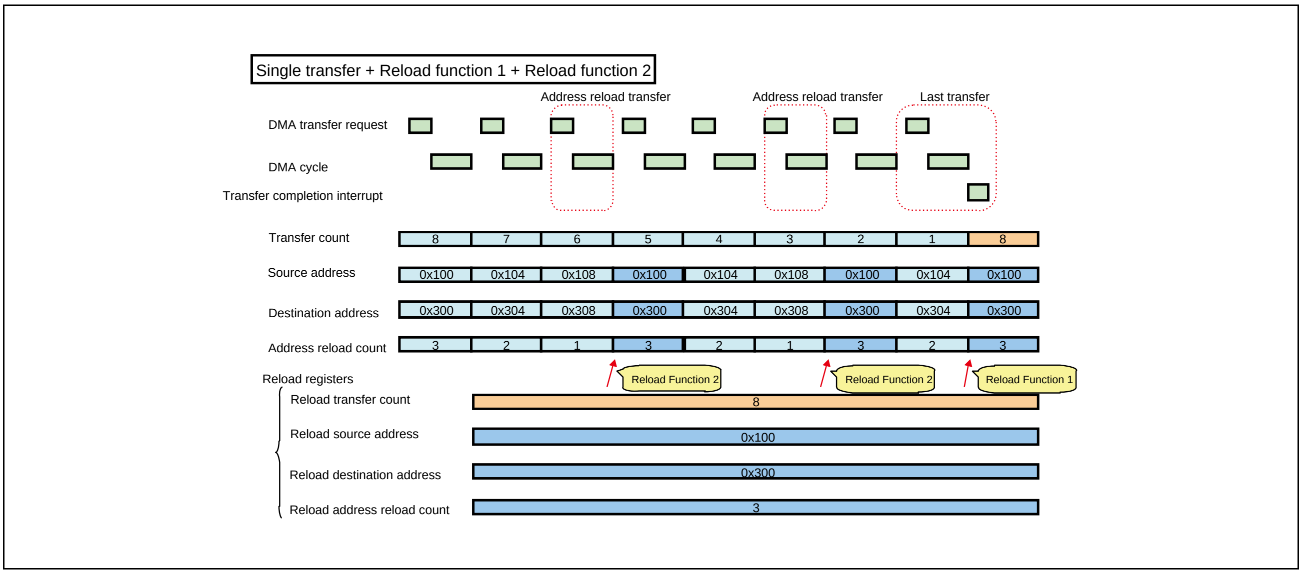


Figure 8.6 Operation when Combining the Reload Function 1 and the Reload Function 2

8.3.3.4 Timing of Setting DMAC Reload Registers

You can set up the reload source address register, reload destination address register, and reload transfer count register any time (even during DMA transfer). However, if you update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, there may be a conflict between reloading at the last transfer or address reload transfer and update of the reload register by users. In order to avoid this conflict, setting up reload registers must be completed before the last transfer or address reload transfer starts.

8.3.4 Chain Function

8.3.4.1 Overview

DMA offers a function called a chain function. With this function, the completion of the DMA cycle or last transfer for one channel can trigger a DMA transfer request for another channel. A DMA transfer request for another channel initiated by the chain function is called a chain request.

You can select the condition for generating a chain request from the following two options:

- Always chain: A chain request is generated at the completion of every DMA cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 8.7, Operation of “Always Chain” shows the operation of “always chain.”

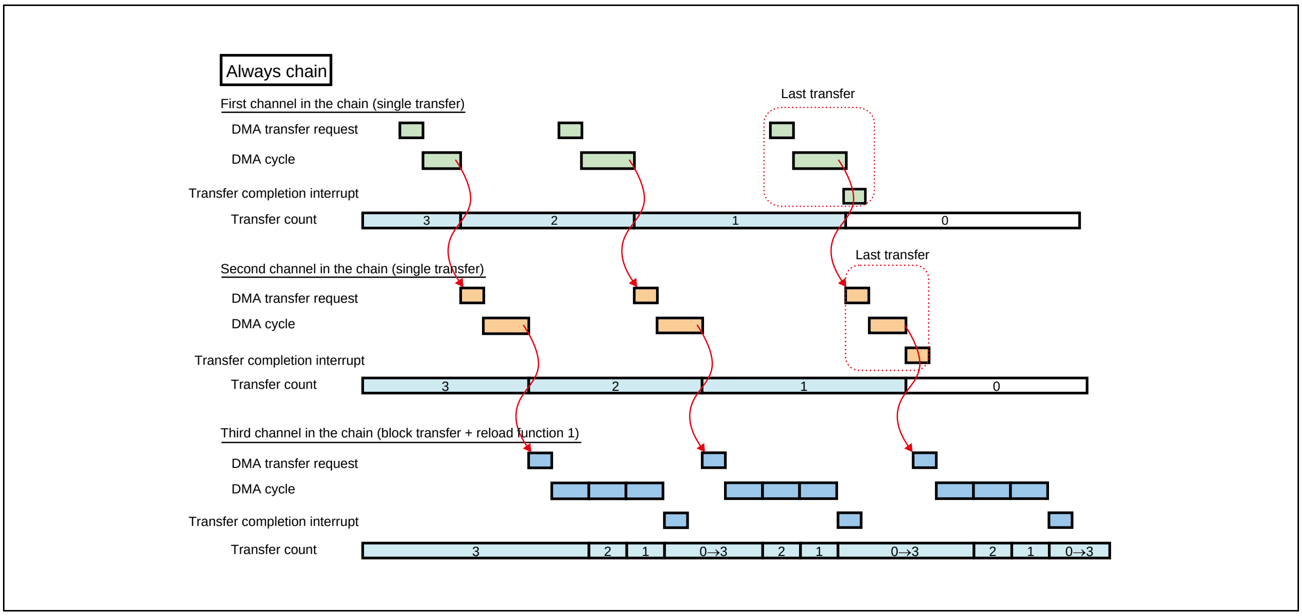


Figure 8.7 Operation of “Always Chain”

Figure 8.8, Operation of "Chain at the Last Transfer" shows the operation of "chain at the last transfer."

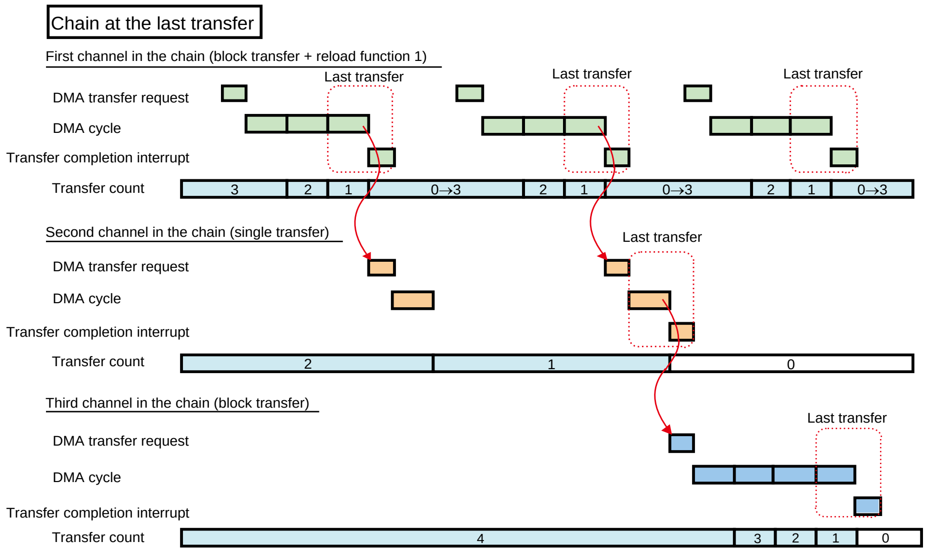


Figure 8.8 Operation of "Chain at the Last Transfer"

8.3.4.2 Setting Up the Chain Function

For a DMAC, you need to write to the chain enable (PDMAnDTCTm.CHNE) and the next channel in the chain selection (PDMAnDTCTm.CHNSEL) in the DMAC transfer control register in order to set up the type of chain function and the next channel number in the chain.

8.3.4.3 Caution for Using the Chain Function

The chain function is enabled by setting the software DMA transfer request flag of the next channel in the chain as a part of its function. Therefore, you need to set up the channel settings of the next channel in the chain in the same way as when the software DMA transfer request is used. If you specify a channel using the hardware DMA transfer request for the next channel in the chain, the chain function does not work.

A channel and its next channel in the chain must belong to the same group (DMAC0, DMAC1, DMAC2, and DMAC3). You cannot specify a channel in another group for its next channel in the chain.

8.3.5 DMAC Operation

8.3.5.1 Types of DMA Transfer Requests and Assigning DMA Transfer Requests

A DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request selection assignment (PDMAnDTCTm.DRS) bit in the DMAC transfer control register (PDMAnDTCTm) determines whether a hardware DMA transfer request or a software DMA transfer request is used.

In the case of a hardware DMA transfer request for a DMAC, DTFR selects one out of 128 hardware DMA transfer factor and assigned to each channel of the DMAC. This assignment is configured in the DTFR setting registers.

8.3.5.2 Generating and Accepting a Software DMA Transfer Request

By setting the software DMA transfer request flag (PDMAnDCSTm.SR) in the DMAC transfer status register (PDMAnDCSTm) using the DMAC transfer status set register (PDMAnDCSTSm), a software DMA transfer request can be generated.

The software DMA transfer request flag is automatically cleared when the DMAC processes the DMA transfer request. The timing when the software DMA transfer request flag is automatically cleared differs depending on the transfer mode of the DMA transfer to be executed.

- In the single transfer mode, the software DMA transfer request flag is cleared whenever the software DMA transfer request is accepted.
- In the block transfer 1 mode, the software DMA transfer request flag is cleared when the last transfer starts.
- In the block transfer 2 mode, the software DMA transfer request flag is cleared when the last transfer or address reload transfer starts.

The software DMA transfer request flag can also be cleared by software using the DMAC transfer status clear register (PDMAnDCSTCm). When you abort a DMA transfer of a DMAC channel, you must clear the software DMA transfer request flag.

8.3.5.3 Executing DMA Transfer

When the DMAC accepts a DMA transfer request for a channel, the DMAC executes DMA transfer of the channel. If there are DMA transfer requests from multiple channels, DMAC does DMAC channel arbitration and decides a channel to be acknowledged.

8.4 Suspension, Resume, Transfer Abort, and Clearing of a DMA Transfer Request

8.4.1 DMA Suspension and Resume by Software Control

The DMA control register (PDMAnDMACTL) is used to suspend DMA transfer for all channels.

When the DMA suspension bit (PDMAnDMACTL.DMASPD) in the DMA control register is set, DMA puts all channels into the suspended state. If all channels are in the suspended state and the DMA suspension bit in the DMA control register is cleared, DMA restores all channels from the suspended state to the normal state and resumes the DMA transfer of the suspended channel.

When all channels are in the suspended state, DMA transfer is suspended for all channels without changing the value of the PDMAnDCENm.DTE bit of each DMAC channel.

8.4.2 Suspension, Resume, and Transfer Abort of a DMAC Channel

You can suspend the DMA transfer of a DMAC channel by clearing the channel operation enable bit (PDMAnDCENm.DTE) in the DMAC channel operation enable setting register for the channel. If a DMA cycle is ongoing, the DMA transfer of the channel is suspended after the currently ongoing DMA cycle is finished. If you set the PDMAnDCENm.DTE bit again while the DMA transfer of the channel is suspended, the DMA transfer of the channel is resumed.

If you want to abort the currently ongoing DMA transfer of a DMAC channel, similarly clear the channel operation enable bit (PDMAnDCENm.DTE) in the DMAC channel operation enable setting register, and then clear the hardware DMA transfer request in the DTFR in the case of a hardware DMA transfer request, or clear the software DMA transfer request flag (PDMAnDCSTm.SR) using the DMAC transfer request flag clear bit (PDMAnDCSTCm.SRC) in the DMAC transfer status clear register in the case of a software DMA transfer request.

In case that the continuous transfer enable bit (PDMAnDTCTm.MLE) is set, the channel operation enable bit (PDMAnDCENm.DTE) is kept to be set. Even though the channel operation enable bit (PDMAnDCENm.DTE) is cleared by software during a DMA cycle in a last transfer, the function of the continuous transfer enable bit (PDMAnDTCTm.MLE) is given high priority and the channel operation enable bit (PDMAnDCENm.DTE) is set after completion of the last transfer.

If you want to abort an ongoing DMA transfer of a DMAC channel when continuous transfer function is enabled, please clear the continuous transfer enable bit (PDMAnDTCTm.MLE) first and then clear the channel operation enable bit (PDMAnDCENm.DTE) to abort DMA transfer of the DMAC channel. Only for the operation, DMAC Transfer Control Register (PDMAnDTCTm) can be written under the channel operation is enabled (PDMAnDCENm.DTE = 1).

Figure 8.9, Example of Suspension, Resume, and Transfer Abort of a DMAC Channel shows an example of suspension, resume, and transfer abort of a DMAC channel.

In **Figure 8.9, Example of Suspension, Resume, and Transfer Abort of a DMAC Channel**, both channels 0 and 1 execute block transfer. At time tick 1, DMA transfer of channel 1 starts. At time tick 2, a DMA transfer request for channel 0 is accepted. As a result of DMAC channel arbitration, DMA transfer of channel 0 starts because channel 0 has a higher priority than channel 1. At time tick 3, the last transfer of channel 0 is complete, and the remaining DMA transfer in the block transfer of channel 1 starts. At time tick 4, the last transfer of channel 1 is complete. After time tick 5, DMA transfer of channel 0 and DMA transfer of channel 1 are executed similarly. At time tick 7, the DMA transfer of channel 0 is suspended and, as a result of DMAC channel arbitration, the DMA transfer of channel 1 starts. At time tick 8, the last transfer of channel 1 is complete, and then, at time tick 9, the DMA transfer of channel 0 resumes. At time tick 10, the DMA transfer of channel 0 is suspended again, and then, at time tick 11, the DMA transfer of channel 0 is aborted. At time tick 12, the suspended state for channel 0 is cleared, but the DMA transfer of channel 0 is not executed because the DMA transfer is aborted at time tick 11.

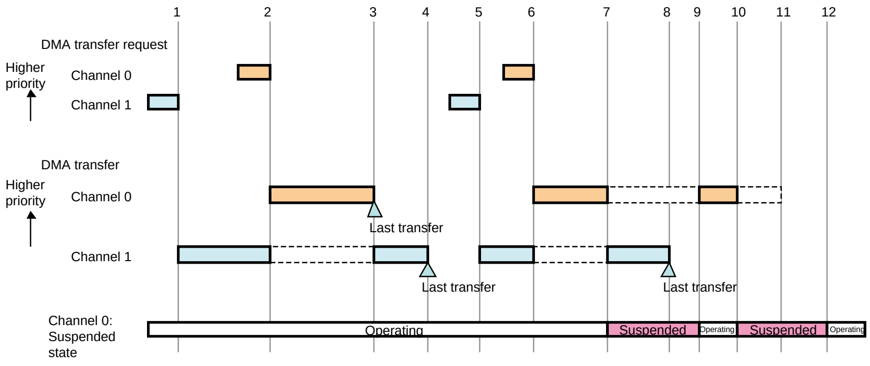


Figure 8.9 Example of Suspension, Resume, and Transfer Abort of a DMAC Channel

8.4.3 Masking and Clearing a Hardware DMA Transfer Request by the DTFR

If a DMAC uses a hardware DMA transfer request, you can temporarily disable (mask) the hardware DMA transfer request output from the DTFR to the DMAC by clearing the hardware DMA transfer source selection enable bit (PDMAnDTFRm.REQEN) in the DTFR setting register.

Also, if a hardware DMA transfer source is used, you can clear a hardware DMA transfer request retained in the DTFR by using the hardware DMA transfer request clear (PDMAnDTFRRQCm.DRQC) bit in the DTFR transfer request clear register.

Even if you suspend or abort DMA transfer of a DMAC channel, the hardware DMA transfer request selection/hold circuit of the DTFR is still running, and consequently, the DTFR may retain a hardware DMA transfer request that was input to the DTFR during the suspension or transfer abort period of the DMAC channel. When you resume or start DMA transfer of a DMAC channel, clear the hardware DMA transfer request retained in the DTFR as required.

In case that DMAC is used both with hardware DMA transfer request and with block transfer (1 or 2) mode, if DTFR hardware DMA transfer source selection enable bit is set to disable (PDMAnDTFRm.REQEN = 0) by software while DMAC is executing block transfer, the ongoing block transfer is suspended.

8.4.4 List of Suspend, Resume, and Transfer Abort Functions

Table 8.20 List of Suspend, Resume, and Transfer Abort Functions

Function	How to Execute the Function	Operation	Possibility of DMA Transfer Abort	Master that can Execute the Function (See **Section 8.6, Reliability Functions**.)
DMA suspension and resume by software control	Setting and clearing the PDMAnDMACTL.DMASPD	All channels are in the suspended state.	Not possible*[1]	Special master
Suspension and resume of a DMAC channel	Clearing and setting the PDMAnDCENm.DTE in each channel register*[2]	DMA transfer of a channel is suspended.	Possible (by clearing the DMA transfer request flag in suspension state)	Special master, and general master assigned to the channel.

Note 1. In order to abort DMA transfer, you need to abort transfer for the DMAC channel.

Note 2. In case that the continuous transfer enable bit (PDMAnDTCTm.MLE) is set, please clear the continuous transfer enable bit (PDMAnDTCTm.MLE) first before clearing the PDMAnDCENm.DTE bit to suspend the channel and set the continuous transfer enable bit (PDMAnDTCTm.MLE) before setting the PDMAnDCENm.DTE bit to resume the channel.

8.5 Error Control

8.5.1 Type of Error

DMA can generate the following type of error.

- DMA Transfer Error
  This error is generated when error is detected in the read cycle or write cycle in a DMA cycle. This error can be generated in all DMAC channels during execution of DMA transfer.
  A DMA transfer error leads to INTDMAERR interrupt request which is a source of FEINT.

8.5.2 DMA Transfer Error

8.5.2.1 Operation of a DMAC When DMA Transfer Error Occurs

When DMA transfer error occurs in a DMAC, the transfer error flag (PDMAnDCSTm.ER) in the DMAC transfer status register of the channel where the DMA transfer error occurred is set. The DMAC error register (PDMAnDMACER) shows the transfer error flags of all 32 DMAC channels.

On a channel where the transfer error flag is set, a new DMA cycle is not executed if the transfer disable on transfer error setting (PDMAnDTCTm.ESE) bit is set. On the other hand, a DMA cycle is executed regardless of the value of the transfer error flag if the transfer error case DMA transfer disable setting (PDMAnDTCTm.ESE) bit is cleared.

If you want to abort the DMA transfer of a channel where the DMA transfer error occurred is set, follow the procedure to abort DMA transfer of the DMAC channel.

If DMA transfer error occurs during the read cycle of a DMA cycle, the write cycle is not executed. If DMA transfer error occurs during the write cycle of a DMA cycle, the validity of the result of write operation is not guaranteed.

Regardless of whether DMA transfer error occurs in the read cycle or write cycle of a DMA cycle, the source address register, destination address register, transfer count register, and address reload count register are updated.

8.6 Reliability Functions

8.6.1 Overview

In this product, DMA provides the following reliability functions:

- Register access protection function
- Master information inherit function

8.6.2 Register Access Protection Function

- RH850/F1KH-D8
  This product is designed to assign each DMA channel to CPU1 or CPU2.
- RH850/F1KM-S4, RH850/F1KM-S1
  This product is designed to assign DMA channel to a CPU1.

The register access protection function allows write access to the transfer information of each DMA channel only from the master assigned to the channel and prohibits write access from other masters.

The register access protection function enables you, for example, to prevent the settings of the channel from being updated by masters other than the one assigned to the channel.

8.6.2.1 Identifying the Accessing Master

DMA identifies a master based on the processor element ID number (PEID) of the accessing master, the system protection ID configured by the accessing CPU (SPID), and whether the CPU is in the supervisor mode (PSW.UM = 0) or the user mode (PSW.UM = 1).

8.6.2.2 Master Access

DMA handles accesses from CPU in the supervisor mode (PSW.UM = 0) as accesses from special masters. Special masters are allowed to read from or write to all DMA registers.

DMA handles the other masters except special masters as general masters. General masters are allowed to read from all DMA registers, but allowed to write to the following specific registers.

- Channel registers of the channels assigned by the channel assignment. (For details, see **Section 8.6.2.3, Channel Assignment**.)

From general masters, write access to registers other than the above is not allowed.

8.6.2.3 Channel Assignment

To each channel, DMA can assign a master so that the master is allowed to use the channel. Channel assignment is configured in the channel master setting register (PDMAnDMyiCM) by the CPU in the supervisor mode (PSW.UM = 0).

In general master access, the master assigned to a channel by the channel assignment is allowed to write the channel registers of the channel. If the channel registers of a channel is written by a master other than the master assigned to the channel, the access is called illegal access. For information about illegal access, see **Section 8.6.2.4, Illegal Access**.

8.6.2.4 Illegal Access

DMA handles the following access as illegal access.

(1) Write access to global registers from general masters

(2) Write access to channel registers from general masters who are not assigned to the channel

DMA never treat read access from any master as illegal access.

DMA’s actions against illegal access are as follows:

For both cases (1) and (2),

- Write access is ignored.

Only for the case (2),

- The information about the illegal access is stored in a register access protection violation register.
- The DMAC0, DMAC1, DMAC2, and DMAC3 have their own register access protection violation registers (PDMAnDM0CMV, PDMAnDM1CMV, PDMAnDM2CMV, and PDMAnDM3CMV respectively).

Only the special master can access the register access protection violation registers. The special master can check whether illegal access has occurred by checking the register access protection violation registers periodically.

In addition, it is recommended that, when a master uses DMA and configures transfer information in the channel registers, the master should check whether the configuration has been successfully completed without illegal access by, for example, reading back the settings.

8.6.3 Master Information Inherit Function

In this product, DMA inherits master information that is equivalent to the master information of the master to which the DMA channel is assigned.

The master information that is output from DMA is shown in **Table 8.21, Master Information that is Output from DMA**.

Table 8.21 Master Information that is Output from DMA

Meaning	Value that is Output from DMA
UM	UM bit value in the channel master setting register
SPID	SPID bit value in the channel master setting register
PEID	PEID bit value in the channel master setting register
DMA	1

8.6.4 Other Reliability Functions

8.6.4.1 Restriction on the Next Channel in the Chain

The reliability function limits the channels you can select as the next channel in the chain.

When you use the chain function, the channel master settings of a channel and its next channel in the chain must be the same.

The chain function is designed so that a channel and its next channel in the chain are managed by the same master.

When DMA detects that different masters are assigned to a channel and its next channel in the chain, it is deemed illegal and the chain function is suppressed. More specifically, when DMA tries to execute the chain function, DMA compares the chain master settings of the channel and its next channel in the chain, and if the settings are the same for both PEID and UM, the chain function is allowed and a chain request is sent to the next channel. If the channel master settings are not the same for either PEID or UM, a chain request is not sent.

8.7 Setting Up DMA Transfer

8.7.1 Overview of Setting Up DMA

Table 8.22 Overview of Setting Up DMA

No.	Master that Configures the Setting	Description	Register		Necessity of the Setting
1	Special master (CPU1/2 in the supervisor mode (UM = 0))	Overall DMA operation setting	PDMAnDM00CM to PDMAnDM37CM	DMAC channel master setting	Mandatory
2		Status clear	PDMAnCMVC	Channel protection violation clear register	Recommended
3	Master assigned to the DMAC channel	Channel setting	PDMAnDSAm	DMAC source address	Mandatory
4			PDMAnDDAm	DMAC destination address	Mandatory
5			PDMAnDTCm	DMAC transfer count	Mandatory
6			PDMAnDTCTm	DMAC transfer control	Mandatory
7			PDMAnDRSAm	DMAC reload source address	Mandatory if the reload function is used
8			PDMAnDRDAm	DMAC reload destination address	Mandatory if the reload function is used
9			PDMAnDRTCm	DMAC reload transfer count	Mandatory if the reload function is used
10			PDMAnDTFRm	DTFR setting register	Mandatory
11		Status clear	PDMAnDCSTCm	DMAC transfer status clear	Mandatory
12			PDMAnDTFRRQCm	DTFR transfer request clear	Recommended
13		Channel operation enable	PDMAnDCENm	DMAC channel operation enable setting	Mandatory

8.7.2 Setting Up the Overall DMA Operation

You need to set up the overall DMA operation before you start using DMA.

To configure the overall DMA operation, the special master (CPU1/2 in the supervisor mode (UM = 0)) needs to set up global registers. Global registers can be set up only by special master access. For details, see **Section 8.6, Reliability Functions**.

The following register must be set up to configure the overall DMA operation.

- DMAC channel master setting registers (PDMAnDMyiCM)

These registers configure channel assignment. (For details, see **Section 8.6, Reliability Functions**.)

If the DMAC channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

Also, if errors are detected in the following registers while the overall DMA operation is set up, clearing the errors is recommended.

- DMAC0 register access protection violation register (PDMAnDM0CMV)
- DMAC1 register access protection violation register (PDMAnDM1CMV)
- DMAC2 register access protection violation register (PDMAnDM2CMV)
- DMAC3 register access protection violation register (PDMAnDM3CMV)

8.7.3 Setting Up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC.

To configure the DMA channel setting, each channel’s master assigned by the channel assignment sets channel registers.

8.7.3.1 Setting Up the DMAC Channel Setting

Follow the procedure below to set up the DMAC channel setting in case of using DMAC.

(1) Disabling the DMAC Channel Operation

If the channel operation enable (PDMAnDCENm.DTE) in the DMAC channel operation enable setting register (PDMAnDCENm) is set, clear the PDMAnDCENm.DTE bit to disable the channel operation.

(2) Setting Up the Transfer Information

When you set up the transfer information of the DMAC, the following registers need to be set up.

- DMAC source address register (PDMAnDSAm)
- DMAC destination address register (PDMAnDDAm)
- DMAC transfer count register (PDMAnDTCm)
- DMAC transfer control register (PDMAnDTCTm)
- DMAC reload source address register (PDMAnDRSAm)
- DMAC reload destination address register (PDMAnDRDAm)
- DMAC reload transfer count register (PDMAnDRTCm)

(3) Setting Up the DMA Transfer Request

While setting the transfer information, you need to set up the DMA transfer request selection assignment (PDMAnDTCTm.DRS) bit in the DMAC transfer control register (PDMAnDTCTm) to define whether the hardware or software DMA transfer request is used.

You cannot use both the hardware and software DMA transfer requests for the same channel at the same time.

If you use the hardware DMA transfer request, you need to select the source used as the hardware DMA transfer request out of 128 hardware DMA transfer sources using the hardware DMA transfer source selection (PDMAnDTFRm.REQSEL) in the DTFR setting register. Also, you need to enable the hardware DMA transfer source selection (PDMAnDTFRm.REQEN) in the same register.

The DTFR may retain a hardware DMA transfer request before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (PDMAnDTFRRQm.DRQ) retained in the DTFR using the DTFR transfer request clear register (PDMAnDTFRRQCm) if necessary.

If you use the software DMA transfer request, disable the hardware DMA transfer source selection (PDMAnDTFRm.REQEN) in the DTFR setting register.

(4) Clearing the Transfer Status

The DMAC transfer status register (PDMAnDCSTm) may retain the result of the previous DMA transfer, so clear the flags in the DMAC transfer status register using the DMAC transfer status clear register (PDMAnDCSTCm).

(5) Enabling the DMAC Channel Operation

Set the channel operation enable (PDMAnDCENm.DTE) bit in the DMAC channel operation enable setting register to enable the channel operation.

After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and become ready for DMA transfer.

8.8 Global Registers

8.8.1 List of Global Register Address

The global register addresses are listed in the table below.

For <DMAn_base>, see **Section 8.1.2, Register Base Addresses**.

Table 8.23 List of Global Register Address (RH850/F1KH-D8)

				Access Permission	
Unit Name	Register Name	Symbol	Address	Special Master	General Master
PDMAn	DMA control register	PDMAnDMACTL	<DMAn_base> + 0000_H	✓	—[*1]
	DMAC error register	PDMAnDMACER	<DMAn_base> + 0020_H	✓	✓
	DMAC0 register access protection violation register	PDMAnDM0CMV	<DMAn_base> + 0030_H	✓	✓
	DMAC1 register access protection violation register	PDMAnDM1CMV	<DMAn_base> + 0034_H	✓	✓
	Register access protection violation clear register	PDMAnCMVC	<DMAn_base> + $003C_H$	✓	—[*1]
	DMAC2 register access protection violation register	PDMAnDM2CMV	<DMAn_base> + 0040_H	✓	✓
	DMAC3 register access protection violation register	PDMAnDM3CMV	<DMAn_base> + 0044_H	✓	✓
	DMAC0 channel 0 channel master setting	PDMAnDM00CM	<DMAn_base> + 0100_H	✓	—[*1]
	DMAC0 channel 1 channel master setting	PDMAnDM01CM	<DMAn_base> + 0104_H	✓	—[*1]
	DMAC0 channel 2 channel master setting	PDMAnDM02CM	<DMAn_base> + 0108_H	✓	—[*1]
	DMAC0 channel 3 channel master setting	PDMAnDM03CM	<DMAn_base> + $010C_H$	✓	—[*1]
	DMAC0 channel 4 channel master setting	PDMAnDM04CM	<DMAn_base> + 0110_H	✓	—[*1]
	DMAC0 channel 5 channel master setting	PDMAnDM05CM	<DMAn_base> + 0114_H	✓	—[*1]
	DMAC0 channel 6 channel master setting	PDMAnDM06CM	<DMAn_base> + 0118_H	✓	—[*1]
	DMAC0 channel 7 channel master setting	PDMAnDM07CM	<DMAn_base> + $011C_H$	✓	—[*1]
	DMAC1 channel 0 channel master setting	PDMAnDM10CM	<DMAn_base> + 0120_H	✓	—[*1]
	DMAC1 channel 1 channel master setting	PDMAnDM11CM	<DMAn_base> + 0124_H	✓	—[*1]
	DMAC1 channel 2 channel master setting	PDMAnDM12CM	<DMAn_base> + 0128_H	✓	—[*1]
	DMAC1 channel 3 channel master setting	PDMAnDM13CM	<DMAn_base> + $012C_H$	✓	—[*1]
	DMAC1 channel 4 channel master setting	PDMAnDM14CM	<DMAn_base> + 0130_H	✓	—[*1]
	DMAC1 channel 5 channel master setting	PDMAnDM15CM	<DMAn_base> + 0134_H	✓	—[*1]
	DMAC1 channel 6 channel master setting	PDMAnDM16CM	<DMAn_base> + 0138_H	✓	—[*1]
	DMAC1 channel 7 channel master setting	PDMAnDM17CM	<DMAn_base> + $013C_H$	✓	—[*1]
	DMAC2 channel 0 channel master setting	PDMAnDM20CM	<DMAn_base> + 0140_H	✓	—[*1]
	DMAC2 channel 1 channel master setting	PDMAnDM21CM	<DMAn_base> + 0144_H	✓	—[*1]
	DMAC2 channel 2 channel master setting	PDMAnDM22CM	<DMAn_base> + 0148_H	✓	—[*1]
	DMAC2 channel 3 channel master setting	PDMAnDM23CM	<DMAn_base> + $014C_H$	✓	—[*1]
	DMAC2 channel 4 channel master setting	PDMAnDM24CM	<DMAn_base> + 0150_H	✓	—[*1]
	DMAC2 channel 5 channel master setting	PDMAnDM25CM	<DMAn_base> + 0154_H	✓	—[*1]
	DMAC2 channel 6 channel master setting	PDMAnDM26CM	<DMAn_base> + 0158_H	✓	—[*1]
	DMAC2 channel 7 channel master setting	PDMAnDM27CM	<DMAn_base> + $015C_H$	✓	—[*1]
	DMAC3 channel 0 channel master setting	PDMAnDM30CM	<DMAn_base> + 0160_H	✓	—[*1]
	DMAC3 channel 1 channel master setting	PDMAnDM31CM	<DMAn_base> + 0164_H	✓	—[*1]
	DMAC3 channel 2 channel master setting	PDMAnDM32CM	<DMAn_base> + 0168_H	✓	—[*1]
	DMAC3 channel 3 channel master setting	PDMAnDM33CM	<DMAn_base> + $016C_H$	✓	—[*1]
	DMAC3 channel 4 channel master setting	PDMAnDM34CM	<DMAn_base> + 0170_H	✓	—[*1]
	DMAC3 channel 5 channel master setting	PDMAnDM35CM	<DMAn_base> + 0174_H	✓	—[*1]

Table 8.23 List of Global Register Address (RH850/F1KH-D8)

Unit Name	Register Name	Symbol	Address	Access Permission	
				Special Master	General Master
PDMAn	DMAC3 channel 6 channel master setting	PDMAnDM36CM	<DMAn_base> + 0178_H	✓	—[*1]
	DMAC3 channel 7 channel master setting	PDMAnDM37CM	<DMAn_base> + $017C_H$	✓	—[*1]

Note 1. The registers are read only for general master.

Table 8.24 List of Global Register Address (RH850/F1KM-S4)

Unit Name	Register Name	Symbol	Address	Access Permission	
				Special Master	General Master
PDMAn	DMA control register	PDMAnDMACTL	<DMAn_base> + 0000_H	✓	—[*1]
	DMAC error register	PDMAnDMACER	<DMAn_base> + 0020_H	✓	✓
	DMAC0 register access protection violation register	PDMAnDM0CMV	<DMAn_base> + 0030_H	✓	✓
	DMAC1 register access protection violation register	PDMAnDM1CMV	<DMAn_base> + 0034_H	✓	✓
	Register access protection violation clear register	PDMAnCMVC	<DMAn_base> + $003C_H$	✓	—[*1]
	DMAC2 register access protection violation register	PDMAnDM2CMV	<DMAn_base> + 0040_H	✓	✓
	DMAC3 register access protection violation register	PDMAnDM3CMV	<DMAn_base> + 0044_H	✓	✓
	DMAC0 channel 0 channel master setting	PDMAnDM00CM	<DMAn_base> + 0100_H	✓	—[*1]
	DMAC0 channel 1 channel master setting	PDMAnDM01CM	<DMAn_base> + 0104_H	✓	—[*1]
	DMAC0 channel 2 channel master setting	PDMAnDM02CM	<DMAn_base> + 0108_H	✓	—[*1]
	DMAC0 channel 3 channel master setting	PDMAnDM03CM	<DMAn_base> + $010C_H$	✓	—[*1]
	DMAC0 channel 4 channel master setting	PDMAnDM04CM	<DMAn_base> + 0110_H	✓	—[*1]
	DMAC0 channel 5 channel master setting	PDMAnDM05CM	<DMAn_base> + 0114_H	✓	—[*1]
	DMAC0 channel 6 channel master setting	PDMAnDM06CM	<DMAn_base> + 0118_H	✓	—[*1]
	DMAC0 channel 7 channel master setting	PDMAnDM07CM	<DMAn_base> + $011C_H$	✓	—[*1]
	DMAC1 channel 0 channel master setting	PDMAnDM10CM	<DMAn_base> + 0120_H	✓	—[*1]
	DMAC1 channel 1 channel master setting	PDMAnDM11CM	<DMAn_base> + 0124_H	✓	—[*1]
	DMAC1 channel 2 channel master setting	PDMAnDM12CM	<DMAn_base> + 0128_H	✓	—[*1]
	DMAC1 channel 3 channel master setting	PDMAnDM13CM	<DMAn_base> + $012C_H$	✓	—[*1]
	DMAC1 channel 4 channel master setting	PDMAnDM14CM	<DMAn_base> + 0130_H	✓	—[*1]
	DMAC1 channel 5 channel master setting	PDMAnDM15CM	<DMAn_base> + 0134_H	✓	—[*1]
	DMAC1 channel 6 channel master setting	PDMAnDM16CM	<DMAn_base> + 0138_H	✓	—[*1]
	DMAC1 channel 7 channel master setting	PDMAnDM17CM	<DMAn_base> + $013C_H$	✓	—[*1]
	DMAC2 channel 0 channel master setting	PDMAnDM20CM	<DMAn_base> + 0140_H	✓	—[*1]
	DMAC2 channel 1 channel master setting	PDMAnDM21CM	<DMAn_base> + 0144_H	✓	—[*1]
	DMAC2 channel 2 channel master setting	PDMAnDM22CM	<DMAn_base> + 0148_H	✓	—[*1]
	DMAC2 channel 3 channel master setting	PDMAnDM23CM	<DMAn_base> + $014C_H$	✓	—[*1]
	DMAC2 channel 4 channel master setting	PDMAnDM24CM	<DMAn_base> + 0150_H	✓	—[*1]
	DMAC2 channel 5 channel master setting	PDMAnDM25CM	<DMAn_base> + 0154_H	✓	—[*1]
	DMAC2 channel 6 channel master setting	PDMAnDM26CM	<DMAn_base> + 0158_H	✓	—[*1]
	DMAC2 channel 7 channel master setting	PDMAnDM27CM	<DMAn_base> + $015C_H$	✓	—[*1]
	DMAC3 channel 0 channel master setting	PDMAnDM30CM	<DMAn_base> + 0160_H	✓	—[*1]
	DMAC3 channel 1 channel master setting	PDMAnDM31CM	<DMAn_base> + 0164_H	✓	—[*1]

Table 8.24 List of Global Register Address (RH850/F1KM-S4)

Unit Name	Register Name	Symbol	Address	Access Permission	
				Special Master	General Master
PDMAn	DMAC3 channel 2 channel master setting	PDMAnDM32CM	<DMAn_base> + 0168_H	✓	—*1
	DMAC3 channel 3 channel master setting	PDMAnDM33CM	<DMAn_base> + $016C_H$	✓	—*1
	DMAC3 channel 4 channel master setting	PDMAnDM34CM	<DMAn_base> + 0170_H	✓	—*1
	DMAC3 channel 5 channel master setting	PDMAnDM35CM	<DMAn_base> + 0174_H	✓	—*1
	DMAC3 channel 6 channel master setting	PDMAnDM36CM	<DMAn_base> + 0178_H	✓	—*1
	DMAC3 channel 7 channel master setting	PDMAnDM37CM	<DMAn_base> + $017C_H$	✓	—*1

Note 1. The registers are read only for general master.

Table 8.25 List of Global Register Address (RH850/F1KM-S1)

Unit Name	Register Name	Symbol	Address	Access Permission	
				Special Master	General Master
PDMAn	DMA control register	PDMAnDMACTL	<DMAn_base> + 0000_H	✓	—*1
	DMAC error register	PDMAnDMACER	<DMAn_base> + 0020_H	✓	✓
	DMAC0 register access protection violation register	PDMAnDM0CMV	<DMAn_base> + 0030_H	✓	✓
	DMAC1 register access protection violation register	PDMAnDM1CMV	<DMAn_base> + 0034_H	✓	✓
	Register access protection violation clear register	PDMAnCMVC	<DMAn_base> + $003C_H$	✓	—*1
	DMAC0 channel 0 channel master setting	PDMAnDM00CM	<DMAn_base> + 0100_H	✓	—*1
	DMAC0 channel 1 channel master setting	PDMAnDM01CM	<DMAn_base> + 0104_H	✓	—*1
	DMAC0 channel 2 channel master setting	PDMAnDM02CM	<DMAn_base> + 0108_H	✓	—*1
	DMAC0 channel 3 channel master setting	PDMAnDM03CM	<DMAn_base> + $010C_H$	✓	—*1
	DMAC0 channel 4 channel master setting	PDMAnDM04CM	<DMAn_base> + 0110_H	✓	—*1
	DMAC0 channel 5 channel master setting	PDMAnDM05CM	<DMAn_base> + 0114_H	✓	—*1
	DMAC0 channel 6 channel master setting	PDMAnDM06CM	<DMAn_base> + 0118_H	✓	—*1
	DMAC0 channel 7 channel master setting	PDMAnDM07CM	<DMAn_base> + $011C_H$	✓	—*1
	DMAC1 channel 0 channel master setting	PDMAnDM10CM	<DMAn_base> + 0120_H	✓	—*1
	DMAC1 channel 1 channel master setting	PDMAnDM11CM	<DMAn_base> + 0124_H	✓	—*1
	DMAC1 channel 2 channel master setting	PDMAnDM12CM	<DMAn_base> + 0128_H	✓	—*1
	DMAC1 channel 3 channel master setting	PDMAnDM13CM	<DMAn_base> + $012C_H$	✓	—*1
	DMAC1 channel 4 channel master setting	PDMAnDM14CM	<DMAn_base> + 0130_H	✓	—*1
	DMAC1 channel 5 channel master setting	PDMAnDM15CM	<DMAn_base> + 0134_H	✓	—*1
	DMAC1 channel 6 channel master setting	PDMAnDM16CM	<DMAn_base> + 0138_H	✓	—*1
	DMAC1 channel 7 channel master setting	PDMAnDM17CM	<DMAn_base> + $013C_H$	✓	—*1

Note 1. The registers are read only for general master.

8.8.2 Details of Global Registers

8.8.2.1 PDMAnDMACTL — DMA Control Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA SPD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.26 PDMAnDMACTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DMASPD	DMA suspension This bit indicates that DMA transfer for all channels is suspended. If a user writes 1 to this bit, DMA transfer for all channels can be suspended. If a user writes 0 to this bit, suspension of DMA transfer for all channels can be released. The suspension controlled by this bit is independent from the suspension controlled by the transfer enable bit (PDMAnDCENm.DTE) of each DMAC channel. That means, if this bit is set to 1, all DMA transfers are suspended regardless of the values of the PDMAnDCENm.DTE bit of each DMAC channel. Writing to this bit does not affect the PDMAnDCENm.DTE bit of each DMAC channel. 0: DMA suspension cleared 1: DMA suspension request/DMA suspension ongoing

8.8.2.2 PDMAnDMACER — DMAC Error Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0020$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DM3 ER7	DM3 ER6	DM3 ER5	DM3 ER4	DM3 ER3	DM3 ER2	DM3 ER1	DM3 ER0	DM2 ER7	DM2 ER6	DM2 ER5	DM2 ER4	DM2 ER3	DM2 ER2	DM2 ER1	DM2 ER0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM1 ER7	DM1 ER6	DM1 ER5	DM1 ER4	DM1 ER3	DM1 ER2	DM1 ER1	DM1 ER0	DM0 ER7	DM0 ER6	DM0 ER5	DM0 ER4	DM0 ER3	DM0 ER2	DM0 ER1	DM0 ER0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.27 PDMAnDMACER Register Contents

Bit Position	Bit Name	Function
31 to 24	DM3ER[7:0]	RH850F1KH-D8, RH850/F1KM-S4 DMAC3 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC3. Each bit is mapped from the PDMAnDCSTm.ER bit of each channel of the DMAC3 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated RH850/F1KM-S1: When read, the value after reset is returned.
23 to 16	DM2ER[7:0]	RH850F1KH-D8, RH850/F1KM-S4 DMAC2 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC2. Each bit is mapped from the PDMAnDCSTm.ER bit of each channel of the DMAC2 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated RH850/F1KM-S1: When read, the value after reset is returned.
15 to 8	DM1ER[7:0]	DMAC1 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC1. Each bit is mapped from the PDMAnDCSTm.ER bit of each channel of the DMAC1 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated
7 to 0	DM0ER[7:0]	DMAC0 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC0. Each bit is mapped from the PDMAnDCSTm.ER bit of each channel of the DMAC0 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated

8.8.2.3 PDMAnDM0CMV — DMAC0 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.28 PDMAnDM0CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 17	PEID[2:0] SPID[1:0] UM	Illegal access master information These bits retain the accessing master information of the first illegal access after the PDMAnDM0CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM0CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred. These bits retain the channel number (0 to 7) of the first illegal access after the PDMAnDM0CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM0CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC0. 0: No illegal access has occurred in the DMAC0 1: Illegal access has occurred in the DMAC0 If illegal access occurs in the DMAC0 while this bit is 0, this bit is set, and PDMAnDM0CMV.PEID[2:0], PDMAnDM0CMV.SPID[1:0], PDMAnDM0CMV.UM and PDMAnDM0CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC0 while this bit is 1, this bit remains 1, and the contents of PDMAnDM0CMV.PEID[2:0], PDMAnDM0CMV.SPID[1:0], PDMAnDM0CMV.UM and PDMAnDM0CMV.VCH[2:0] do not change. This bit can be cleared by using the PDMAnCMVC register.

8.8.2.4 PDMAnDM1CMV — DMAC1 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.29 PDMAnDM1CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 17	PEID[2:0] SPID[1:0] UM	Illegal access master information These bits retain the accessing master information of the first illegal access after the PDMAnDM1CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM1CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred. These bits retain the channel number (0 to 7) of the first illegal access after the PDMAnDM1CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM1CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC1. 0: No illegal access has occurred in the DMAC1 1: Illegal access has occurred in the DMAC1 If illegal access occurs in the DMAC1 while this bit is 0, this bit is set, and PDMAnDM1CMV.PEID[2:0], PDMAnDM1CMV.SPID[1:0], PDMAnDM1CMV.UM and PDMAnDM1CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC1 while this bit is 1, this bit remains 1, and the contents of PDMAnDM1CMV.PEID[2:0], PDMAnDM1CMV.SPID[1:0], PDMAnDM1CMV.UM and PDMAnDM1CMV.VCH[2:0] do not change. This bit can be cleared by using the PDMAnCMVC register.

8.8.2.5 PDMAnDM2CMV — DMAC2 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.30 PDMAnDM2CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 17	PEID[2:0] SPID[1:0] UM	Illegal access master information These bits retain the accessing master information of the first illegal access after the PDMAnDM2CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM2CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred. These bits retain the channel number (0 to 7) of the first illegal access after the PDMAnDM2CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM2CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC2. 0: No illegal access has occurred in the DMAC2 1: Illegal access has occurred in theDMAC2 If illegal access occurs in the DMAC2 while this bit is 0, this bit is set, and PDMAnDM2CMV.PEID[2:0], PDMAnDM2CMV.SPID[1:0], PDMAnDM2CMV.UM and PDMAnDM2CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC2 while this bit is 1, this bit remains 1, and the contents of PDMAnDM2CMV.PEID[2:0], PDMAnDM2CMV.SPID[1:0], PDMAnDM2CMV.UM and PDMAnDM2CMV.VCH[2:0] do not change. This bit can be cleared by using the PDMAnCMVC register.

8.8.2.6 PDMAnDM3CMV — DMAC3 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0044$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.31 PDMAnDM3CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 17	PEID[2:0] SPID[1:0] UM	Illegal access master information These bits retain the accessing master information of the first illegal access after the PDMAnDM3CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM3CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred. These bits retain the channel number (0 to 7) of the first illegal access after the PDMAnDM3CMV.VF bit is cleared to 0. If illegal access occurs while the PDMAnDM3CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC3. 0: No illegal access has occurred in the DMAC3 1: Illegal access has occurred in the DMAC3 If illegal access occurs in the DMAC3 while this bit is 0, this bit is set, and PDMAnDM3CMV.PEID[2:0], PDMAnDM3CMV.SPID[1:0], PDMAnDM3CMV.UM and PDMAnDM3CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC3 while this bit is 1, this bit remains 1, and the contents of PDMAnDM3CMV.PEID[2:0], PDMAnDM3CMV.SPID[1:0], PDMAnDM3CMV.UM and PDMAnDM3CMV.VCH[2:0] do not change. This bit can be cleared by using the PDMAnCMVC register.

8.8.2.7 PDMAnCMVC — Register Access Protection Violation Clear Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 003C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DM3VC	DM2VC	—	—	DM1VC	DM0VC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 8.32 PDMAnCMVC Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DM3VC	RH850/F1KH-D8, RH850/F1KM-S4: DMAC3 illegal access flag clear The DMAC3 illegal access flag (PDMAnDM3CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0. RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
4	DM2VC	RH850/F1KH-D8, RH850/F1KM-S4: DMAC2 illegal access flag clear The DMAC2 illegal access flag (PDMAnDM2CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0. RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DM1VC	DMAC1 illegal access flag clear The DMAC1 illegal access flag (PDMAnDM1CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
0	DM0VC	DMAC0 illegal access flag clear The DMAC0 illegal access flag (PDMAnDM0CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.

8.8.2.8 PDMAnDMyiCM — DMAC Channel Master Setting (yi = 00 to 07, 10 to 17, 20 to 27, and 30 to 37 in RH850/F1KH-D8, yi = 00 to 07, 10 to 17, 20 to 27, and 30 to 37 in RH850/F1KM-S4, yi = 00 to 07 and 10 to 17 in RH850/F1KM-S1)

Access:	This register can be read or written in 32-bit units.
Address:	PDMAnDM0iCM: <DMAn_base> + 0100_H + 4_H × Ch. No. i (i = 0 to 7) PDMAnDM1iCM: <DMAn_base> + 0120_H + 4_H × Ch. No. i (i = 0 to 7) PDMAnDM2iCM: <DMAn_base> + 0140_H + 4_H × Ch. No. i (i = 0 to 7) PDMAnDM3iCM: <DMAn_base> + 0160_H + 4_H × Ch. No. i (i = 0 to 7)
Value after reset:	0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 8.33 PDMAnDMyiCM Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	PEID[2:0]	Channel master PEID setting Specifies the PEID information of the master assigned to the channel. Set PEID value of PE that configures this register.
3, 2	SPID[1:0]	Channel master SPID setting Specifies the SPID information used by the master assigned to the channel.
1	UM	Channel master UM setting 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

PDMAnDM00CM to PDMAnDM07CM configure the channel master information of the DMAC0 channel 0 to 7 respectively.
PDMAnDM10CM to PDMAnDM17CM configure the channel master information of the DMAC1 channel 0 to 7 respectively.
PDMAnDM20CM to PDMAnDM27CM configure the channel master information of the DMAC2 channel 0 to 7 respectively.
PDMAnDM30CM to PDMAnDM37CM configure the channel master information of the DMAC3 channel 0 to 7 respectively.

For information about the functions this register offers, see **Section 8.6, Reliability Functions**.

8.9 DMAC Channel Registers

8.9.1 DMAC Channel Register Addresses

The DMAC channel registers are listed in the table below.

For <DMAn_base>, see **Section 8.1.2, Register Base Addresses**.

Table 8.34 DMAC Channel Register Addresses

Module Name	Register Name	Symbol	Address	Access Permission	
				Special Master	General Master
PDMAn	DMAC source address	PDMAnDSAm	<DMAn_base> + 0400_H + 40_H × [channel number]	✓	✓
	DMAC destination address	PDMAnDDAm	<DMAn_base> + 0404_H + 40_H × [channel number]	✓	✓
	DMAC transfer count	PDMAnDTCm	<DMAn_base> + 0408_H + 40_H × [channel number]	✓	✓
	DMAC transfer control	PDMAnDTCTm	<DMAn_base> + $040C_H$ + 40_H × [channel number]	✓	✓
	DMAC reload source address	PDMAnDRSAm	<DMAn_base> + 0410_H + 40_H × [channel number]	✓	✓
	DMAC reload destination address	PDMAnDRDAm	<DMAn_base> + 0414_H + 40_H × [channel number]	✓	✓
	DMAC reload transfer count	PDMAnDRTCm	<DMAn_base> + 0418_H + 40_H × [channel number]	✓	✓
	DMAC channel operation enable setting	PDMAnDCENm	<DMAn_base> + 0420_H + 40_H × [channel number]	✓	✓
	DMAC transfer status	PDMAnDCSTm	<DMAn_base> + 0424_H + 40_H × [channel number]	✓	✓
	DMAC transfer status set	PDMAnDCSTSm	<DMAn_base> + 0428_H + 40_H × [channel number]	✓	✓
	DMAC transfer status clear	PDMAnDCSTCm	<DMAn_base> + $042C_H$ + 40_H × [channel number]	✓	✓
	DTFR setting	PDMAnDTFRm	<DMAn_base> + 0430_H + 40_H × [channel number]	✓	✓
	DTFR transfer request status	PDMAnDTFRRQm	<DMAn_base> + 0434_H + 40_H × [channel number]	✓	✓
	DTFR transfer request clear	PDMAnDTFRRQCm	<DMAn_base> + 0438_H + 40_H × [channel number]	✓	✓

Note: The channel number in the offset addresses and “m” in the register symbols are numbers in the range from 0 to 31 in RH850/F1KH-D8, RH850/F1KM-S4, from 0 to 15 in RH850/F1KM-S1. The correspondence is as follows.

Channel Number m	Channel
0	DMAC0 channel 0
1	DMAC0 channel 1
2	DMAC0 channel 2
3	DMAC0 channel 3
4	DMAC0 channel 4
5	DMAC0 channel 5
6	DMAC0 channel 6
7	DMAC0 channel 7
8	DMAC1 channel 0
9	DMAC1 channel 1
10	DMAC1 channel 2
11	DMAC1 channel 3
12	DMAC1 channel 4
13	DMAC1 channel 5
14	DMAC1 channel 6
15	DMAC1 channel 7
16	DMAC2 channel 0
17	DMAC2 channel 1
18	DMAC2 channel 2
19	DMAC2 channel 3
20	DMAC2 channel 4
21	DMAC2 channel 5
22	DMAC2 channel 6
23	DMAC2 channel 7
24	DMAC3 channel 0
25	DMAC3 channel 1
26	DMAC3 channel 2
27	DMAC3 channel 3
28	DMAC3 channel 4
29	DMAC3 channel 5
30	DMAC3 channel 6
31	DMAC3 channel 7

8.9.2 Details of DMAC Channel Registers

8.9.2.1 PDMAnDSAm — DMAC Source Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0400_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SA31	SA30	SA29	SA28	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.35 PDMAnDSAm Register Contents

Bit Position	Bit Name	Function
31 to 0	SA[31:0]	Source address Specifies the DMA transfer source address. These bits are updated whenever a DMA cycle is executed. If you read from these bits, the transfer source address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to these bits when the channel operation is enabled (PDMAnDCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the source address is updated.
3. DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary bit.)
   The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	SA3	SA2	SA1	SA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.2 PDMAnDDAm — DMAC Destination Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0404_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.36 PDMAnDDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	DA[31:0]	Destination address Specifies the DMA transfer destination address. These bits are updated whenever a DMA cycle is executed. If you read from these bits, the transfer destination address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to these bits when the channel operation is enabled (PDMAnDCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the destination address is updated.
3. DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary bit.)
   The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	DA3	DA2	DA1	DA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.3 PDMAnDTCm — DMAC Transfer Count Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0408_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARC15	ARC14	ARC13	ARC12	ARC11	ARC10	ARC9	ARC8	ARC7	ARC6	ARC5	ARC4	ARC3	ARC2	ARC1	ARC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRC15	TRC14	TRC13	TRC12	TRC11	TRC10	TRC9	TRC8	TRC7	TRC6	TRC5	TRC4	TRC3	TRC2	TRC1	TRC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.37 PDMAnDTCm Register Contents

Bit Position	Bit Name	Function
31 to 16	ARC[15:0]	Address reload count Specifies the number of transfers until the address reload when the reload function 2 is used, and also specifies the number of transfers when the block transfer 2 is used. If you read from those bits during DMA transfer, the address reload count for the next DMA cycle is read. When the reload function 2 or block transfer 2 is used, PDMAnDTCm.ARC[15:0] is decremented by one for every DMA cycle. When the reload function 2 or block transfer 2 is not used, PDMAnDTCm.ARC[15:0] is not updated. If the value is 0000_H, it means that the number of transfers until the address reload when the reload function 2 is used and the number of transfers when the block transfer 2 is used are 65536.
15 to 0	TRC[15:0]	Transfer count Configures the number of transfers. PDMAnDTCm.TRC[15:0] is decremented by one whenever a DMA cycle is executed. If you read from those bits, the remaining number of transfers for the next DMA cycle is read. If the reload function is not used, after the last transfer is complete, the value at the completion (0000_H) is retained.

TRC15-0	Operation
0000_H	The number of transfers is 65536, or the transfer is complete.
0001_H	The number of transfers is 1, or remaining transfer count is 1.
:	:
$FFFF_H$	The number of transfers is 65535, or remaining transfer count is 65535.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (PDMAnDCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the transfer count and the address reload count are updated.

8.9.2.4 PDMAnDTCTm — DMAC Transfer Control Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + $040C_{H}$ + 40_{H} × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ESE	DRS	—	—	—	—	—	CHNSEL2	CHNSEL1	CHNSEL0	CHNE1	CHNE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TCE	MLE	RLD2M1	RLD2M0	RLD1M1	RLD1M0	DACM1	DACM0	SACM1	SACM0	DS2	DS1	DS0	TRM1	TRM0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.38 PDMAnDTCTm Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	ESE	DMA transfer disable on transfer error setting Specifies whether to execute a DMA cycle when the PDMAnDCSTm.ER bit is set due to DMA transfer error. If this bit is cleared to 0, even when the PDMAnDCSTm.ER bit is set due to DMA transfer error, the following DMA cycles can be executed. If this bit is set to 1, the following DMA cycles are not executed when the PDMAnDCSTm.ER bit is set due to DMA transfer error. 0: DMA cycles are executed while the PDMAnDCSTm.ER bit is set. 1: DMA cycles are not executed while the PDMAnDCSTm.ER bit is set.
26	DRS	DMA transfer request selection assignment Selects the type of DMA transfer requests to be accepted. 0: Software DMA transfer request 1: Hardware DMA transfer request
25 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 18	CHNSEL[2:0]	Selection of next channel in the chain Specifies the next channel in the chain. The next channel must be another channel in the same DMAC. You cannot specify a channel in the different DMAC. It is prohibited to specify the channel as the next channel in the chain. (If you do, the correct operation is not guaranteed.)
17, 16	CHNE[1:0]	Chain enable Selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DMA cycle when the remaining transfer count is one. 10: Setting prohibited. (The operation is not guaranteed.) 11: Always chain A chain request is generated at the completion of every DMA cycle.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	TCE	Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer.

Table 8.38 PDMAnDTCTm Register Contents

Bit Position	Bit Name	Function
13	MLE	Continuous transfer enable If this bit is set, the PDMAnDCENm.DTE bit is not cleared at the completion of DMA transfer. Even if the PDMAnDCSTm.TC bit is not cleared, DMA transfer starts when there is a DMA transfer request. 0: The PDMAnDCENm.DTE bit is cleared at the completion of DMA transfer. The next DMA transfer starts only after the PDMAnDCSTm.TC bit is cleared. 1: The PDMAnDCENm.DTE bit is not cleared at the completion of DMA transfer. Even if the PDMAnDCSTm.TC bit is not cleared, DMA transfer starts when there is a DMA transfer request.
12, 11	RLD2M[1:0]	Reload function 2 setting Specifies the reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.
10, 9	RLD1M[1:0]	Reload function 1 setting Specifies the reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)
8, 7	DACM[1:0]	Destination address count direction Specifies the count direction of the destination address. DACM1 / DACM0 / Direction of Count 0 / 0 / Increment 0 / 1 / Decrement 1 / 0 / Fixed 1 / 1 / Setting prohibited (The operation is not guaranteed.)
6, 5	SACM[1:0]	Source address count direction Specifies the count direction of the source address. SACM1 / SACM0 / Direction of Count 0 / 0 / Increment 0 / 1 / Decrement 1 / 0 / Fixed 1 / 1 / Setting prohibited (The operation is not guaranteed.)

Table 8.38 PDMAnDTCTm Register Contents

Bit Position	Bit Name	Function
4 to 2	DS[2:0]	Transfer data size Specifies the transfer data size.
1, 0	TRM[1:0]	Transfer mode Specifies the DMA transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Setting prohibited (The operation is not guaranteed.)

DS2	DS1	DS0	Transfer Data Size
0	0	0	8 bits
0	0	1	16 bits
0	1	0	32 bits
0	1	1	64 bits
1	0	0	128 bits
Other than the above			Setting prohibited (The operation is not guaranteed.)

CAUTIONS

1. Except for the case to clear PDMAnDTCTm.MLE bit, it is forbidden to write to those bits when the channel operation is enabled (PDMAnDCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If forbidden settings are used for each bits, the correct operation is not guaranteed.

8.9.2.5 PDMAnDRSAm — DMAC Reload Source Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0410_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.39 PDMAnDRSAm Register Contents

Bit Position	Bit Name	Function
31 to 0	RSA[31:0]	Reload source address Specifies the source address to be reloaded to the DMA source address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary bit.) The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	RSA3	RSA2	RSA1	RSA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.6 PDMAnDRDAm — DMAC Reload Destination Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0414$_H$ + 40$_H$ × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.40 PDMAnDRDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	RDA[31:0]	Reload destination address Specifies the destination address to be reloaded to the DMA destination address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary bit.) The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	RDA3	RDA2	RDA1	RDA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.7 PDMAnDRTCm — DMAC Reload Transfer Count Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0418_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RARC 15	RARC 14	RARC 13	RARC 12	RARC 11	RARC 10	RARC9	RARC8	RARC7	RARC6	RARC5	RARC4	RARC3	RARC2	RARC1	RARC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTRC 15	RTRC 14	RTRC 13	RTRC 12	RTRC 11	RTRC 10	RTRC9	RTRC8	RTRC7	RTRC6	RTRC5	RTRC4	RTRC3	RTRC2	RTRC1	RTRC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.41 PDMAnDRTCm Register Contents

Bit Position	Bit Name	Function
31 to 16	RARC[15:0]	Reload address reload count Specifies the value to be reloaded to the address reload count in the transfer count register at the timing of reload when the reload function 2 is used.
15 to 0	RTRC[15:0]	Reload transfer count Specifies the value to be reloaded to the transfer count in the transfer count register at the timing of reload when the reload function 1 is used.

8.9.2.8 PDMAnDCENm — DMAC Channel Operation Enable Setting Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0420_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.42 PDMAnDCENm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DTE	Channel operation enable Specifies whether to enable or disable the transfer operation of the channel. If the PDMAnDCENm.DTE bit is 1, DMA transfer starts when there is a DMA transfer request. If the PDMAnDTCTm.MLE bit is 0, this bit is automatically cleared when DMA transfer is completed. In addition, if 0 is written to the PDMAnDCENm.DTE bit during DMA transfer, the DMA transfer is suspended. If 1 is written to the PDMAnDCENm.DTE bit during suspension, the suspension is cleared and the DMA transfer resumes. 0: Channel operation is disabled/Channel suspended 1: Channel operation is enabled/Channel suspension cleared

8.9.2.9 PDMAnDCSTm — DMAC Transfer Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0424_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ERWR	—	—	CY	ER	—	—	TC	—	—	DR	SR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.43 PDMAnDCSTm Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	ERWR	DMA Transfer Error occurring cycle This bit is updated at the same time as setting of the DMA transfer error flag (PDMAnDCSTm.ER), indicating in which cycle of read or write the DMA transfer error occurred. This bit is not updated when a new DMA transfer error occurs after the PDMAnDCSTm.ER bit has been set. If the PDMAnDCSTm.ER bit is cleared, this bit is also cleared to 0. 0: DMA transfer error occurs in the read cycle. 1: DMA transfer error occurs in the write cycle.
10, 9	Reserved	When read, the value after reset is returned.
8	CY	DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing.
7	ER	Transfer error flag This bit is set when DMA transfer error is generated. If this bit is 1 and the PDMAnDTCTm.ESE bit is set, a DMA cycle is not executed even when a DMA transfer request is generated. 0: No DMA transfer error is generated 1: DMA transfer error is generated
6, 5	Reserved	When read, the value after reset is returned.
4	TC	Transfer completion flag This bit is set at the completion of the last transfer and indicates that the DMA transfer is complete. If the PDMAnDTCTm.MLE bit is 0 and this bit is 1, a DMA cycle is not executed when a DMA transfer request is generated. 0: DMA transfer is incomplete 1: DMA transfer is complete
3, 2	Reserved	When read, the value after reset is returned.

Table 8.43 PDMAnDCSTm Register Contents

Bit Position	Bit Name	Function
1	DR	Hardware DMA transfer request status This bit shows whether there is a hardware DMA transfer request (PDMAnDMARQ) from the DTFR. This bit changes regardless of the value of the PDMAnDCENm.DTE bit when a hardware DMA transfer request from the DTFR is generated. If the software DMA transfer request has been selected in the transfer request selection bit (PDMAnDTCTm.DRS) in the DMAC transfer control register, this bit is not set even when a hardware DMA transfer request is input from the DTFR. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request
0	SR	Software DMA transfer request flag This bit shows whether there is a software DMA transfer request. This bit is automatically cleared when executing the DMA transfer. A user can set this bit by writing 1 to the PDMAnDCSTSm.SRS bit in the DMAC transfer status set register (PDMAnDCSTSm). In addition, a user can clear this bit by writing 1 to the PDMAnDCSTCm.SRC bit in the DMAC transfer status clear register (PDMAnDCSTCm), but if this is done, the ongoing DMA transfer is aborted and cannot be resumed. 0: There is no software DMA transfer request 1: There is a software DMA transfer request

8.9.2.10 PDMAnDCSTSm — DMAC Transfer Status Set Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0428_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.44 PDMAnDCSTSm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRS	Software DMA transfer request flag A user can set the software DMA transfer request flag (PDMAnDCSTm.SR) by writing 1 to this bit. When read, this bit is always read as 0.

8.9.2.11 PDMAnDCSTCm — DMAC Transfer Status Clear Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + $042C_H$ + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ERC	—	—	TCC	—	—	—	SRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R/W

Table 8.45 PDMAnDCSTCm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ERC	Transfer error flag clear The DMA transfer error flag (PDMAnDCSTm.ER) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TCC	Transfer completion flag clear The transfer completion flag (PDMAnDCSTm.TC) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRC	Software DMA transfer request flag clear The software DMA transfer request flag (PDMAnDCSTm.SR) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.

8.9.2.12 PDMAnDTFRm — DTFR Setting Register

Access: This register can be read or written in 32-bit units.

Address: <DMAn_base> + 0430_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	REQSEL6	REQSEL5	REQSEL4	REQSEL3	REQSEL2	REQSEL1	REQSEL0	REQEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.46 PDMAnDTFRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 1	REQSEL[6:0]	Hardware DMA transfer source selection Selects one out of 128 hardware DMA transfer sources as the hardware DMA transfer request. 000_0000: Select the DMACTRG[0] input : 111_1111: Select the DMACTRG[127] input
0	REQEN	Hardware DMA transfer source selection enable This bit enables/disables the hardware DMA transfer source selection. 0: Hardware DMA transfer source selection is disabled. 1: Hardware DMA transfer source selection is enabled. If this bit is 0, even when the hardware DMA transfer source selected by the PDMAnDTFRm.REQSEL[6:0] bits is activated, it is not recognized as a hardware DMA transfer request, and a hardware DMA transfer request is not generated.

8.9.2.13 PDMAnDTFRRQm — DTFR Transfer Request Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMAn_base> + 0434_H + 40_H × Ch. No. m

(m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.47 PDMAnDTFRRQm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DRQ	Hardware DMA transfer request status This bit indicates that a hardware DMA transfer request exists or is retained. ● This bit indicates whether a hardware DMA transfer request is retained or not. When the DMA transfer request acceptance signal from the DMAC is asserted, this bit is automatically cleared. A user can clear this bit by writing 1 to the PDMAnDTFRRQCm.DRQC bit. This bit changes regardless of the value of the PDMAnDTFRm.REQEN bit when a hardware DMA transfer request from the outside is generated. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request

8.9.2.14 PDMAnDTFRRQCm — DTFR Transfer Request Clear Register

Access:	This register can be read or written in 32-bit units.
Address:	<DMAn_base> + 0438_H + 40_H × Ch. No. m (m = 0 to 31 in RH850/F1KH-D8, m = 0 to 31 in RH850/F1KM-S4, m = 0 to 15 in RH850/F1KM-S1)
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.48 PDMAnDTFRRQCm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DRQC	Hardware DMA transfer request clear A user can clear the PDMAnDTFRRQm.DRQ bit by writing 1 to this bit. When read, this bit is always read as 0.

8.10 DMA Trigger Factor Select Registers

The following registers are used to select a DMA Trigger Factor.

8.10.1 List of DMA Trigger Factor Select Register Address

The DMA Trigger Factor Select Register addresses are listed in the table below.

Table 8.49 List of DMA Trigger Factor Select Register (RH850/F1KH-D8)

Module Name	Register Name	Symbol	Address
SL_DMAC	DMA trigger factor select register of TAUD0	DTFSEL_TAUD0	FFC0 2000_H
	DMA trigger factor select register of TAUB0	DTFSEL_TAUB0	FFC0 2004_H
	DMA trigger factor select register of TAUB1	DTFSEL_TAUB1	FFC0 2008_H

Table 8.50 List of DMA Trigger Factor Select Register (RH850/F1KM-S4)

Module Name	Register Name	Symbol	Address
SL_DMAC	DMA trigger factor select register of TAUD0	DTFSEL_TAUD0	FFC0 2000_H
	DMA trigger factor select register of TAUB0	DTFSEL_TAUB0	FFC0 2004_H
	DMA trigger factor select register of TAUB1	DTFSEL_TAUB1	FFC0 2008_H

Note: DTFSEL_TAUB1 is not supported on 100-pin and 144-pin products.

Table 8.51 List of DMA Trigger Factor Select Register (RH850/F1KM-S1)

Module Name	Register Name	Symbol	Address
SL_DMAC	DMA trigger factor select register of TAUD0	DTFSEL_TAUD0	FFC0 2000_H
	DMA trigger factor select register of TAUB0	DTFSEL_TAUB0	FFC0 2004_H

8.10.2 DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1 — DMA Trigger Factor Select Register

When 16 or 17 DMA Trigger Factors are assigned to one DMA Trigger Number, this register selects which DMA Trigger Factor is enabled.

Access:	This register can be read or written in 32-bit units.
Address:	DTFSEL_TAUD0: FFC0 2000$_H$
	DTFSEL_TAUB0: FFC0 2004$_H$
	DTFSEL_TAUB1: FFC0 2008$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	REQSEL6*2	REQSEL5[3:0]				REQSEL4[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REQSEL3[3:0]				REQSEL2[3:0]				REQSEL1[3:0]				REQSEL0[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.52 DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1 Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	REQSEL6*2	RH850/F1KH-D8: DMA Trigger Factor selection 0: TAUB1REQSEL5 1: RSCANFDRF15 RH850/F1KM-S4, RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset
23 to 20	REQSEL5[3:0]	DMA Trigger Factor selection of TAUD0REQSEL5/TAUB0REQSEL5/TAUB1REQSEL5.*1
19 to 16	REQSEL4[3:0]	DMA Trigger Factor selection of TAUD0REQSEL4/TAUB0REQSEL4/TAUB1REQSEL4.*1
15 to 12	REQSEL3[3:0]	DMA Trigger Factor selection of TAUD0REQSEL3/TAUB0REQSEL3/TAUB1REQSEL3.*1
11 to 8	REQSEL2[3:0]	DMA Trigger Factor selection of TAUD0REQSEL2/TAUB0REQSEL2/TAUB1REQSEL2.*1
7 to 4	REQSEL1[3:0]	DMA Trigger Factor selection of TAUD0REQSEL1/TAUB0REQSEL1/TAUB1REQSEL1.*1
3 to 0	REQSEL0[3:0]	DMA Trigger Factor selection of TAUD0REQSEL0/TAUB0REQSEL0/TAUB1REQSEL0.*1

Note 1. For Detail, see **Table 8.53, DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj/TAUB1REQSELj (RH850/F1KH-D8)**, **Table 8.54, DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj/TAUB1REQSELj (RH850/F1KM-S4)** or **Table 8.55, DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj (RH850/F1KM-S1)**.

Note 2. This bit is only supported for DTFSEL_TAUB1. For DTFSEL_TAUD0 and DTFSEL_TAUB0, when writing, write the value after reset.

NOTE

For details on DMA Trigger Factors, see **Section 8.1.4, DMA Trigger Factors**.

Table 8.53 DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj/TAUB1REQSELj (RH850/F1KH-D8)

REQSELj[3:0]	TAUD0REQSELj	TAUB0REQSELj	TAUB1REQSELj
0000_B	INTTAUD0I0	INTTAUB0I0	INTTAUB1I0
0001_B	INTTAUD0I1	INTTAUB0I1	INTTAUB1I1
0010_B	INTTAUD0I2	INTTAUB0I2	INTTAUB1I2
	:		
1110_B	INTTAUD0I14	INTTAUB0I14	INTTAUB1I14
1111_B	INTTAUD0I15	INTTAUB0I15	INTTAUB1I15

Note: j = 5 to 0

Table 8.54 DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj/TAUB1REQSELj (RH850/F1KM-S4)

REQSELj[3:0]	TAUD0REQSELj	TAUB0REQSELj	TAUB1REQSELj
0000_B	INTTAUD0I0	INTTAUB0I0	INTTAUB1I0
0001_B	INTTAUD0I1	INTTAUB0I1	INTTAUB1I1
0010_B	INTTAUD0I2	INTTAUB0I2	INTTAUB1I2
	:		
1110_B	INTTAUD0I14	INTTAUB0I14	INTTAUB1I14
1111_B	INTTAUD0I15	INTTAUB0I15	INTTAUB1I15

Note: j = 5 to 0

Table 8.55 DMA Trigger Factor Selection of TAUD0REQSELj/TAUB0REQSELj (RH850/F1KM-S1)

REQSELj[3:0]	TAUD0REQSELj	TAUB0REQSELj
0000_B	INTTAUD0I0	INTTAUB0I0
0001_B	INTTAUD0I1	INTTAUB0I1
0010_B	INTTAUD0I2	INTTAUB0I2
	:	
1110_B	INTTAUD0I14	INTTAUB0I14
1111_B	INTTAUD0I15	INTTAUB0I15

Note: j = 5 to 0

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1.

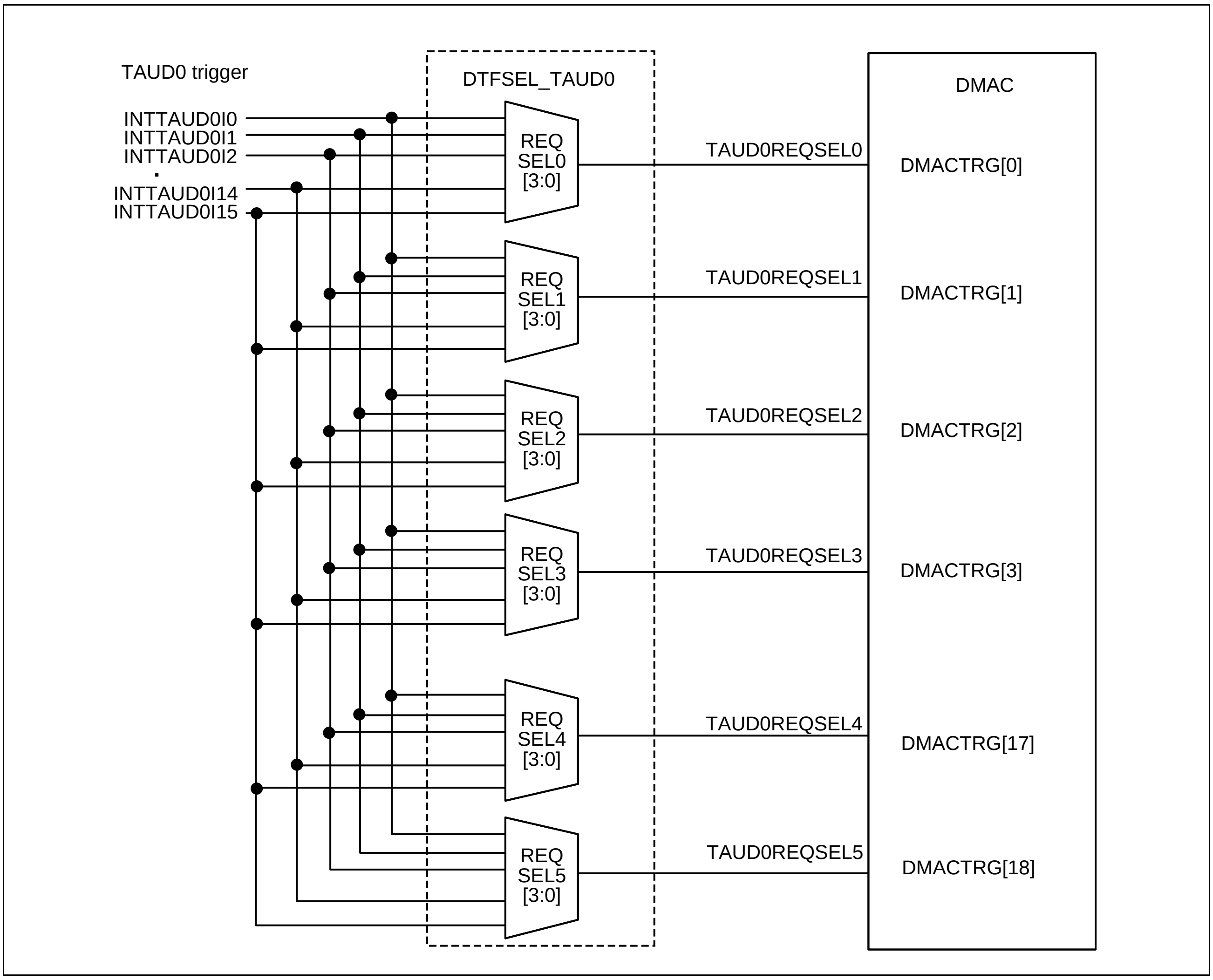


Figure 8.10 Configuration Diagram of DTFSEL_TAUD0

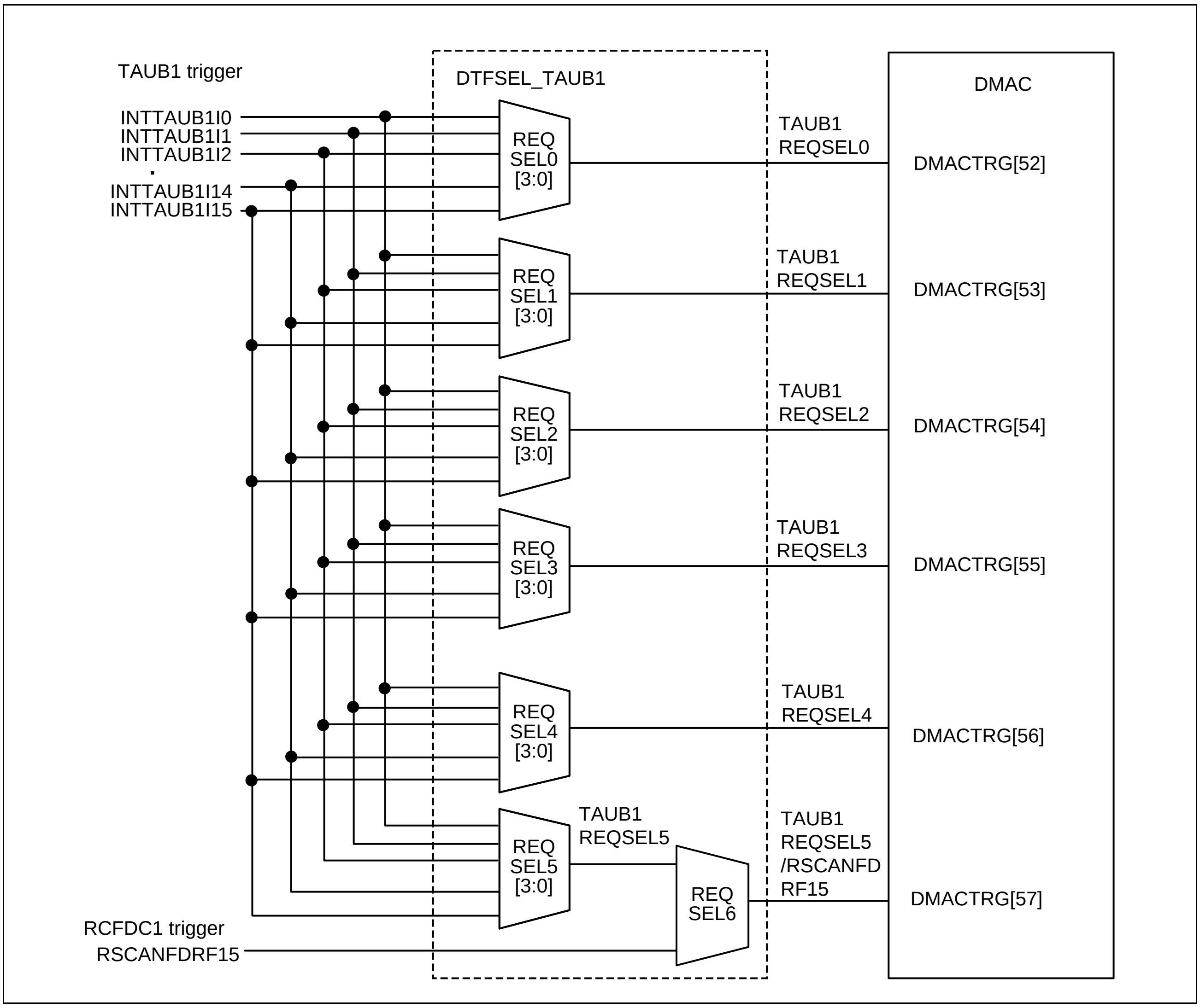


Figure 8.11 Configuration Diagram of DTFSEL_TAUB1 (RH850/F1KH-D8)

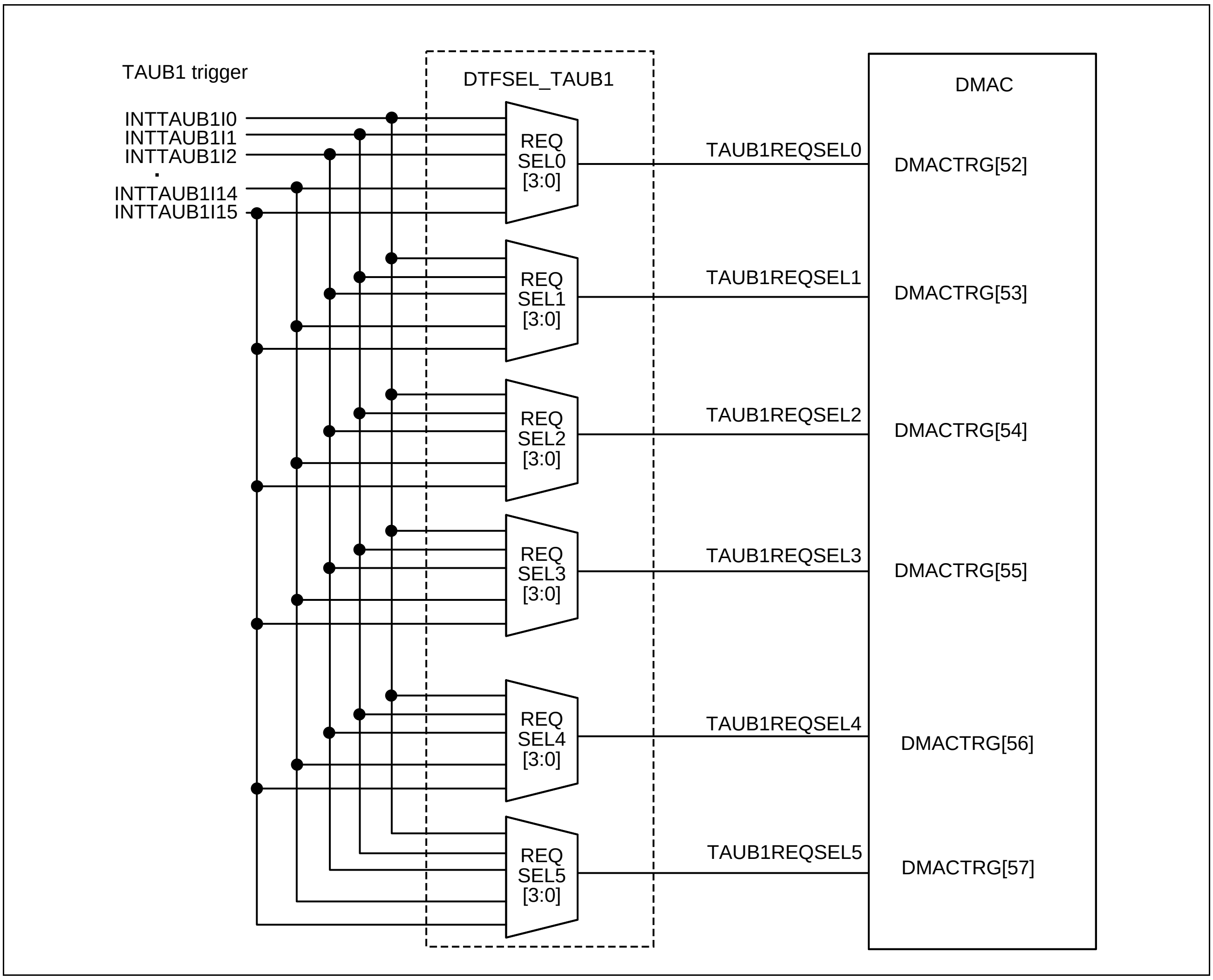


Figure 8.12 Configuration Diagram of DTFSEL_TAUB1 (RH850/F1KM-S4)

Section 9A Reset Controller of RH850/F1KH-D8

9A.1 Overview

Several system reset functions are provided in order to initialize CPU core and peripheral functions as well as their associated registers.

A reset can be caused by the following events:

Table 9A.1 Reset Sources and Reset Targets

Reset Source	Symbol	RH850/F1KH-D8
External reset	$\overline{\text{RESET}}$	✓
Power-On Clear	POCRES	✓
Watchdog timer reset	WDTA0RES	✓
	WDTA1RES	✓
	WDTA2RES	✓
Clock monitor reset	$\overline{\text{CLMA0RES}}$	✓
	$\overline{\text{CLMA1RES}}$	✓
	$\overline{\text{CLMA2RES}}$	✓
	$\overline{\text{CLMA3RES}}$	✓
Low-voltage indicator reset	$\overline{\text{LVIRES}}$	✓
Software reset	SWRES	✓
Debugger reset	$\overline{\text{DBRES}}$	✓
Core voltage monitor reset	$\overline{\text{CVMRES}}$	✓
Transition to DeepSTOP mode		✓

9A.1.1 Reset Sources

Reset levels and reset sources are shown below.

Various reset sources are assigned to the different levels of the reset.

Table 9A.2 Reset Sources and Reset Targets

Reset Level	Reset Source	Clock Generation (except PLL)/Real-Time Clock/CVM/LVI	Always-On area (AWO area) Modules*[1]	Isolated area (ISO area) Modules*[2]
1	Power-On Clear (POCRES) Debugger reset ($\overline{\text{DBRES}}$)	Reset	Reset	Reset
2	External reset ($\overline{\text{RESET}}$) Watchdog timer reset (WDTA0RES, WDTA1RES, WDTA2RES) Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, $\overline{\text{CLMA3RES}}$) Core voltage monitor reset ($\overline{\text{CVMRES}}$) Low voltage indicator reset ($\overline{\text{LVIRES}}$) Software reset (SWRES)	Not reset target*[3]	Reset	Reset
3	Reset by DeepSTOP mode	Not reset target	Not reset target	Reset

Note 1. Clock generator, real-time clock, CVM, and LVI are excluded.

Note 2. PLL is included.

Note 3. In clock monitor reset, oscillator-related registers for clock monitoring are initialized.

Reset level 1: Initializes the entire microcontroller.

Reset level 2: For a quick return to normal operating mode by eliminating the oscillator stabilization time, initializes the entire microcontroller except for the clock generator and the real-time clock.

Reset level 3: At the transition to DeepSTOP mode, initializes all the Isolated area (ISO area).

If each reset is generated, local RAM data and global RAM data are not guaranteed to retaine.

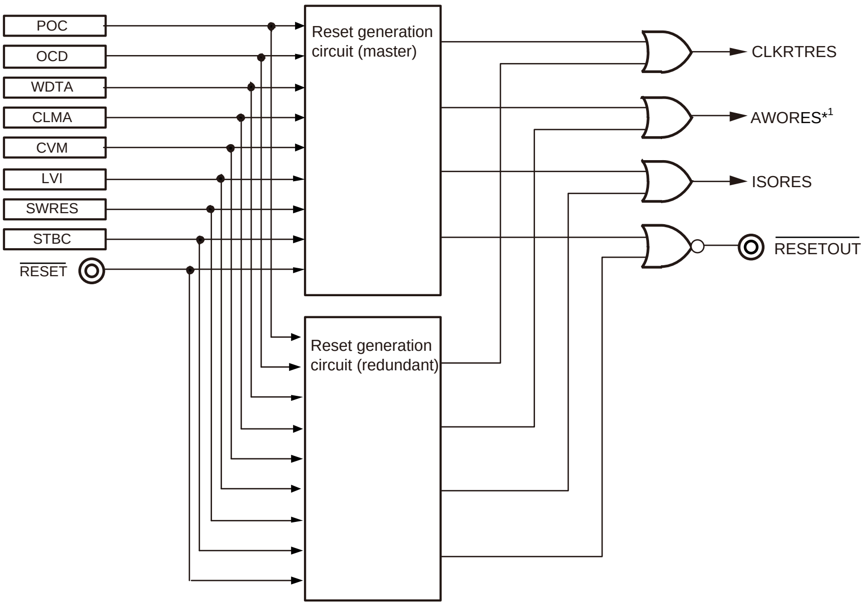
If each reset is generated, retention RAM data can be retained as long as the power-supply voltage (REG0VCC) does not fall below the RAM retention voltage (VVLVI).

In this case, data value writing to retention RAM address will be before or after writing.

9A.1.2 Reset Controller Redundancy

The reset controller of the microcontroller has a redundant configuration, and includes duplicated reset generation circuits. Such configuration enables initialization of the reset targeted area without failure even if one of the two reset generation circuits fails.

The configuration of the reset generation circuits is shown in the figure below.



Note 1. Real-time clock is excluded.

Figure 9A.1 Reset Controller Redundancy

At the generation of a reset, the same reset source signal is input to two reset generation circuits.

According to the reset source, the two reset generation circuits output the Always-On area (AWO area) reset signal (AWORES), Isolated area (ISO area) reset signal (ISORES), clock generator/real-time clock reset signal (CLKRTRES), and $\overline{\text{RESETOUT}}$ signal.

The AWORES, ISORES, CLKRTRES, and $\overline{\text{RESETOUT}}$ signals are generated by executing the logical OR of the signals output from two reset generation circuits. Thus, a reset signal is generated normally even if one of the two reset generation circuits fails.

Whether a reset generation circuit operates normally can be checked by reading and comparing the reset factor registers of the respective reset generation circuits.

9A.1.3 Reset Output ($\overline{\text{RESETOUT}}$)

When a reset source of reset level 1 or 2 is generated, a reset output signal ($\overline{\text{RESETOUT}}$) is output to the outside. Reset output is used to reset external devices at the same time as a reset is generated inside the microcontroller.

For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** .

9A.1.4 Reset Flag

To identify a reset source, two registers with a flag for each reset source are provided. The main elements of the reset controller are shown in **Figure 9A.2, Block Diagram of the Reset Controller**.

9A.1.5 Clock Supply

The clock supply to the reset controller is shown in the following table.

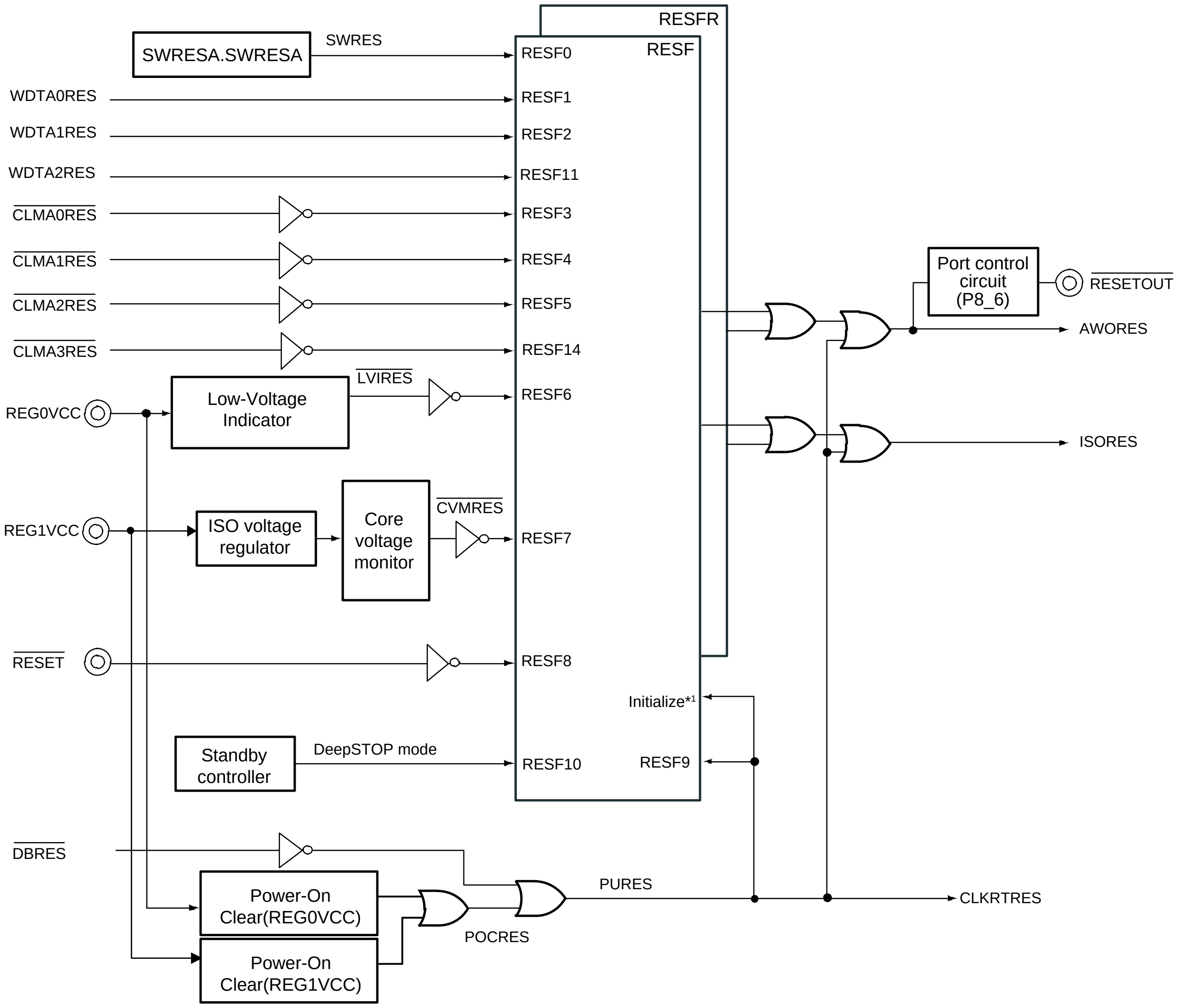
Table 9A.3 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Reset	Register access clock	CPUCLK_UL

9A.2 Configuration

9A.2.1 Block Diagram

Block diagram of reset circuits are shown below.



Note 1. Registers RESF and RESFR are initialized by PURES.
After an initialization, bits RESF9 and RESFR9 are set to 1.

Figure 9A.2 Block Diagram of the Reset Controller

(1) Reset Signals

The reset controller manages the generation of three reset signals upon occurrence of reset signals from various reset sources:

- Always-On area (AWO area) reset (AWORES)
  AWORES is generated by all reset sources except the transition to DeepSTOP mode.
  AWORES resets all modules in the Always-On area (AWO area) except clock generation circuit, real-time clock, core voltage monitor, and low-voltage detection circuit.
- Isolated area (ISO area) reset (ISORES)
  ISORES is generated by all reset sources.
  ISORES resets all modules (including PLL) in the Isolated area (ISO area).
- CLKRTRES
  CLKRTRES is generated by the power-on clear or debugger reset sources.
  CLKRTRES resets the clock generation circuit (excluding PLL) and real-time clock.

The power-up reset (PURES) is caused by the power-on clear and debugger reset sources.

Following the generation of an AWORES reset, with the exception of the PLL, all clock-generation circuits that were operating at the time (LS IntOSC, HS IntOSC, MainOSC, SubOSC) continue to operate. On the generation of a $\overline{\text{CLMA0RES}}$ reset, the HS IntOSC that was the target for CLMA0 monitoring is reset. On the generation of a $\overline{\text{CLMA1RES}}$ reset, the MainOSC that was the target for CLMA1 monitoring is reset.

The PURES initializes all of the clock generation circuits. It is necessary to restart the clock generation circuit after recovery from the PURES.

The CPU reset is the Isolated area (ISO area) reset (ISORES) to the CPU sub system.

(2) Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) hold a flag for each reset source, and this flag is set when the corresponding reset is asserted.

All reset flags except RESF9 and RESFR9 are initialized by a power-up reset (PURES). (Bits RESF9 and RESFR9 are set to 1 after initialization.) In addition, all the bits can be cleared by software.

For details, see **Section 9A.1.4, Reset Flag**.

(3) On-Chip Module Resets

(a) Watchdog Timer Resets

The watchdog timers can generate three types of resets: WDTA0RES, WDTA1RES and WDTA2RES.

For details, see **Section 9A.4.6, Watchdog Timer (WDTA) Reset**

(b) Clock Monitor Resets

The clock monitors can generate four resets: $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$.

For details, see **Section 9A.4.8, Clock Monitor (CLMA) Reset**.

(c) Debugger Reset

A reset is generated by a command from a debugger. This leads to a generation of power-up reset

PURES. For details, see **Section 9A.4.9, Debugger Reset**.

(4) Software Controlled Reset (SWRES)

A software reset SWRES can be generated by use of the software reset register SWRESA.

For details, see **Section 9A.4.7, Software Reset**.

(5) Reset Output Signal

During reset and after release from the reset, port P8_6 outputs low level as $\overline{\text{RESETOUT}}$ function.

For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

(6) Power Supply Monitoring

The following power supply detection circuits observe the level of the external power supply REG0VCC and REG1VCC.

(a) Low-Voltage Indicator

The low-voltage indicator (LVI) generates the $\overline{\text{LVIRES}}$ reset, if the voltage level of REG0VCC drops below a certain level. The level can be adjusted and the $\overline{\text{LVIRES}}$ can be masked.

For details, see **Section 9A.4.3, Low-Voltage Indicator (LVI) Reset**.

(b) Power-On Clear

The power-on clear circuit (POC) continuously compares the power supply voltage REG0VCC and REG1VCC with an internal reference voltage. Thus, a reset is generated when the power supply voltage goes below a certain level.

For details, see **Section 9A.4.2, Power-On Clear (POC) Reset**.

(c) Core Voltage Monitor

A reset can be generated when the core voltage monitor (CVM) detects over- or undervoltage in core voltage. (Output/not output can be set by option byte.)

For details, see **Section 9A.4.4, Core Voltage Monitor (CVM) Reset**.

(7) Masking of Reset Sources in Debugging Mode

The following reset sources can be masked during debugging:

Table 9A.4 Reset Sources to be Masked during Debugging

Reset Source	Maskable/Non-maskable
Power-on clear (POCRES)	—
Debugger reset ($\overline{\text{DBRES}}$)	—
External reset ($\overline{\text{RESET}}$)	✓
Low-voltage indicator reset ($\overline{\text{LVIRES}}$)	✓
Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, $\overline{\text{CLMA3RES}}$)	✓
Watchdog timer reset (WDTA0RES, WDTA1RES, WDTA2RES)	✓
Core voltage monitor reset ($\overline{\text{CVMRES}}$)	✓
Software reset (SWRES)	✓
Reset by DeepSTOP mode	—

9A.3 Registers

This section contains a description of all registers of the reset controller.

9A.3.1 Reset Controller Registers Overview

The reset controller is controlled and operated by the following registers:

Table 9A.5 Reset Controller Registers Overview

Module Name	Register Name	Symbol	Address
RESCTL	Reset flag registers		
	Reset factor register	RESF	FFF8 0760$_H$
	Reset factor clear register	RESFC	FFF8 0768$_H$
	Redundant reset factor register	RESFR	FFF8 0860$_H$
	Redundant reset factor clear register	RESFCR	FFF8 0868$_H$
	Software reset control register		
	Software reset register	SWRESA	FFF8 0A04$_H$

NOTES

1. For the LVI related, RAM store related, and CVM related registers, see **Section 11A, Supply Voltage Monitor of RH850/F1KH-D8**.
2. As for the protection registers, see **Section 5, Write-Protected Registers**.

9A.3.2 Details of Reset Flag Registers

9A.3.2.1 RESF — Reset Factor Register

This register contains information about which type of resets occurred after the last power-on clear reset. This register is initialized by a power-up reset PURES.

Each reset condition sets the corresponding flag in the register.

For example, if a clock monitor reset $\overline{\text{CLMA0RES}}$ occurs after a watchdog timer reset WDTA0RES, RESF reads 0000 000A$_H$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0760$_H$

Value after reset: 0000 0200$_H$ / 0000 0300$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESF 14	—	—	RESF 11	RESF 10	RESF9	RESF8	RESF7	RESF6	RESF5	RESF4	RESF3	RESF2	RESF1	RESF0
Value after reset	0	0	0	0	0	0	1	1/0^{*1}	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9A.4, When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed**, When $\overline{\text{RESET}}$ is released before Execution of Flash Sequence.

Table 9A.6 RESF Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	RESF14	CLMA3 reset flag 0: No reset occurred 1: Reset has occurred
13, 12	Reserved	When read, the value after reset is returned.
11	RESF11	WDTA2 reset flag 0: No reset occurred 1: Reset has occurred
10	RESF10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESF9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESF8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESF7	CVM reset flag 0: No reset occurred 1: Reset has occurred

Table 9A.6 RESF Register Contents

Bit Position	Bit Name	Function
6	RESF6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESF5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESF4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESF3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESF2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESF1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESF0	Software reset flag 0: No reset occurred 1: Reset has occurred

9A.3.2.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

Access:	This register is a write-only register that can be written in 32-bit units.
Address:	FFF8 0768$_H$
Value after reset:	Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFC 14	—	—	RESFC 11	RESFC 10	RESFC 9	RESFC 8	RESFC 7	RESFC 6	RESFC 5	RESFC 4	RESFC 3	RESFC 2	RESFC 1	RESFC 0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	W	R	R	W	W	W	W	W	W	W	W	W	W	W	W

Table 9A.7 RESFC Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write “0”.
14	RESFC14	CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag
13, 12	Reserved	When writing, write “0”.
11	RESFC11	WDTA2 reset flag clear 0: Do not clear flag 1: Clear flag
10	RESFC10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFC9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFC8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFC7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFC6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFC5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag
4	RESFC4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag

Table 9A.7 RESFC Register Contents

Bit Position	Bit Name	Function
3	RESFC3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFC2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFC1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFC0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9A.3.2.3 RESFR — Redundant Reset Factor Register

This register is a duplication of the reset factor register. This register is initialized by a power-up reset PURES.

In accordance with the setting conditions for each bit in the reset factor register, the same bits are set in this register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0860$_H$

Value after reset: 0000 0200$_H$ / 0000 0300$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFR 14	—	—	RESFR 11	RESFR 10	RESFR 9	RESFR 8	RESFR 7	RESFR 6	RESFR 5	RESFR 4	RESFR 3	RESFR 2	RESFR 1	RESFR 0
Value after reset	0	0	0	0	0	0	1	1/0*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9A.4, When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed**, When $\overline{\text{RESET}}$ is released before Execution of Flash Sequence.

Table 9A.8 RESFR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	RESFR14	CLMA3 reset flag 0: No reset occurred 1: Reset has occurred
13, 12	Reserved	When read, the value after reset is returned.
11	RESFR11	WDTA2 reset flag 0: No reset occurred 1: Reset has occurred
10	RESFR10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESFR9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESFR8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESFR7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESFR6	LVI reset flag 0: No reset occurred 1: Reset has occurred

Table 9A.8 RESFR Register Contents

Bit Position	Bit Name	Function
5	RESFR5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESFR4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESFR3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESFR2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESFR1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESFR0	Software reset flag 0: No reset occurred 1: Reset has occurred

9A.3.2.4 RESFCR — Redundant Reset Factor Clear Register

This register clears the reset flags of the RESFR.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0868$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFCR14	—	—	RESFCR11	RESFCR10	RESFCR9	RESFCR8	RESFCR7	RESFCR6	RESFCR5	RESFCR4	RESFCR3	RESFCR2	RESFCR1	RESFCR0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	W	R	R	W	W	W	W	W	W	W	W	W	W	W	W

Table 9A.9 RESFCR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write "0".
14	RESFCR14	CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag
13, 12	Reserved	When writing, write "0".
11	RESFCR11	WDTA2 reset flag clear 0: Do not clear flag 1: Clear flag
10	RESFCR10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFCR9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFCR8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFCR7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFCR6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFCR5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag

Table 9A.9 RESFCR Register Contents

Bit Position	Bit Name	Function
4	RESFCR4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFCR3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFCR2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFCR1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFCR0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9A.3.3 Details of Software Reset Control Register

9A.3.3.1 SWRESA — Software Reset Register

This register is used to generate a software reset SWRES. The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0A04$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRESA
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9A.10 SWRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write “0”.
0	SWRESA	Software reset trigger 0: No Software reset trigger is generated. 1: Software reset trigger is generated.

9A.4 Functional Description

9A.4.1 Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) provide reset flags for each reset source.

If a reset has occurred, the corresponding flag is set. According to this, the source of the reset is evaluated.

RESF and RESFR are initialized by a power-up reset PURES (POCRES or $\overline{\text{DBRES}}$) (though bits RESF9 and RESFR9 are set to 1 after initialization). In addition, flags in RESF and RESFR can be cleared by the reset factor clear register (RESFC) and the redundant reset factor clear register (RESFCR).

Each reset source can set the corresponding flag independently from other reset sources.

9A.4.2 Power-On Clear (POC) Reset

The power-on clear circuit (POC) constantly compares the power supply voltage REG0VCC and REG1VCC with the internal reference voltage VPOC. It ensures that the microcontroller only operates as long as the power supply exceeds a certain level.

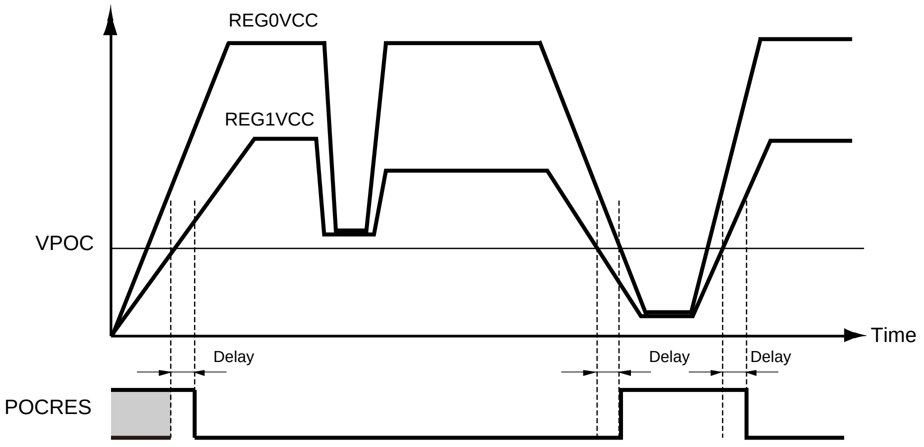
If REG0VCC or REG1VCC falls below the internal reference voltage (REG0VCC < VPOC or REG1VCC < VPOC), the internal reset signal POCRES and a power-up reset PURES are generated.

For details on the specification of the internal voltage reference level VPOC, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**.

The reset factor register (RESF) and the redundant reset factor register (RESFR) are cleared by the power-on clear reset. RESF9 and RESFR9 are set to 1 after initialization.

The power-on clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level VPOC.

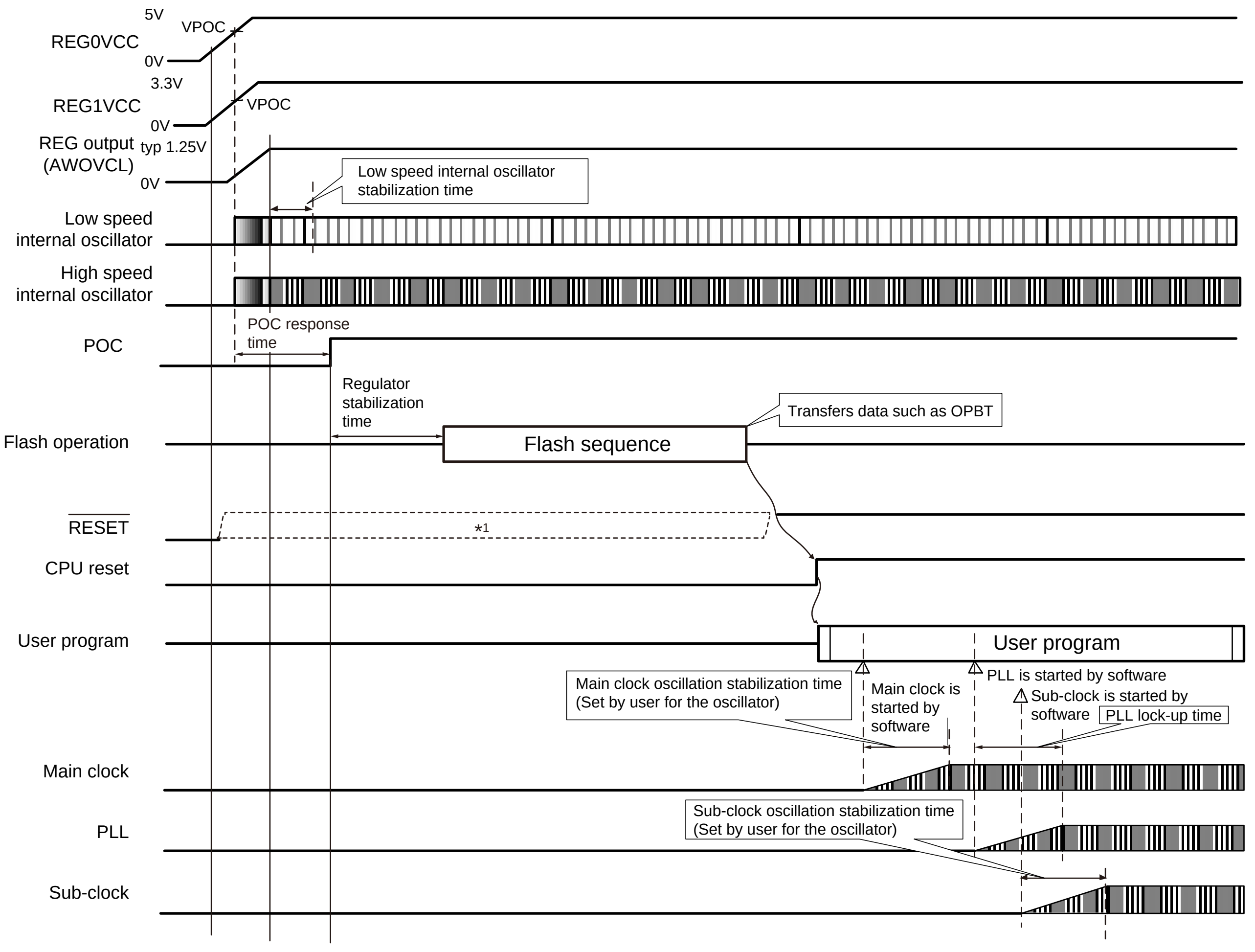
The following figure illustrates the timing of a POCRES.



Note: Delay:
When the REG0VCC and REG1VCC exceeds the VPOC, the POCRES is released after certain delay. When the REG0VCC and REG1VCC falls below the VPOC, the POCRES is asserted after certain delay.
See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for the delay time.

Figure 9A.3 POC Reset Timing

(1) Overview of CPU System Startup after Power-On Clear



Note 1. The timing of releasing the $\overline{\text{RESET}}$ pin depends on the external circuit of the microcontroller.

If the $\overline{\text{RESET}}$ pin is low level, the values after reset of both RESF.RESF8 and RESFR.RESFR8 are set to 1. If the $\overline{\text{RESET}}$ pin is high level, the values after reset of both RESF.RESF8 and RESFR.RESFR8 are set to 0.

Figure 9A.4 When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed

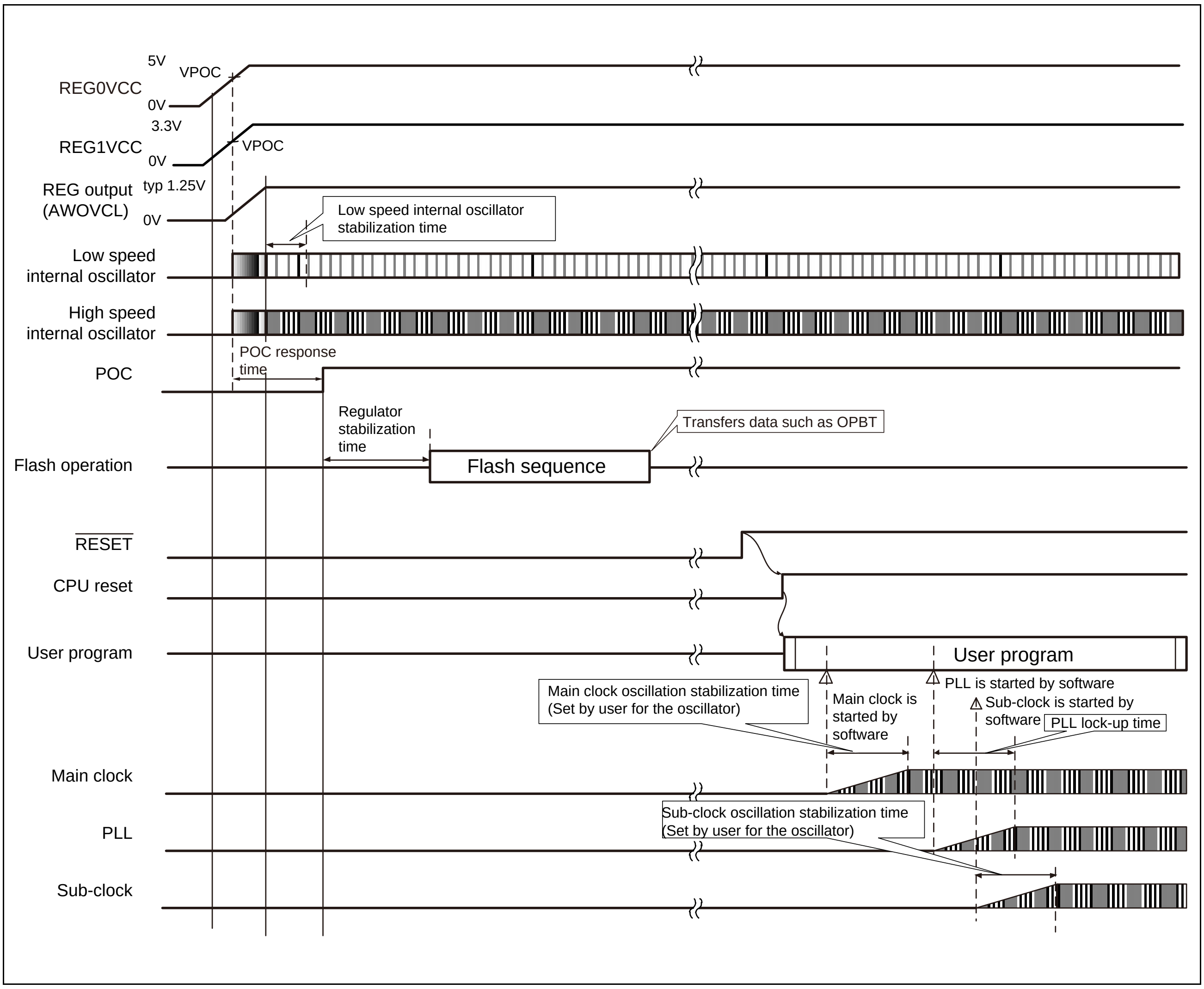


Figure 9A.5 When $\overline{\text{RESET}}$ is Released after the Flash Sequence is Completed

9A.4.3 Low-Voltage Indicator (LVI) Reset

The low-voltage indicator (LVI) constantly compares the power supply voltage REG0VCC with the LVI internal reference voltage VLVIn.

When setting the LVI detection voltage and releasing the LVIRESMK, if REG0VCC falls below the internal reference voltage (REG0VCC < VLVIn), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

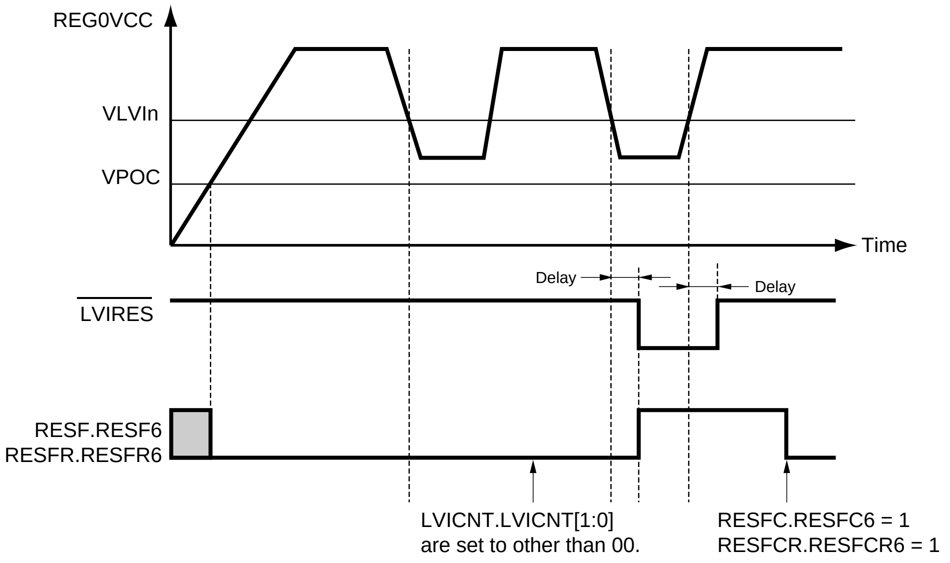
Additionally, the $\overline{\text{LVIRES}}$ flags (bits RESF.RESF6 and RESFR.RESFR6) are set.

After that, even if REG0VCC exceeds VLVIn, bits RESF.RESF6 and RESFR.RESFR6 are not cleared automatically. They are cleared as described below.

- Setting the RESFC.RESFC6 bit to 1 clears the RESF.RESF6 bit.
  Setting the RESFCR.RESFCR6 bit to 1 clears the RESFR.RESFR6 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

For details on the LVI functions, see **Section 11A, Supply Voltage Monitor of RH850/F1KH-D8**.

The following figure illustrates the timing of a $\overline{\text{LVIRES}}$ and bits RESF.RESF6 and RESFR.RESFR6.



Note: Delay:
When REG0VCC falls below VLVIn, $\overline{\text{LVIRES}}$ is asserted and the VLVIn reset flags (RESF.RESF6 and RESFR.RESFR6) are set after a certain delay. After that, when REG0VCC exceeds VLVIn, $\overline{\text{LVIRES}}$ is released after a certain delay.
See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for the delay time.

Figure 9A.6 LVI Reset Timing

9A.4.4 Core Voltage Monitor (CVM) Reset

Core voltage monitor is used to monitor the core voltage inside the microcontroller.

The reset $\overline{\text{CVMRES}}$ is generated if the core voltage is not in the specified voltage range while CVM is enabled. Moreover, the $\overline{\text{CVMRES}}$ flags (RESF.RESF7 and RESFR.RESFR7) are set.

After that, the RESF.RESF7 and RESFR.RESFR7 bits are not automatically cleared even if the core voltage returns to the specified voltage range. The RESF.RESF7 and RESFR.RESFR7 bits are cleared as described below.

- Setting the RESFC.RESFC7 bit to 1 clears the RESF.RESF7 bit.
  Setting the RESFCR.RESFCR7 bit to 1 clears the RESFR.RESFR7 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

If the CVM detects an abnormal high voltage, the power supply to the Isolated area (ISO area) is switched off. Once $\overline{\text{CVMRES}}$ is generated upon high voltage detection, the microcontroller stays in the reset state. To cancel this state, it is mandatory to use the external reset ($\overline{\text{RESET}}$) input. Release the external reset ($\overline{\text{RESET}}$) after the voltage level becomes lower than the high detection voltage.

For details on the CVM function, see **Section 11A, Supply Voltage Monitor of RH850/F1KH-D8**.

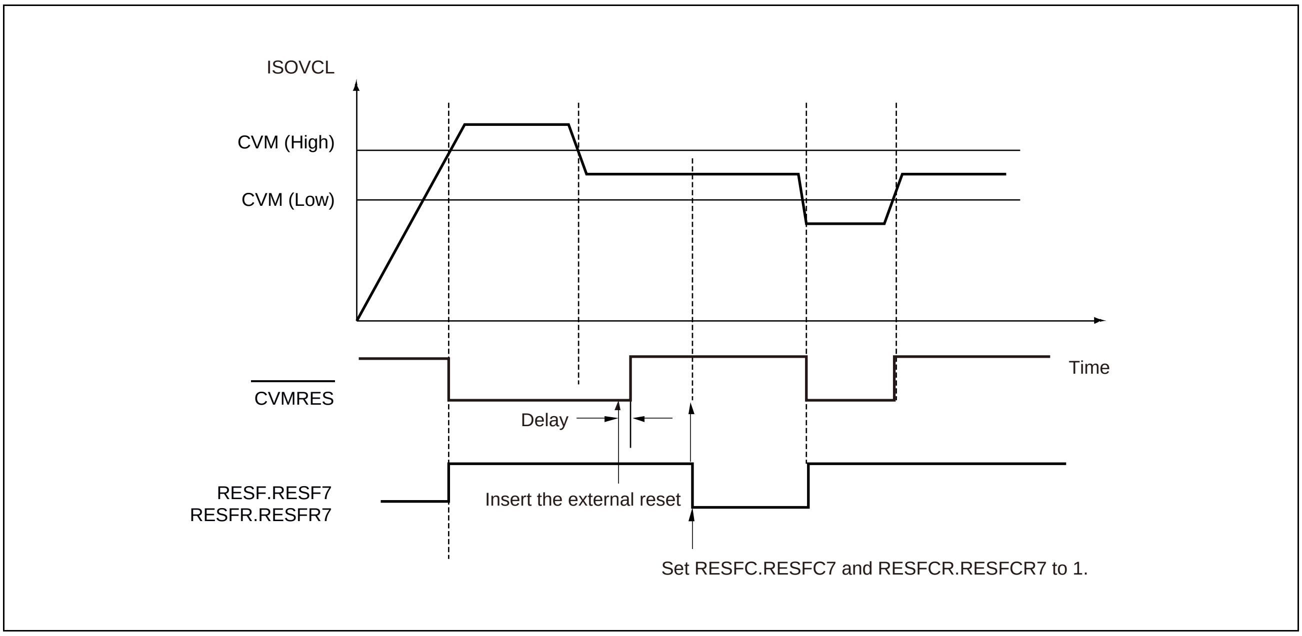


Figure 9A.7 CVM Reset Timing

9A.4.5 External Reset ($\overline{\text{RESET}}$)

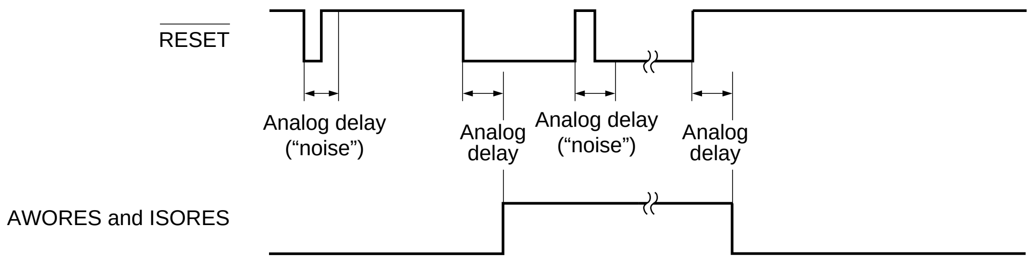
When a low level input is applied to the $\overline{\text{RESET}}$ pin, a reset is asserted and the RESF.RESF8 and RESFR.RESFR8 bits are set.

After that, bits RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if the low-level input to the $\overline{\text{RESET}}$ pin is released. Bits RESF.RESF8 and RESFR.REFR8 are cleared as described below.

- Setting the RESFC.RESFC8 bit to 1 clears the RESF.RESF8 bit.
  Setting the RESFCR.RESFCR8 bit to 1 clears the RESFR.RESFR8 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

The $\overline{\text{RESET}}$ pin includes an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when AWORES and ISORES are generated by the external reset. This figure also shows the effect of the noise filter.



Note: The analog delay is caused by the analog filter. The filter regards pulses up to a certain width as noise and suppresses them. For the minimum $\overline{\text{RESET}}$ pulse width, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**.

Figure 9A.8 External Reset ($\overline{\text{RESET}}$)

9A.4.6 Watchdog Timer (WDTA) Reset

The watchdog timers can be configured to generate a reset if the overflow time is exceeded. After a watchdog timer reset is asserted, the corresponding watchdog timer reset flags (the RESF.RESF1 and RESFR.RESFR1 bits for WDTA0RES, and the RESF.RESF2 and RESFR.RESFR2 bits for WDTA1RES, and the RESF.RESF11 and RESFR.RESFR11 bits for WDTA2RES) are set.

After that, bits RESF.RESF1 and RESFR.RESFR1 (or bits RESF.RESF2 and RESFR.RESFR2 or bits RESF.RESF11 and RESFR.RESFR11) are not cleared automatically, even if WDTA0RES (or WDTA1RES or WDTA2RES) is released.

Bits RESF.RESF1 and RESFR.RESFR1, and bits RESF.RESF2 and RESFR.RESFR2 and bits RESF.RESF11 and RESFR.RESFR11 are cleared as described below.

- WDTA0RES:
  Setting the RESFC.RESFC1 bit to 1 clears the RESF.RESF1 bit.
  Setting the RESFCR.RESFCR1 bit to 1 clears the RESFR.RESFR1 bit.
- WDTA1RES:
  Setting the RESFC.RESFC2 bit to 1 clears the RESF.RESF2 bit.
  Setting the RESFCR.RESFCR2 bit to 1 clears the RESFR.RESFR2 bit.
- WDTA2RES:
  Setting the RESFC.RESFC11 bit to 1 clears the RESF.RESF11 bit.
  Setting the RESFCR.RESFCR11 bit to 1 clears the RESFR.RESFR11 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9A.4.7 Software Reset

The software reset SWRES can be asserted by setting SWRESA.SWRESA to 1.

SWRES sets the reset flag RESF.RESF0 and the RESFR.RESFR0 bit.

RESF.RESF0 and RESFR.RESFR0 are not cleared automatically. RESF.RESF0 and RESFR.RESFR0 are cleared as described below.

- Setting the RESFC.RESFC0 bit to 1 clears the RESF.RESF0 bit.
  Setting the RESFCR.RESFCR0 bit to 1 clears the RESFR.RESFR0 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9A.4.8 Clock Monitor (CLMA) Reset

The clock monitors can generate the following resets:

- $\overline{\text{CLMA0RES}}$, if a frequency abnormality in HS IntOSC is detected
- $\overline{\text{CLMA1RES}}$, if a frequency abnormality in MainOSC is detected
- $\overline{\text{CLMA2RES}}$, if a frequency abnormality in PLL0 is detected
- $\overline{\text{CLMA3RES}}$, if a frequency abnormality in PLL1 is detected

When the Clock Monitor detects frequency abnormality of the respective clocks, resets $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$ are generated.

In addition, flags $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$ (RESF.RESF3, RESFR.RESFR3, RESF.RESF4, RESFR.RESFR4, RESF.RESF5, RESFR.RESFR5, RESF.RESF14, and RESFR.RESFR14) are set.

These flags are not cleared automatically. They are cleared as described below.

- $\overline{\text{CLMA0RES}}$:
  Setting the RESFC.RESFC3 bit to 1 clears the RESF.RESF3 bit.
  Setting the RESFCR.RESFCR3 bit to 1 clears the RESFR.RESFR3 bit.
- $\overline{\text{CLMA1RES}}$:
  Setting the RESFC.RESFC4 bit to 1 clears the RESF.RESF4 bit.
  Setting the RESFCR.RESFCR4 bit to 1 clears the RESFR.RESFR4 bit.
- $\overline{\text{CLMA2RES}}$:
  Setting the RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.
  Setting the RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.
- $\overline{\text{CLMA3RES}}$:
  Setting the RESFC.RESFC14 bit to 1 clears the RESF.RESF14 bit.
  Setting the RESFCR.RESFCR14 bit to 1 clears the RESFR.RESFR14 bit
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9A.4.9 Debugger Reset

Debugger reset ($\overline{\text{DBRES}}$) is generated via a debugger command. $\overline{\text{DBRES}}$ activates PURES, and therefore operates in the same way as the power-on clear reset POCRES:

- The clock generators are reset and stop operating. The clock generators should be restarted after release from the reset state.
- The reset factor register RESF and the redundant reset factor register RESFR are cleared (Bits RESF9 and RESFR9 are set to 1 after initialization).

Section 9BC Reset Controller of RH850/F1KM

9BC.1 Overview

Several system reset functions are provided in order to initialize CPU core and peripheral functions as well as their associated registers.

A reset can be caused by the following events:

Table 9BC.1 Reset Sources and Reset Targets (RH850/F1KM-S4)

Reset Source	Symbol	RH850/F1KM-S4
External reset	$\overline{\text{RESET}}$	✓
Power-On Clear	POCRES	✓
Watchdog timer reset	WDTA0RES	✓
	WDTA1RES	✓
Clock monitor reset	$\overline{\text{CLMA0RES}}$	✓
	$\overline{\text{CLMA1RES}}$	✓
	$\overline{\text{CLMA2RES}}$	✓
	$\overline{\text{CLMA3RES}}$	✓
Low-voltage indicator reset	$\overline{\text{LVIRES}}$	✓
Software reset	SWRES	✓
Debugger reset	$\overline{\text{DBRES}}$	✓
Core voltage monitor reset	$\overline{\text{CVMRES}}$	✓
Transition to DeepSTOP mode		✓

Table 9BC.2 Reset Sources and Reset Targets (RH850/F1KM-S1)

Reset Source	Symbol	RH850/F1KM-S1
External reset	$\overline{\text{RESET}}$	✓
Power-On Clear	POCRES	✓
Watchdog timer reset	WDTA0RES	✓
	WDTA1RES	✓
Clock monitor reset	$\overline{\text{CLMA0RES}}$	✓
	$\overline{\text{CLMA1RES}}$	✓
	$\overline{\text{CLMA2RES}}$	—
	$\overline{\text{CLMA3RES}}$	✓
Low-voltage indicator reset	$\overline{\text{LVIRES}}$	✓
Software reset	SWRES	✓
Debugger reset	$\overline{\text{DBRES}}$	✓
Core voltage monitor reset	$\overline{\text{CVMRES}}$	✓
Transition to DeepSTOP mode		✓

9BC.1.1 Reset Sources

Reset levels and reset sources are shown below.

Various reset sources are assigned to the different levels of the reset.

Table 9BC.3 Reset Sources and Reset Targets

Reset Level	Reset Source	Clock Generation (except PLL)/Real-Time Clock/CVM/LVI	Always-On area (AWO area) Modules*[1]	Isolated area (ISO area) Modules*[2]
1	Power-On Clear (POCRES) Debugger reset ($\overline{\text{DBRES}}$)	Reset	Reset	Reset
2	External reset ($\overline{\text{RESET}}$) Watchdog timer reset (WDTA0RES, WDTA1RES) Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, $\overline{\text{CLMA3RES}}$) Core voltage monitor reset ($\overline{\text{CVMRES}}$) Low voltage indicator reset ($\overline{\text{LVIRES}}$) Software reset (SWRES)	Not reset target*[3]	Reset	Reset
3	Reset by DeepSTOP mode	Not reset target	Not reset target	Reset

Note 1. Clock generator, real-time clock, CVM, and LVI are excluded.

Note 2. PLL is included.

Note 3. In clock monitor reset, oscillator-related registers for clock monitoring are initialized.

Reset level 1: Initializes the entire microcontroller.

Reset level 2: For a quick return to normal operating mode by eliminating the oscillator stabilization time, initializes the entire microcontroller except for the clock generator and the real-time clock.

Reset level 3: At the transition to DeepSTOP mode, initializes all the Isolated area (ISO area).

If each reset is generated, local RAM data and global RAM data*[1] are not guaranteed to retaine.

If each reset is generated, retention RAM data can be retained as long as the power-supply voltage (REGVCC) does not fall below the RAM retention voltage (VVLVI).

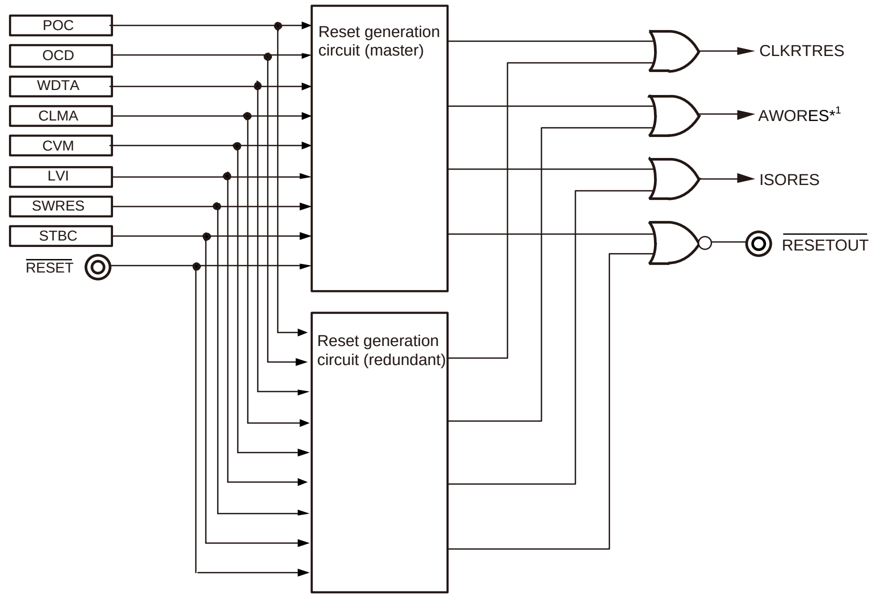
In this case, data value writing to retention RAM address will be before or after writing.

Note 1. The global RAM is not supported in RH850/F1KM-S1.

9BC.1.2 Reset Controller Redundancy

The reset controller of the microcontroller has a redundant configuration, and includes duplicated reset generation circuits. Such configuration enables initialization of the reset targeted area without failure even if one of the two reset generation circuits fails.

The configuration of the reset generation circuits is shown in the figure below.



Note 1. Real-time clock is excluded.

Figure 9BC.1 Reset Controller Redundancy

At the generation of a reset, the same reset source signal is input to two reset generation circuits.

According to the reset source, the two reset generation circuits output the Always-On area (AWO area) reset signal (AWORES), Isolated area (ISO area) reset signal (ISORES), clock generator/real-time clock reset signal (CLKRTRES), and $\overline{\text{RESETOUT}}$ signal.

The AWORES, ISORES, CLKRTRES, and $\overline{\text{RESETOUT}}$ signals are generated by executing the logical OR of the signals output from two reset generation circuits. Thus, a reset signal is generated normally even if one of the two reset generation circuits fails.

Whether a reset generation circuit operates normally can be checked by reading and comparing the reset factor registers of the respective reset generation circuits.

9BC.1.3 Reset Output ($\overline{\text{RESETOUT}}$)

When a reset source of reset level 1 or 2 is generated, a reset output signal ($\overline{\text{RESETOUT}}$) is output to the outside. Reset output is used to reset external devices at the same time as a reset is generated inside the microcontroller.

For details, see **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** .

9BC.1.4 Reset Flag

To identify a reset source, two registers with a flag for each reset source are provided. The main elements of the reset controller are shown in **Figure 9BC.2, Block Diagram of the Reset Controller**.

9BC.1.5 Clock Supply

The clock supply to the reset controller is shown in the following table.

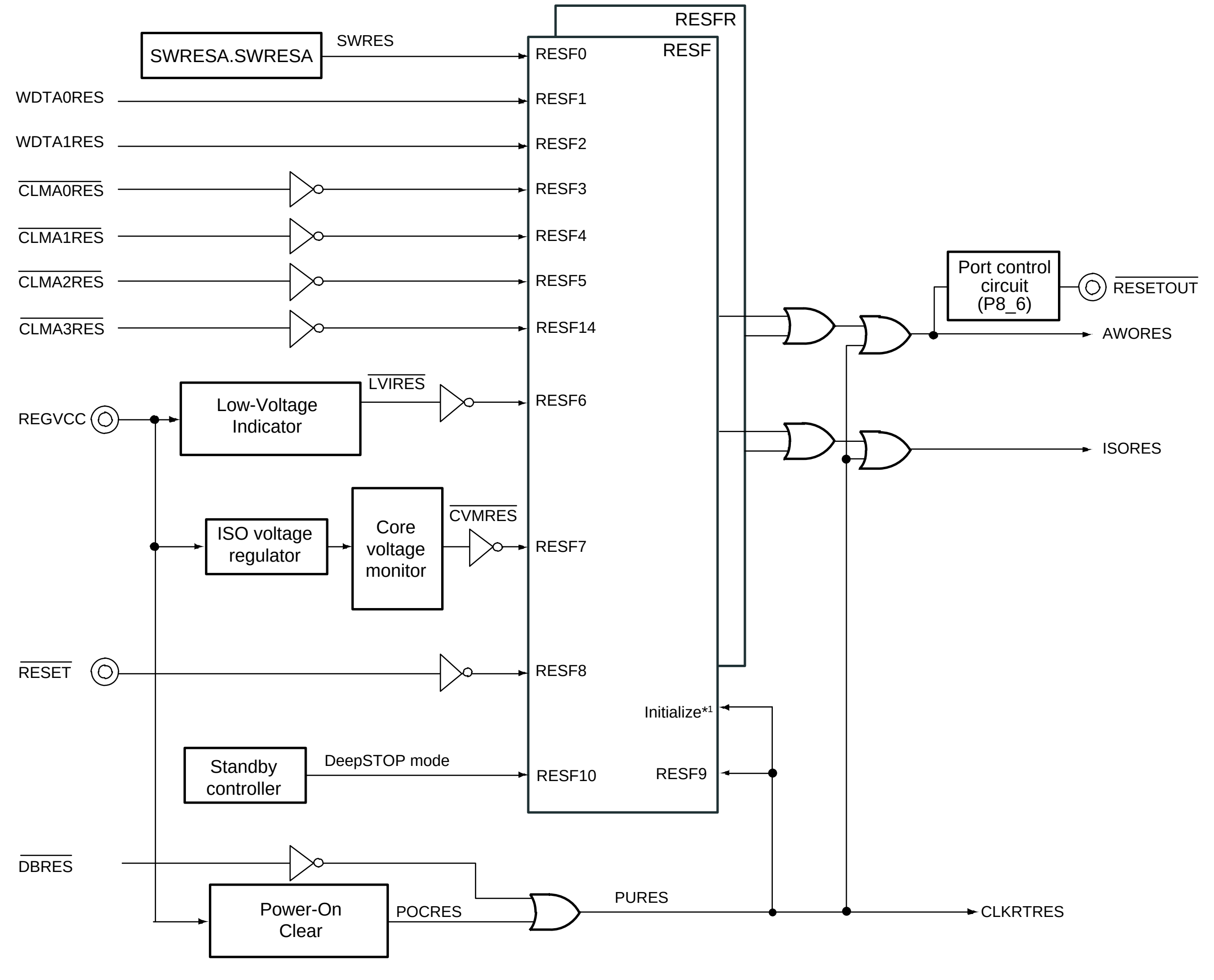
Table 9BC.4 Clock Supply (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
Reset	Register access clock	CPUCLK_UL

9BC.2 Configuration

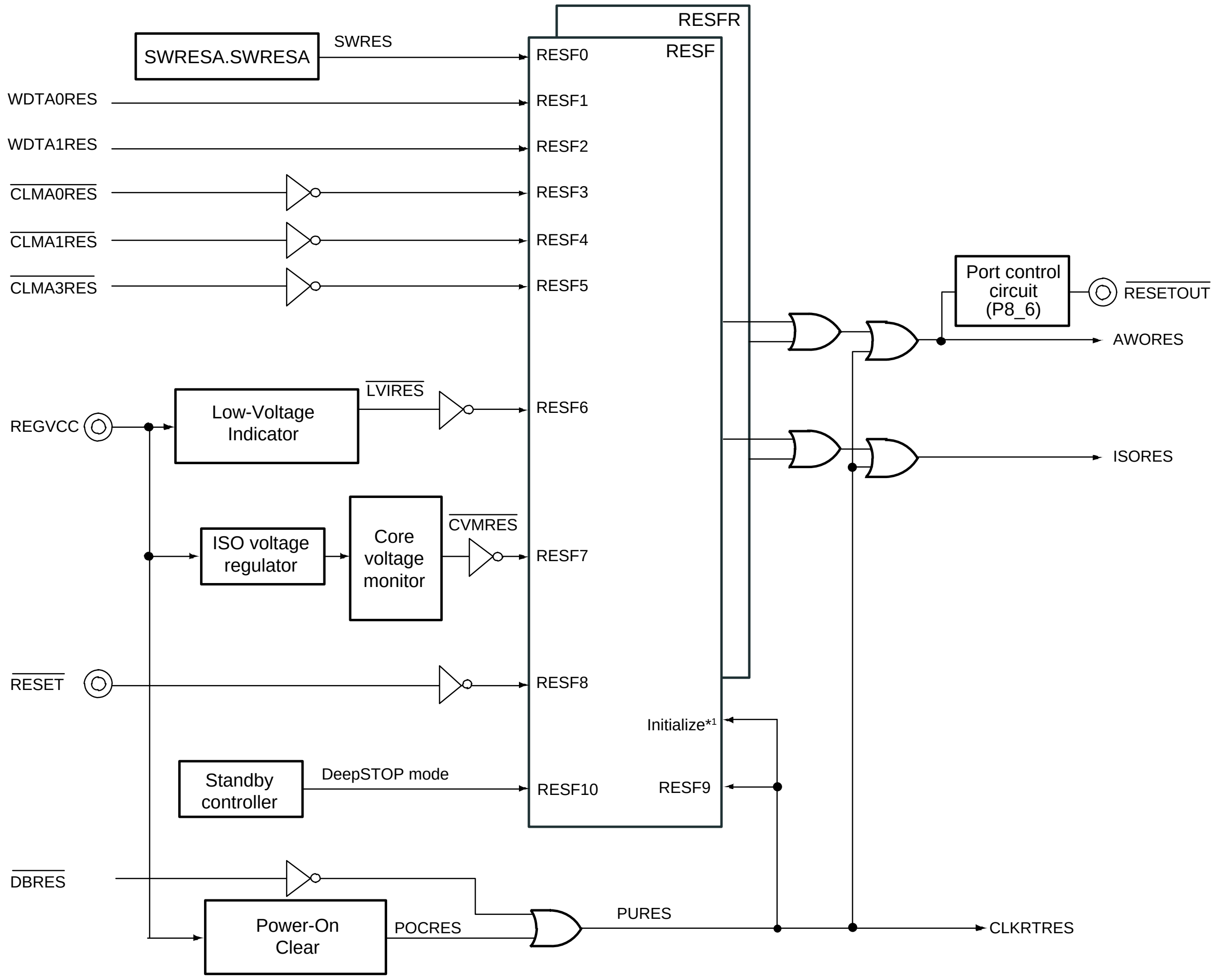
9BC.2.1 Block Diagram

Block diagram of reset circuits are shown below.



Note 1. Registers RESF and RESFR are initialized by PURES.
After an initialization, bits RESF9 and RESFR9 are set to 1.

Figure 9BC.2 Block Diagram of the Reset Controller (RH850/F1KM-S4)



Note 1. Registers RESF and RESFR are initialized by PURES.
After an initialization, bits RESF9 and RESFR9 are set to 1.

Figure 9BC.3 Block Diagram of the Reset Controller (RH850/F1KM-S1)

(1) Reset Signals

The reset controller manages the generation of three reset signals upon occurrence of reset signals from various reset sources:

- Always-On area (AWO area) reset (AWORES)
  AWORES is generated by all reset sources except the transition to DeepSTOP mode.
  AWORES resets all modules in the Always-On area (AWO area) except clock generation circuit, real-time clock, core voltage monitor, and low-voltage detection circuit.
- Isolated area (ISO area) reset (ISORES)
  ISORES is generated by all reset sources.
  ISORES resets all modules (including PLL) in the Isolated area (ISO area).
- CLKRTRES
  CLKRTRES is generated by the power-on clear or debugger reset sources.
  CLKRTRES resets the clock generation circuit (excluding PLL) and real-time clock.

The power-up reset (PURES) is caused by the power-on clear and debugger reset sources.

Following the generation of an AWORES reset, with the exception of the PLL, all clock-generation circuits that were operating at the time (LS IntOSC, HS IntOSC, MainOSC, SubOSC) continue to operate. On the generation of a $\overline{\text{CLMA0RES}}$ reset, the HS IntOSC that was the target for CLMA0 monitoring is reset. On the generation of a $\overline{\text{CLMA1RES}}$ reset, the MainOSC that was the target for CLMA1 monitoring is reset.

The PURES initializes all of the clock generation circuits. It is necessary to restart the clock generation circuit after recovery from the PURES.

The CPU reset is the Isolated area (ISO area) reset (ISORES) to the CPU sub system.

(2) Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) hold a flag for each reset source, and this flag is set when the corresponding reset is asserted.

All reset flags except RESF9 and RESFR9 are initialized by a power-up reset (PURES). (Bits RESF9 and RESFR9 are set to 1 after initialization.) In addition, all the bits can be cleared by software.

For details, see **Section 9BC.1.4, Reset Flag**.

(3) On-Chip Module Resets

(a) Watchdog Timer Resets

The watchdog timers can generate two types of resets: WDTA0RES and WDTA1RES.

For details, see **Section 9BC.4.6, Watchdog Timer (WDTA) Reset**.

(b) Clock Monitor Resets

The clock monitors can generate four resets: $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$.

For details, see **Section 9BC.4.8, Clock Monitor (CLMA) Reset**.

(c) Debugger Reset

A reset is generated by a command from a debugger. This leads to a generation of power-up reset PURES.

For details, see **Section 9BC.4.9, Debugger Reset**.

(4) Software Controlled Reset (SWRES)

A software reset SWRES can be generated by use of the software reset register SWRESA.

For details, see **Section 9BC.4.7, Software Reset**.

(5) Reset Output Signal

During reset and after release from the reset, port P8_6 outputs low level as $\overline{\text{RESETOUT}}$ function.

For details, see **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

(6) Power Supply Monitoring

The following power supply detection circuits observe the level of the external power supply REGVCC.

(a) Low-Voltage Indicator

The low-voltage indicator (LVI) generates the $\overline{\text{LVIRES}}$ reset, if the voltage level of REGVCC drops below a certain level. The level can be adjusted and the $\overline{\text{LVIRES}}$ can be masked.

For details, see **Section 9BC.4.3, Low-Voltage Indicator (LVI) Reset**.

(b) Power-On Clear

The power-on clear circuit (POC) continuously compares the power supply voltage REGVCC with an internal reference voltage. Thus, a reset is generated when the power supply voltage goes below a certain level.

For details, see **Section 9BC.4.2, Power-On Clear (POC) Reset.**

(c) Core Voltage Monitor

A reset can be generated when the core voltage monitor (CVM) detects over- or undervoltage in core voltage. (Output/not output can be set by option byte.)

For details, see **Section 9BC.4.4, Core Voltage Monitor (CVM) Reset**.

(7) Masking of Reset Sources in Debugging Mode

The following reset sources can be masked during debugging:

Table 9BC.5 Reset Sources to be Masked during Debugging

Reset Source	Maskable/Non-maskable
Power-on clear (POCRES)	—
Debugger reset ($\overline{\text{DBRES}}$)	—
External reset ($\overline{\text{RESET}}$)	✓
Low-voltage indicator reset ($\overline{\text{LVIRES}}$)	✓
Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, $\overline{\text{CLMA3RES}}$)	✓
Watchdog timer reset (WDTA0RES, WDTA1RES)	✓
Core voltage monitor reset ($\overline{\text{CVMRES}}$)	✓
Software reset (SWRES)	✓
Reset by DeepSTOP mode	—

9BC.3 Registers

This section contains a description of all registers of the reset controller.

9BC.3.1 Reset Controller Registers Overview

The reset controller is controlled and operated by the following registers:

Table 9BC.6 Reset Controller Registers Overview (RH850/F1KM-S4)

Module Name	Register Name	Symbol	Address
RESCTL	Reset flag registers		
	Reset factor register	RESF	FFF8 0760$_H$
	Reset factor clear register	RESFC	FFF8 0768$_H$
	Redundant reset factor register	RESFR	FFF8 0860$_H$
	Redundant reset factor clear register	RESFCR	FFF8 0868$_H$
	Software reset control register		
	Software reset register	SWRESA	FFF8 0A04$_H$

Table 9BC.7 Reset Controller Registers Overview (RH850/F1KM-S1)

Module Name	Register Name	Symbol	Address
RESCTL	Reset flag registers		
	Reset factor register	RESF	FFF8 0760$_H$
	Reset factor clear register	RESFC	FFF8 0768$_H$
	Redundant reset factor register	RESFR	FFF8 0860$_H$
	Redundant reset factor clear register	RESFCR	FFF8 0868$_H$
	Software reset control register		
	Software reset register	SWRESA	FFF8 0A04$_H$
	Cyclic RUN mode reset vector address register		
	Cyclic RUN mode RBASE register	CYCRBASE	FFF8 3600$_H$

NOTES

1. For the LVI related, RAM store related, and CVM related registers, see **Section 11BC, Supply Voltage Monitor of RH850/F1KM**.
2. As for the protection registers, see **Section 5, Write-Protected Registers**.

9BC.3.2 Details of Reset Flag Registers

9BC.3.2.1 RESF — Reset Factor Register

This register contains information about which type of resets occurred after the last power-on clear reset. This register is initialized by a power-up reset PURES.

Each reset condition sets the corresponding flag in the register.

For example, if a clock monitor reset $\overline{\text{CLMA0RES}}$ occurs after a watchdog timer reset WDTA0RES, RESF reads 0000 000A$_{\text{H}}$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0760$_{\text{H}}$

Value after reset: 0000 0200$_{\text{H}}$ / 0000 0300$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESF 14	—	—	—	RESF 10	RESF9	RESF8	RESF7	RESF6	RESF5	RESF4	RESF3	RESF2	RESF1	RESF0
Value after reset	0	0	0	0	0	0	1	1/0^{*1}	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9BC.5, When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed**, When $\overline{\text{RESET}}$ is released before Execution of Flash Sequence.

Table 9BC.8 RESF Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	RESF14	RH850/F1KM-S4: CLMA3 reset flag 0: No reset occurred 1: Reset has occurred RH850/F1KM-S1: When read, the value after reset is returned.
13 to 11	Reserved	When read, the value after reset is returned.
10	RESF10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESF9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESF8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESF7	CVM reset flag 0: No reset occurred 1: Reset has occurred

Table 9BC.8 RESF Register Contents

Bit Position	Bit Name	Function
6	RESF6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESF5	RH850/F1KM-S4: CLMA2 reset flag 0: No reset occurred 1: Reset has occurred RH850/F1KM-S1: CLMA3 reset flag 0: No reset occurred 1: Reset has occurred
4	RESF4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESF3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESF2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESF1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESF0	Software reset flag 0: No reset occurred 1: Reset has occurred

9BC.3.2.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0768_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFC 14	—	—	—	RESFC 10	RESFC 9	RESFC 8	RESFC 7	RESFC 6	RESFC 5	RESFC 4	RESFC 3	RESFC 2	RESFC 1	RESFC 0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	W	R	R	R	W	W	W	W	W	W	W	W	W	W	W

Table 9BC.9 RESFC Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write “0”.
14	RESFC14	RH850/F1KM-S4: CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag RH850/F1KM-S1: When writing, write “0”.
13 to 11	Reserved	When writing, write “0”.
10	RESFC10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFC9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFC8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFC7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFC6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFC5	RH850/F1KM-S4: CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag RH850/F1KM-S1: CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag

Table 9BC.9 RESFC Register Contents

Bit Position	Bit Name	Function
4	RESFC4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFC3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFC2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFC1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFC0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9BC.3.2.3 RESFR — Redundant Reset Factor Register

This register is a duplication of the reset factor register. This register is initialized by a power-up reset PURES.

In accordance with the setting conditions for each bit in the reset factor register, the same bits are set in this register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0860_H

Value after reset: 0000 0200_H / 0000 0300_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFR 14	—	—	—	RESFR 10	RESFR 9	RESFR 8	RESFR 7	RESFR 6	RESFR 5	RESFR 4	RESFR 3	RESFR 2	RESFR 1	RESFR 0
Value after reset	0	0	0	0	0	0	1	1/0*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9BC.5, When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed**, When $\overline{\text{RESET}}$ is released before Execution of Flash Sequence.

Table 9BC.10 RESFR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	RESFR14	RH850/F1KM-S4: CLMA3 reset flag 0: No reset occurred 1: Reset has occurred RH850/F1KM-S1: When read, the value after reset is returned.
13 to 11	Reserved	When read, the value after reset is returned.
10	RESFR10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESFR9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESFR8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESFR7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESFR6	LVI reset flag 0: No reset occurred 1: Reset has occurred

Table 9BC.10 RESFR Register Contents

Bit Position	Bit Name	Function
5	RESFR5	RH850/F1KM-S4: CLMA2 reset flag 0: No reset occurred 1: Reset has occurred RH850/F1KM-S1: CLMA3 reset flag 0: No reset occurred 1: Reset has occurred
4	RESFR4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESFR3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESFR2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESFR1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESFR0	Software reset flag 0: No reset occurred 1: Reset has occurred

9BC.3.2.4 RESFCR — Redundant Reset Factor Clear Register

This register clears the reset flags of the RESFR.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0868$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RESFCR14	—	—	—	RESFCR10	RESFCR9	RESFCR8	RESFCR7	RESFCR6	RESFCR5	RESFCR4	RESFCR3	RESFCR2	RESFCR1	RESFCR0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	W	R	R	R	W	W	W	W	W	W	W	W	W	W	W

Table 9BC.11 RESFCR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write “0”.
14	RESFCR14	RH850/F1KM-S4: CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag RH850/F1KM-S1: When writing, write “0”
13 to 11	Reserved	When writing, write “0”.
10	RESFCR10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFCR9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFCR8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFCR7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFCR6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFCR5	RH850/F1KM-S4: CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag RH850/F1KM-S1: CLMA3 reset flag clear 0: Do not clear flag 1: Clear flag

Table 9BC.11 RESFCR Register Contents

Bit Position	Bit Name	Function
4	RESFCR4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFCR3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFCR2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFCR1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFCR0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9BC.3.3 Details of Software Reset Control Register

9BC.3.3.1 SWRESA — Software Reset Register

This register is used to generate a software reset SWRES. The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access:	This register is a write-only register that can be written in 32-bit units.
Address:	FFF8 0A04$_H$
Value after reset:	Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRESA
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9BC.12 SWRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write “0”.
0	SWRESA	Software reset trigger 0: No Software reset trigger is generated. 1: Software reset trigger is generated.

9BC.3.4 Details of Cyclic RUN Mode Reset Vector Address Register

CYCRBASE Register is supported in RH850/F1KM-S1.

9BC.3.4.1 CYCRBASE — Cyclic RUN Mode RBASE Register

This register is used to specify reset vector address (RBASE) of the CPU, when the CPU returns to Cyclic RUN mode from DeepSTOP mode. The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Update of this register is only permitted when the chip is in RUN mode. Do not change the value in Cyclic RUN mode.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can only be read or written in 32-bit units.

Address: FFF8 3600_H

Value after reset: FEBF 8000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CYCRBASE[31:16]															
Value after reset	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCRBASE[15:9]							—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Table 9BC.13 CYCRBASE Register Contents

Bit Position	Bit Name	Function
31 to 9	CYCRBASE	Reset vector base address (RBASE) in Cyclic RUN mode. Specifies the RBASE value in Cyclic RUN mode. The default value is set to the start address of the retention RAM.
8 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

9BC.4 Functional Description

9BC.4.1 Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) provide reset flags for each reset source.

If a reset has occurred, the corresponding flag is set. According to this, the source of the reset is evaluated.

RESF and RESFR are initialized by a power-up reset PURES (POCRES or $\overline{\text{DBRES}}$) (though bits RESF9 and RESFR9 are set to 1 after initialization). In addition, flags in RESF and RESFR can be cleared by the reset factor clear register (RESFC) and the redundant reset factor clear register (RESFCR).

Each reset source can set the corresponding flag independently from other reset sources.

9BC.4.2 Power-On Clear (POC) Reset

The power-on clear circuit (POC) constantly compares the power supply voltage REGVCC with the internal reference voltage VPOC. It ensures that the microcontroller only operates as long as the power supply exceeds a certain level.

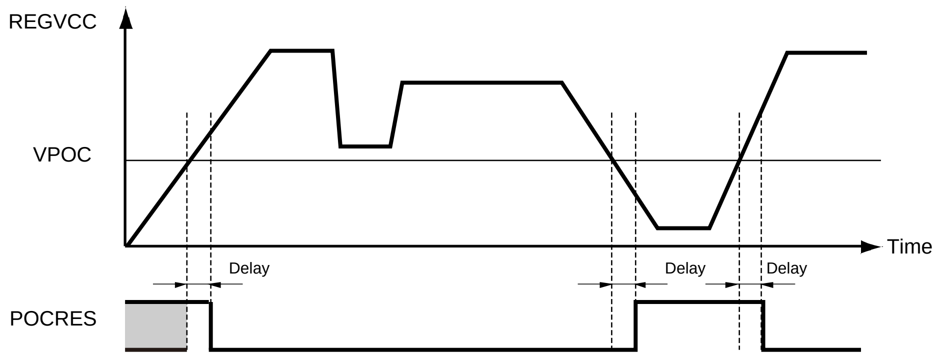
If REGVCC falls below the internal reference voltage (REGVCC < VPOC), the internal reset signal POCRES and a power-up reset PURES are generated.

For details on the specification of the internal voltage reference level VPOC, see **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.

The reset factor register (RESF) and the redundant reset factor register (RESFR) are cleared by the power-on clear reset. RESF9 and RESFR9 are set to 1 after initialization.

The power-on clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level VPOC.

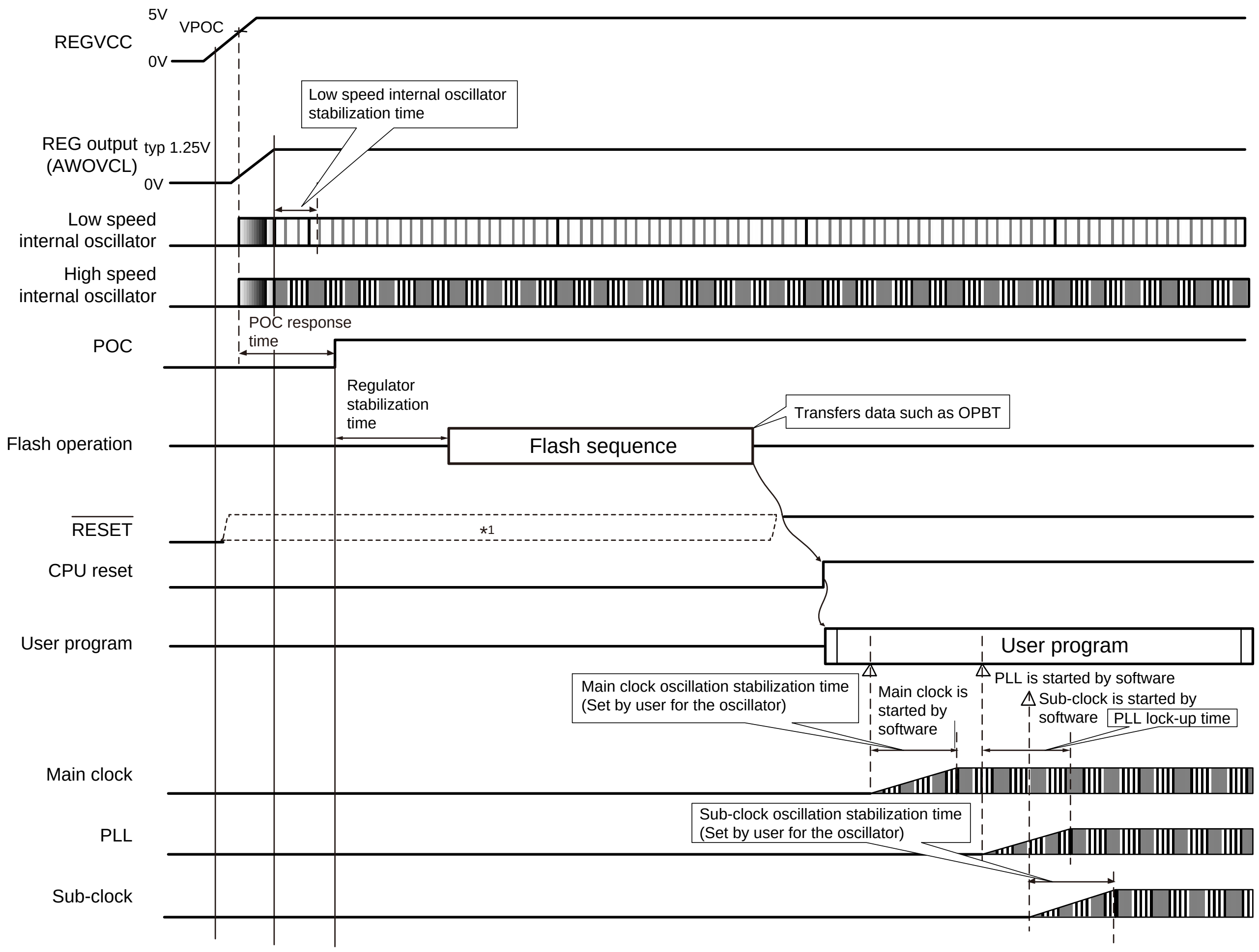
The following figure illustrates the timing of a POCRES.



Note: Delay:
When the REGVCC exceeds the VPOC, the POCRES is released after certain delay. When the REGVCC falls below the VPOC, the POCRES is asserted after certain delay.
See **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for the delay time.

Figure 9BC.4 POC Reset Timing

(1) Overview of CPU System Startup after Power-On Clear



Note 1. The timing of releasing the $\overline{\text{RESET}}$ pin depends on the external circuit of the microcontroller.

If the $\overline{\text{RESET}}$ pin is low level, the values after reset of both RESF.RESF8 and RESFR.RESFR8 are set to 1. If the $\overline{\text{RESET}}$ pin is high level, the values after reset of both RESF.RESF8 and RESFR.RESFR8 are set to 0.

Figure 9BC.5 When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed

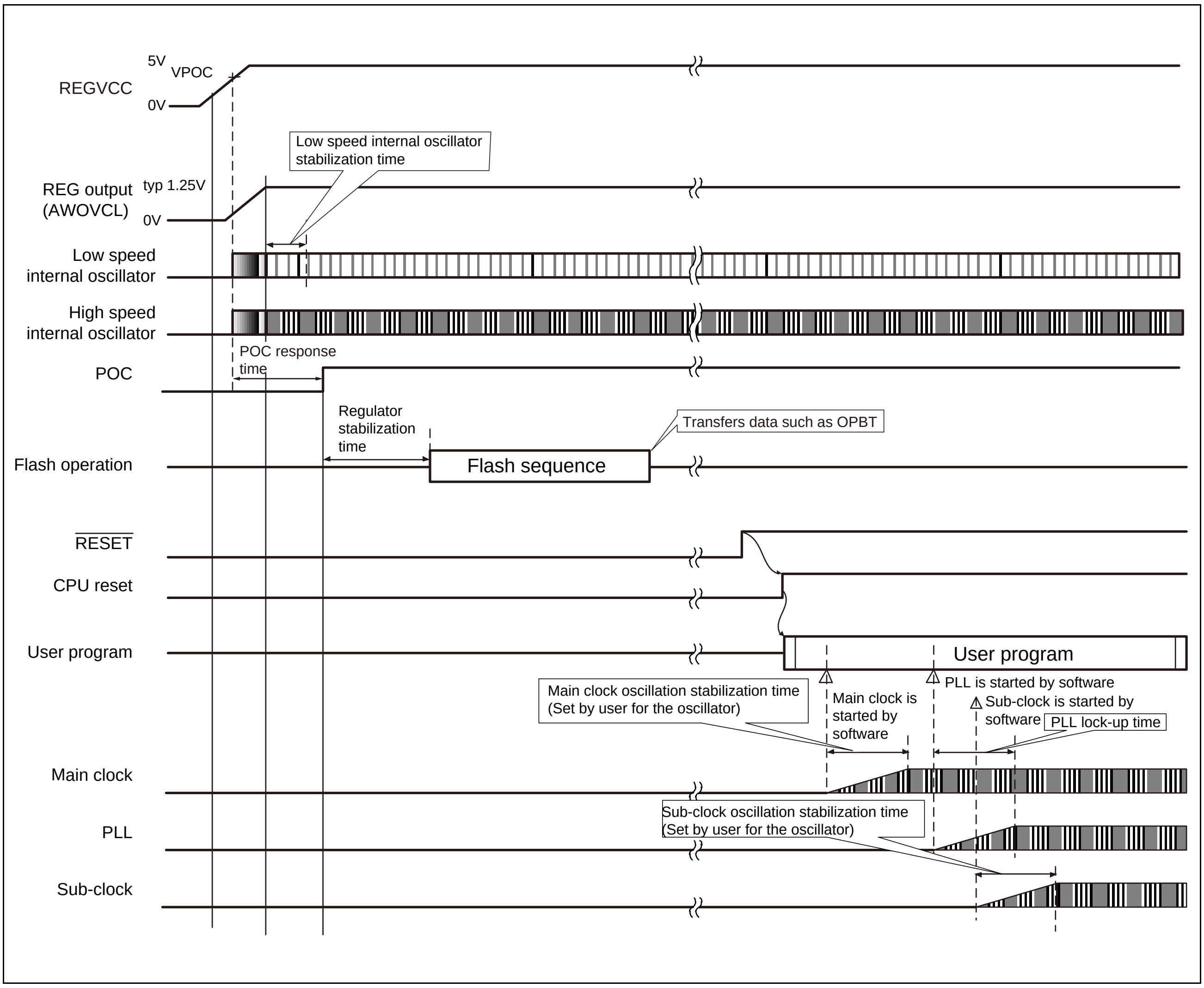


Figure 9BC.6 When $\overline{\text{RESET}}$ is Released after the Flash Sequence is Completed

9BC.4.3 Low-Voltage Indicator (LVI) Reset

The low-voltage indicator (LVI) constantly compares the power supply voltage REGVCC with the LVI internal reference voltage VLVIn.

When setting the LVI detection voltage and releasing the LVIRESMK, if REGVCC falls below the internal reference voltage (REGVCC < VLVIn), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

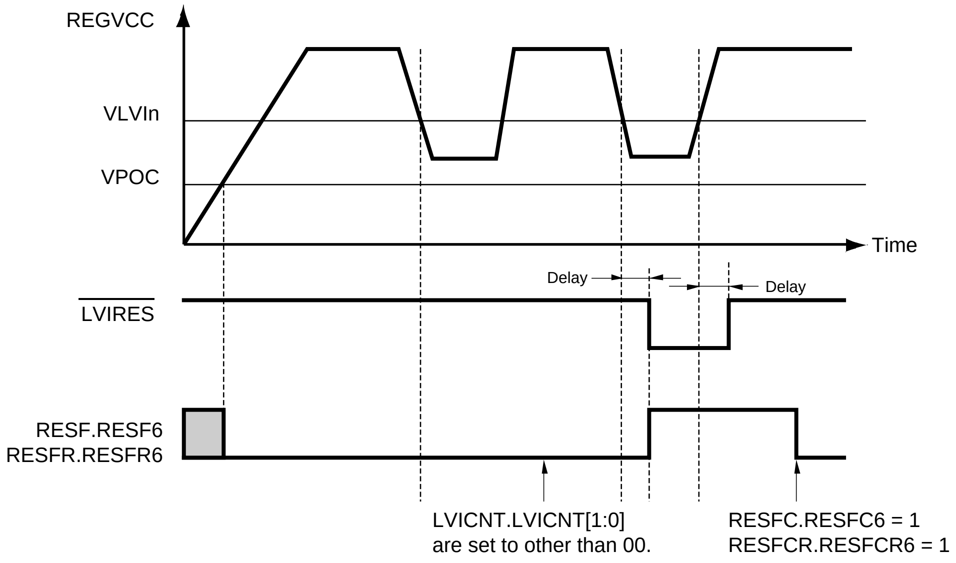
Additionally, the $\overline{\text{LVIRES}}$ flags (bits RESF.RESF6 and RESFR.RESFR6) are set.

After that, even if REGVCC exceeds VLVIn, bits RESF.RESF6 and RESFR.RESFR6 are not cleared automatically. They are cleared as described below.

- Setting the RESFC.RESFC6 bit to 1 clears the RESF.RESF6 bit.
  Setting the RESFCR.RESFCR6 bit to 1 clears the RESFR.RESFR6 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

For details on the LVI functions, see **Section 11BC, Supply Voltage Monitor of RH850/F1KM**.

The following figure illustrates the timing of a $\overline{\text{LVIRES}}$ and bits RESF.RESF6 and RESFR.RESFR6.



Note: Delay:
When REGVCC falls below VLVIn, $\overline{\text{LVIRES}}$ is asserted and the VLVIn reset flags (RESF.RESF6 and RESFR.RESFR6) are set after a certain delay. After that, when REGVCC exceeds VLVIn, $\overline{\text{LVIRES}}$ is released after a certain delay.
See **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for the delay time.

Figure 9BC.7 LVI Reset Timing

9BC.4.4 Core Voltage Monitor (CVM) Reset

Core voltage monitor is used to monitor the core voltage inside the microcontroller.

The reset $\overline{\text{CVMRES}}$ is generated if the core voltage is not in the specified voltage range while CVM is enabled. Moreover, the $\overline{\text{CVMRES}}$ flags (RESF.RESF7 and RESFR.RESFR7) are set.

After that, the RESF.RESF7 and RESFR.RESFR7 bits are not automatically cleared even if the core voltage returns to the specified voltage range. The RESF.RESF7 and RESFR.RESFR7 bits are cleared as described below.

- Setting the RESFC.RESFC7 bit to 1 clears the RESF.RESF7 bit.
  Setting the RESFCR.RESFCR7 bit to 1 clears the RESFR.RESFR7 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

If the CVM detects an abnormal high voltage, the power supply to the Isolated area (ISO area) is switched off. Once $\overline{\text{CVMRES}}$ is generated upon high voltage detection, the microcontroller stays in the reset state. To cancel this state, it is mandatory to use the external reset ($\overline{\text{RESET}}$) input. Release the external reset ($\overline{\text{RESET}}$) after the voltage level becomes lower than the high detection voltage.

For details on the CVM function, see **Section 11BC, Supply Voltage Monitor of RH850/F1KM**.

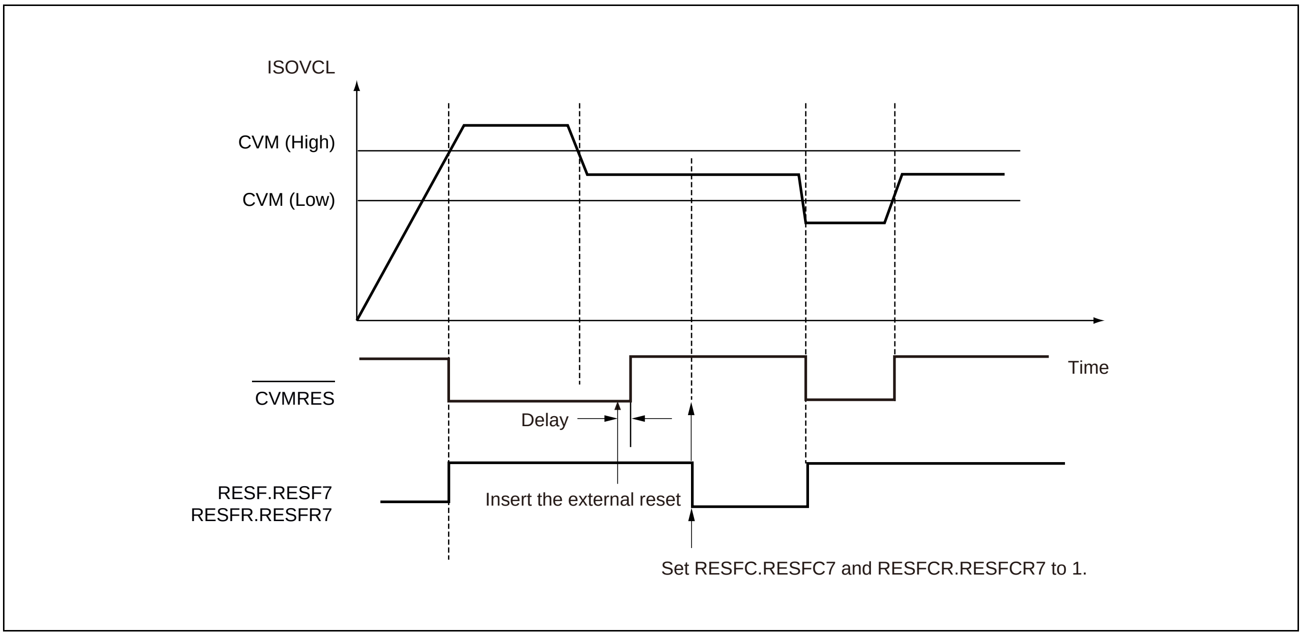


Figure 9BC.8 CVM Reset Timing

9BC.4.5 External Reset ($\overline{\text{RESET}}$)

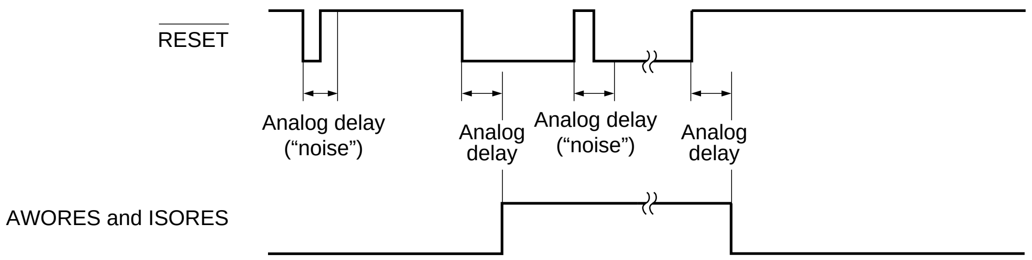
When a low level input is applied to the $\overline{\text{RESET}}$ pin, a reset is asserted and the RESF.RESF8 and RESFR.RESFR8 bits are set.

After that, bits RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if the low-level input to the $\overline{\text{RESET}}$ pin is released. Bits RESF.RESF8 and RESFR.REFR8 are cleared as described below.

- Setting the RESFC.RESFC8 bit to 1 clears the RESF.RESF8 bit.
  Setting the RESFCR.RESFCR8 bit to 1 clears the RESFR.RESFR8 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

The $\overline{\text{RESET}}$ pin includes an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when AWORES and ISORES are generated by the external reset. This figure also shows the effect of the noise filter.



Note: The analog delay is caused by the analog filter. The filter regards pulses up to a certain width as noise and suppresses them. For the minimum $\overline{\text{RESET}}$ pulse width, see **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.

Figure 9BC.9 External Reset ($\overline{\text{RESET}}$)

9BC.4.6 Watchdog Timer (WDTA) Reset

The watchdog timers can be configured to generate a reset if the overflow time is exceeded. After a watchdog timer reset is asserted, the corresponding watchdog timer reset flags (the RESF.RESF1 and RESFR.RESFR1 bits for WDTA0RES, and the RESF.RESF2 and RESFR.RESFR2 bits for WDTA1RES) are set.

After that, bits RESF.RESF1 and RESFR.RESFR1 (or bits RESF.RESF2 and RESFR.RESFR2) are not cleared automatically, even if WDTA0RES (or WDTA1RES) is released.

Bits RESF.RESF1 and RESFR.RESFR1, and bits RESF.RESF2 and RESFR.RESFR2 are cleared as described below.

- WDTA0RES:
  Setting the RESFC.RESFC1 bit to 1 clears the RESF.RESF1 bit.
  Setting the RESFCR.RESFCR1 bit to 1 clears the RESFR.RESFR1 bit.
- WDTA1RES:
  Setting the RESFC.RESFC2 bit to 1 clears the RESF.RESF2 bit.
  Setting the RESFCR.RESFCR2 bit to 1 clears the RESFR.RESFR2 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9BC.4.7 Software Reset

The software reset SWRES can be asserted by setting SWRESA.SWRESA to 1.

SWRES sets the reset flag RESF.RESF0 and the RESFR.RESFR0 bit.

RESF.RESF0 and RESFR.RESFR0 are not cleared automatically. RESF.RESF0 and RESFR.RESFR0 are cleared as described below.

- Setting the RESFC.RESFC0 bit to 1 clears the RESF.RESF0 bit.
  Setting the RESFCR.RESFCR0 bit to 1 clears the RESFR.RESFR0 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9BC.4.8 Clock Monitor (CLMA) Reset

The clock monitors can generate the following resets:

- $\overline{\text{CLMA0RES}}$, if a frequency abnormality in HS IntOSC is detected
- $\overline{\text{CLMA1RES}}$, if a frequency abnormality in MainOSC is detected
- $\overline{\text{CLMA2RES}}$, if a frequency abnormality in PLL0 is detected
- $\overline{\text{CLMA3RES}}$, if a frequency abnormality in PLL1 is detected

When the Clock Monitor detects frequency abnormality of the respective clocks, resets $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$ are generated.

In addition, flags $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$, and $\overline{\text{CLMA3RES}}$ (RESF.RESF3, RESFR.RESFR3, RESF.RESF4, RESFR.RESFR4, RESF.RESF5, RESFR.RESFR5, RESF.RESF14, and RESFR.RESFR14) are set.

These flags are not cleared automatically. They are cleared as described below.

- $\overline{\text{CLMA0RES}}$:

  Setting the RESFC.RESFC3 bit to 1 clears the RESF.RESF3 bit.
  Setting the RESFCR.RESFCR3 bit to 1 clears the RESFR.RESFR3 bit.

- $\overline{\text{CLMA1RES}}$:

  Setting the RESFC.RESFC4 bit to 1 clears the RESF.RESF4 bit.
  Setting the RESFCR.RESFCR4 bit to 1 clears the RESFR.RESFR4 bit.

- $\overline{\text{CLMA2RES}}$:

  [RH850/F1KM-S4]
  Setting the RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.
  Setting the RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.

  [RH850/F1KM-S1]
  This is not supported.

- $\overline{\text{CLMA3RES}}$:

  [RH850/F1KM-S4]
  Setting the RESFC.RESFC14 bit to 1 clears the RESF.RESF14 bit.
  Setting the RESFCR.RESFCR14 bit to 1 clears the RESFR.RESFR14 bit.

  [RH850/F1KM-S1]
  Setting the RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.
  Setting the RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.

- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9BC.4.9 Debugger Reset

Debugger reset ($\overline{\text{DBRES}}$) is generated via a debugger command. $\overline{\text{DBRES}}$ activates PURES, and therefore operates in the same way as the power-on clear reset POCRES:

- The clock generators are reset and stop operating. The clock generators should be restarted after release from the reset state.
- The reset factor register RESF and the redundant reset factor register RESFR are cleared (Bits RESF9 and RESFR9 are set to 1 after initialization).

9BC.4.10 Reset Vector Address of CPU

This function is supported in RH850/F1KM-S1.

The default value of reset vector base address (RBASE) of CPU is set to 0000 0000_H at shipment. In Cyclic RUN mode, the reset vector base address is automatically switched to the address specified by the CYCRBASE register. After returning to the RUN mode, the reset vector base address automatically switches back to the default address.

For details about the RBASE register, see **Section 3BC, CPU System of RH850/F1KM**.

Section 10A Power Supply Circuit of RH850/F1KH-D8

This section describes the power supply and power domains of the RH850/F1KH.

10A.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO area) and the Isolated area (ISO area).

The power supply of the Always-On area (AWO area) is always on in all operating modes and stand-by modes.

The power supply of the Isolated area (ISO area) can be turned off to reduce the overall power consumption depending on the type of stand-by mode.

For each power domain, a dedicated on-chip voltage regulator generates the internal supply voltage.

For operation of the device, the following voltages are required:

- Power supply voltages REG0VCC and REG1VCC for the on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in each power domain.
- Power supply voltages EVCC and BVCC for the I/O ports.
- Power supply voltages A0VREF and A1VREF for the A/D converters and the separated I/O ports.

10A.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

Table 10A.1 Power Supply Pins

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	REG0VCC	● On-chip voltage regulators for the Always-On area (AWO area) ● Port group IP0 ● MainOSC ● SubOSC ● POC (REG0VCC voltage detection) ● POC (REG1VCC voltage detection) ● LVI
	AWOVCL*1	
	AWOVSS	
	REG1VCC	● On-chip voltage regulators for the Isolated area (ISO area)
	ISOVCL*1	
	ISOVSS	
Power supply for I/O port	EVCC	● $\overline{\text{RESET}}$ ● FLMD0 (324-pin devices) ● Port groups JP0, P0, P1, P2, P3, P8, P9, P20, P23 (233-pin devices) ● Port groups JP0, P0, P1, P2, P3, P8, P9, P20 (176-pin devices) ● Port groups JP0, P0, P1, P2, P8, P9, P20
	EVSS	
	BVCC	(324-pin devices) ● Port groups P10, P11, P12, P13, P18, P19, P21, P22, P24 (233-pin devices) ● Port groups P10, P11, P12, P13, P18, P19 (176-pin devices) ● Port groups P10, P11, P12, P18
	BVSS	
Power supply for A/D converters	A0VREF	● Analog circuits of ADCA0, port group AP0
	A0VSS	
	A1VREF	● Analog circuits of ADCA1, port group AP1
	A1VSS	

Note: See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for the voltage range of each power supply.

Note 1. Pin to connect a stabilization capacitor for on-chip voltage regulator.

10A.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

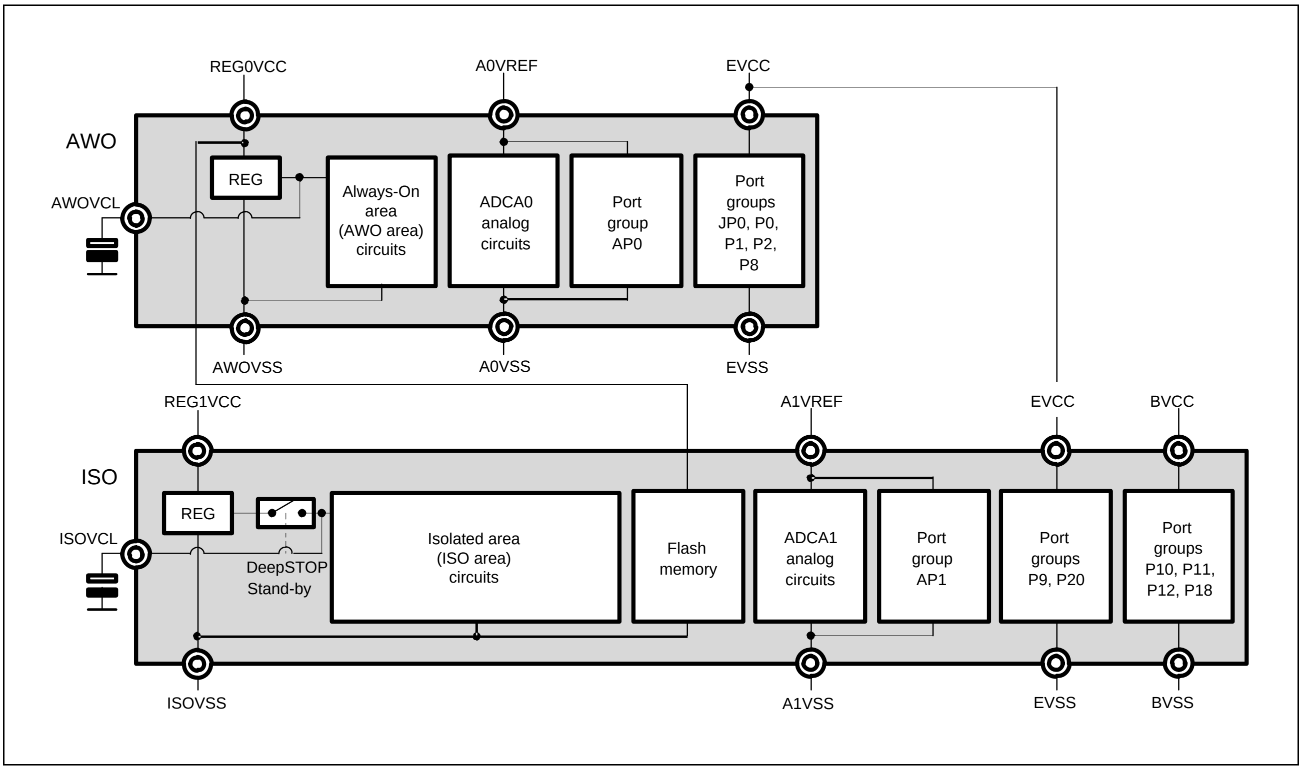


Figure 10A.1 Overview of Power Supply Circuit (176 Pins)

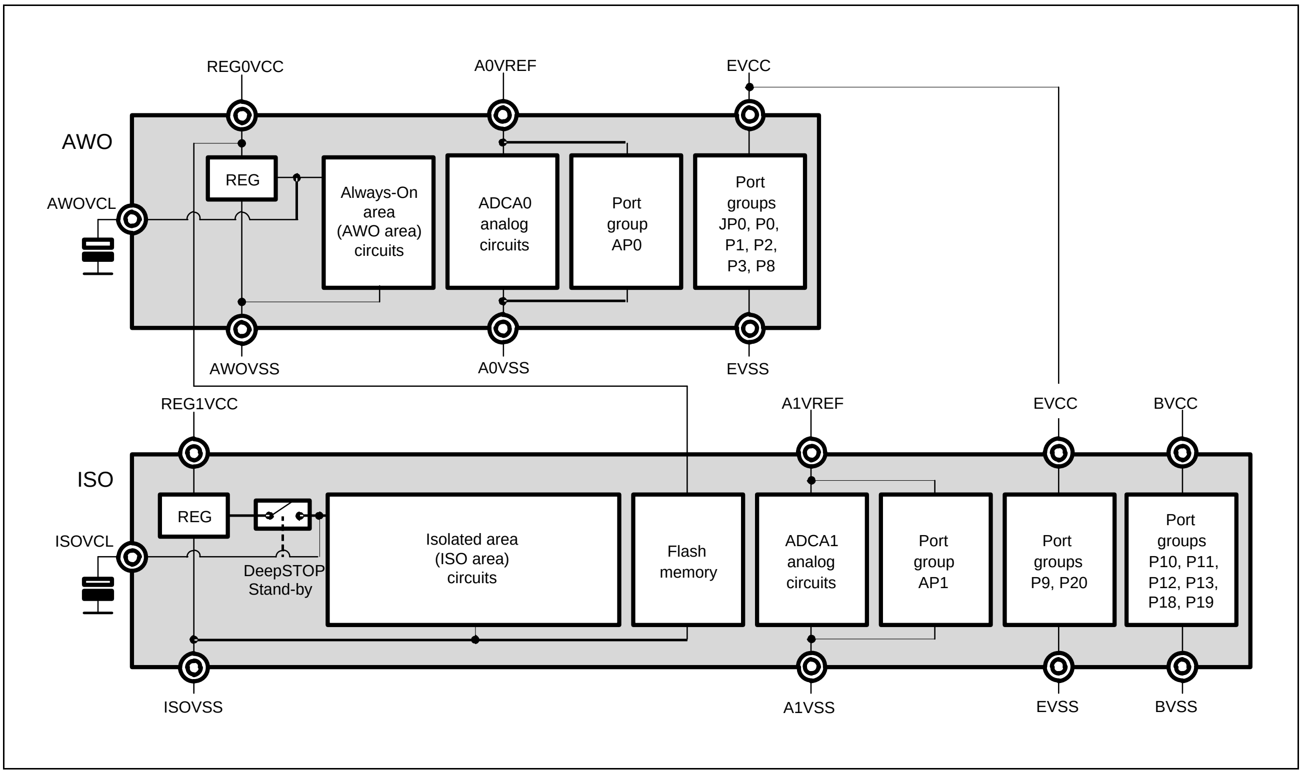


Figure 10A.2 Overview of Power Supply Circuit (233 Pins)

Figure 10A.3 Reserved

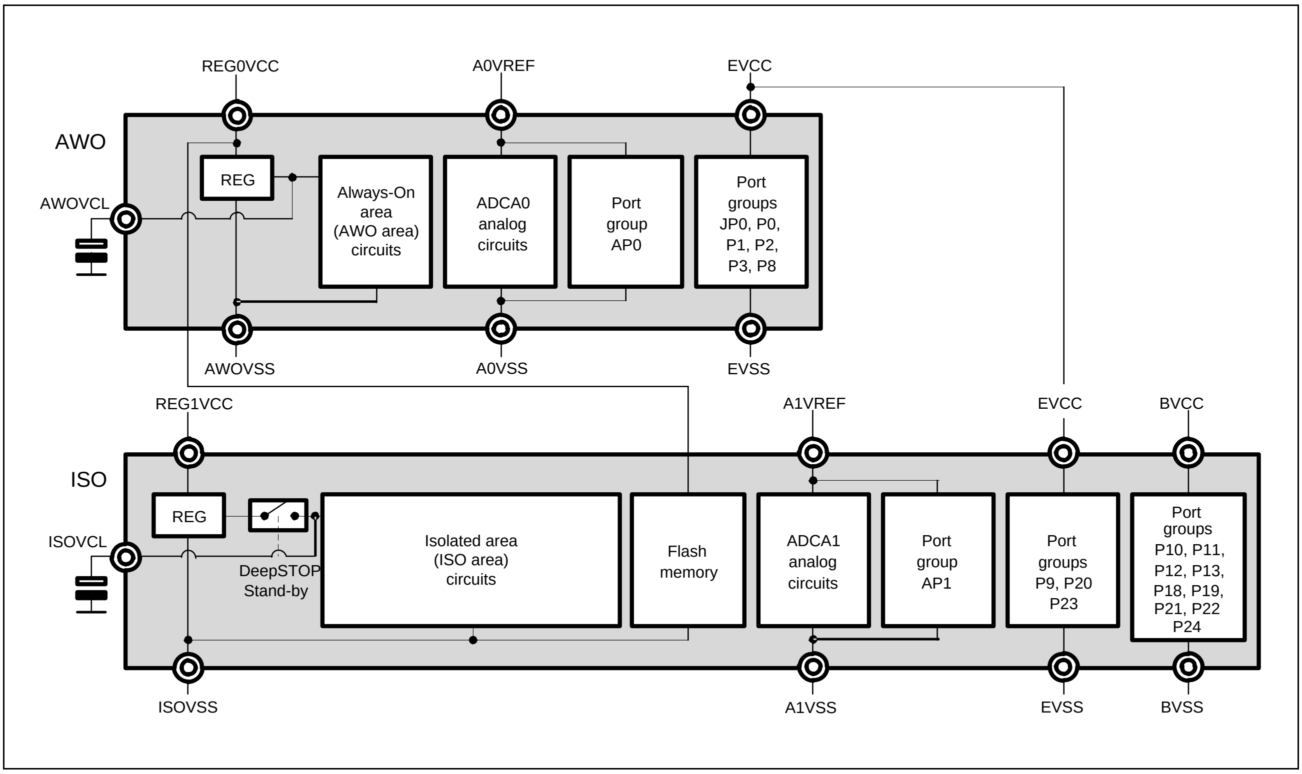


Figure 10A.4 Overview of Power Supply Circuit (324 Pins)

10A.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 10A.2 Functional Modules and Power Domain

Power Domain	Functions
Always-On area (AWO area)	• STBC, Reset controller • Retention RAM • MainOSC, SubOSC, LS IntOSC, HS IntOSC, CLMA0, CLMA1 • WDTA0, RTCAn, TAUJ0, TAUJ2, ADCA0, LPS0 • Port groups JP0, P0, P1, P2, P3, P8, AP0, IP0
Isolated area (ISO area)	• CPU subsystem • Code flash, Data flash, Local RAM, Global RAM • PLL0, PLL1, CLMA2, CLMA3 • WDTA1, WDTA2, DCRAn, TAUDn, TAUBn, TAUJ1, TAUJ3, OSTMn, PWM-Diag, CSIGn, CSIHn, RCFDCn, RLIN24n, RLIN3n, RIICn, ADCA1, Motor Control, ENCAn, KRn, MEMCn, SFMAn, FLXAn, ETNBn, RSENTn, MMCAn • Port groups P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24, AP1

Section 10B Power Supply Circuit of RH850/F1KM-S4

This section describes the power supply and power domains of the RH850/F1KM.

10B.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO area) and the Isolated area (ISO area).

The power supply of the Always-On area (AWO area) is always on in all operating modes and stand-by modes.

The power supply of the Isolated area (ISO area) can be turned off to reduce the overall power consumption depending on the type of stand-by mode.

For each power domain, a dedicated on-chip voltage regulator generates the internal supply voltage.

For operation of the device, the following voltages are required:

- Power supply voltage REGVCC for the on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in each power domain.
- Power supply voltages EVCC and BVCC*[1] for the I/O ports.
- Power supply voltages A0VREF and A1VREF*[1] for the A/D converters and the separated I/O ports.

Note 1. Not supported for the 100-pin devices.

10B.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

Table 10B.1 Power Supply Pins

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	REGVCC	• On-chip voltage regulators for the Always-On area (AWO area) and Isolated area (ISO area) • Port group IP0*2 • MainOSC • SubOSC*2 • POC / LVI
	AWOVCL*1	
	AWOVSS	
	ISOVCL*1	
	ISOVSS	
Power supply for I/O port	EVCC	• $\overline{\text{RESET}}$ • FLMD0 (272-pin devices) • Port groups JP0, P0, P1, P2, P3, P8, P9, P20 (233-pin devices) • Port groups JP0, P0, P1, P2, P3, P8, P9, P20 (176-pin devices) • Port groups JP0, P0, P1, P2, P8, P9, P20 (144-pin devices) • Port groups JP0, P0, P1, P8, P9, P20 (100-pin devices) • Port groups JP0, P0, P8, P9, P10, P11
	EVSS	
	BVCC	(272-pin devices) • Port groups P10, P11, P12, P13, P18, P19, P21, P22 (233-pin devices) • Port groups P10, P11, P12, P13, P18, P19 (176-pin devices) • Port groups P10, P11, P12, P18 (144-pin devices) • Port groups P10, P11, P12, P18
	BVSS	
Power supply for A/D converters	A0VREF	• Analog circuits of ADCA0, port group AP0
	A0VSS	
	A1VREF	• Analog circuits of ADCA1, port group AP1*2
	A1VSS	

Note: See **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for the voltage range of each power supply.

Note 1. Pin to connect a stabilization capacitor for on-chip voltage regulator.

Note 2. Not supported for the 100-pin devices.

10B.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

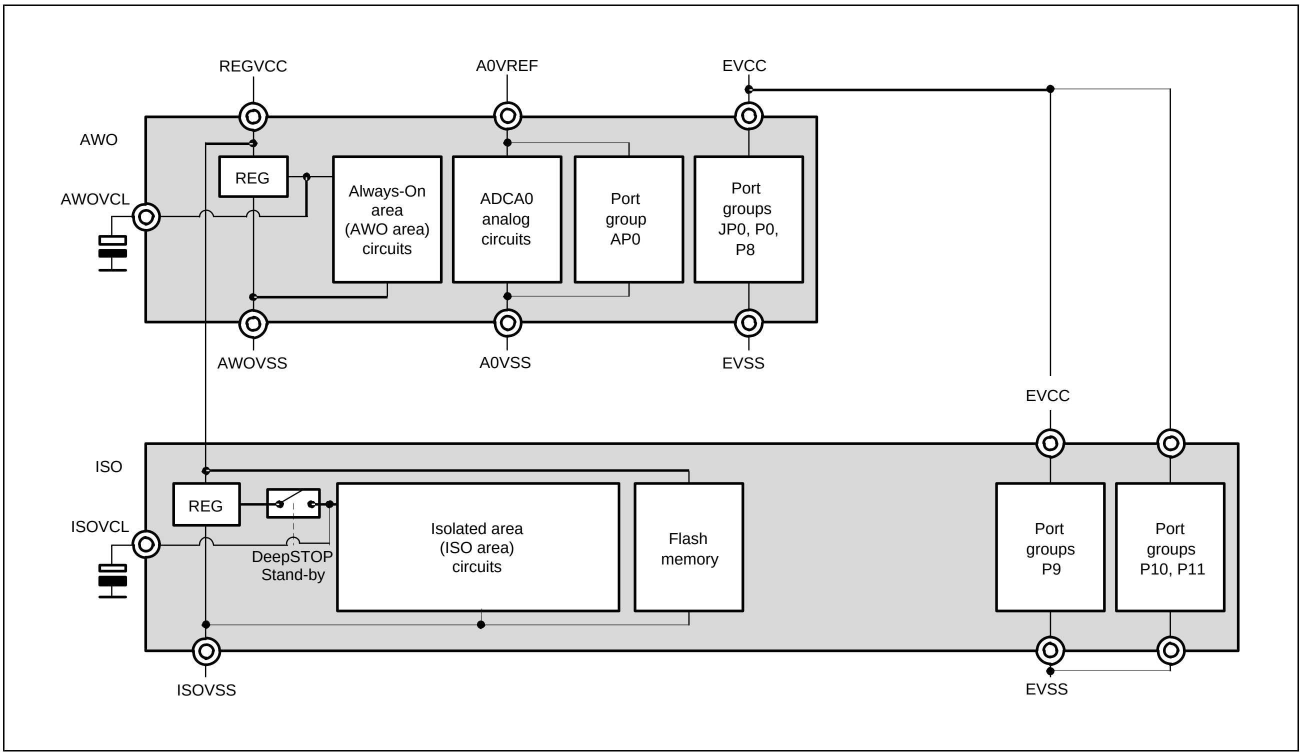


Figure 10B.1 Overview of Power Supply Circuit (100 Pins)

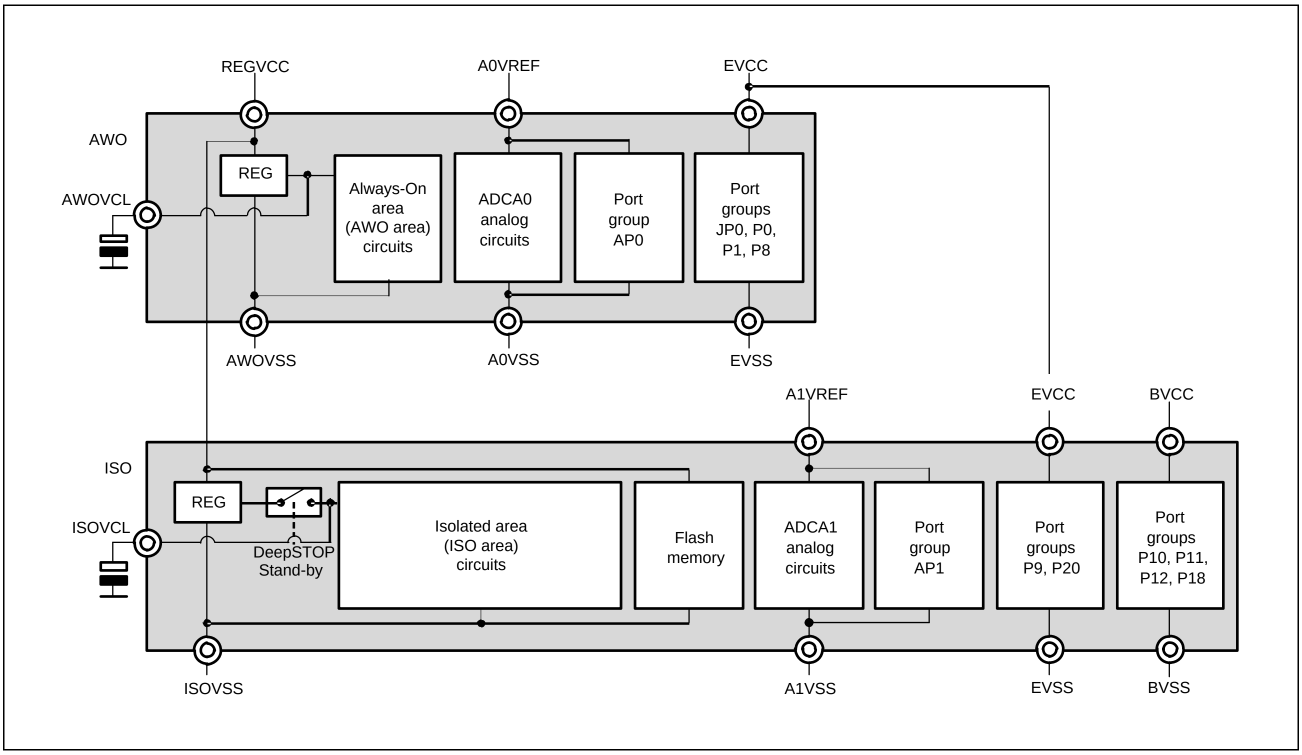


Figure 10B.2 Overview of Power Supply Circuit (144 Pins)

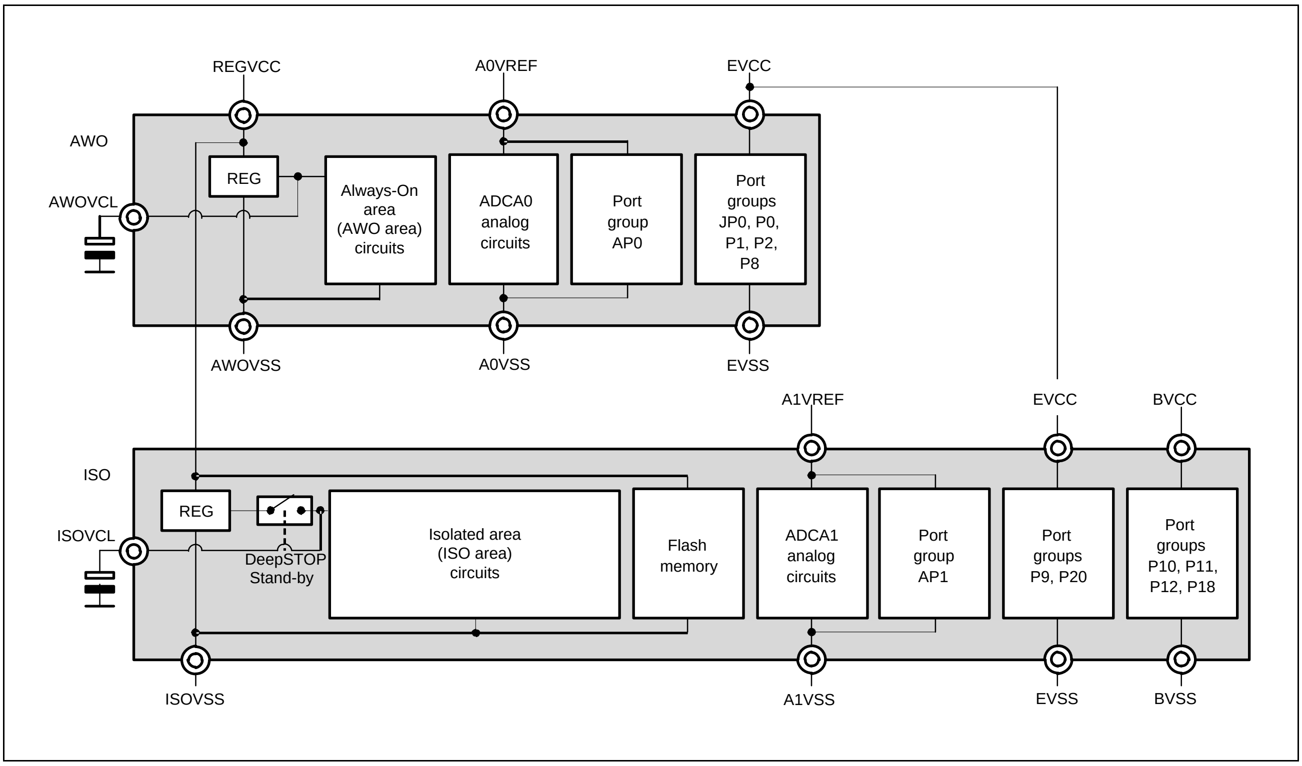


Figure 10B.3 Overview of Power Supply Circuit (176 Pins)

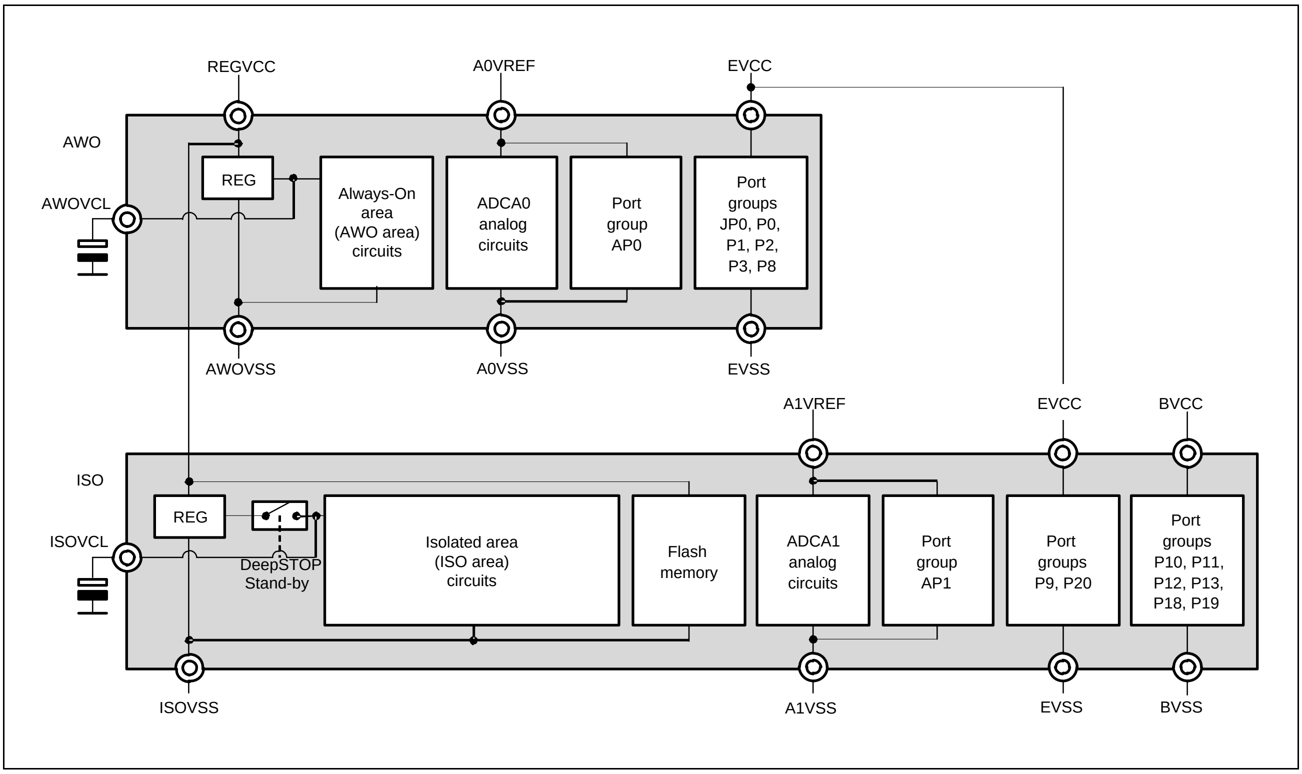


Figure 10B.4 Overview of Power Supply Circuit (233 Pins)

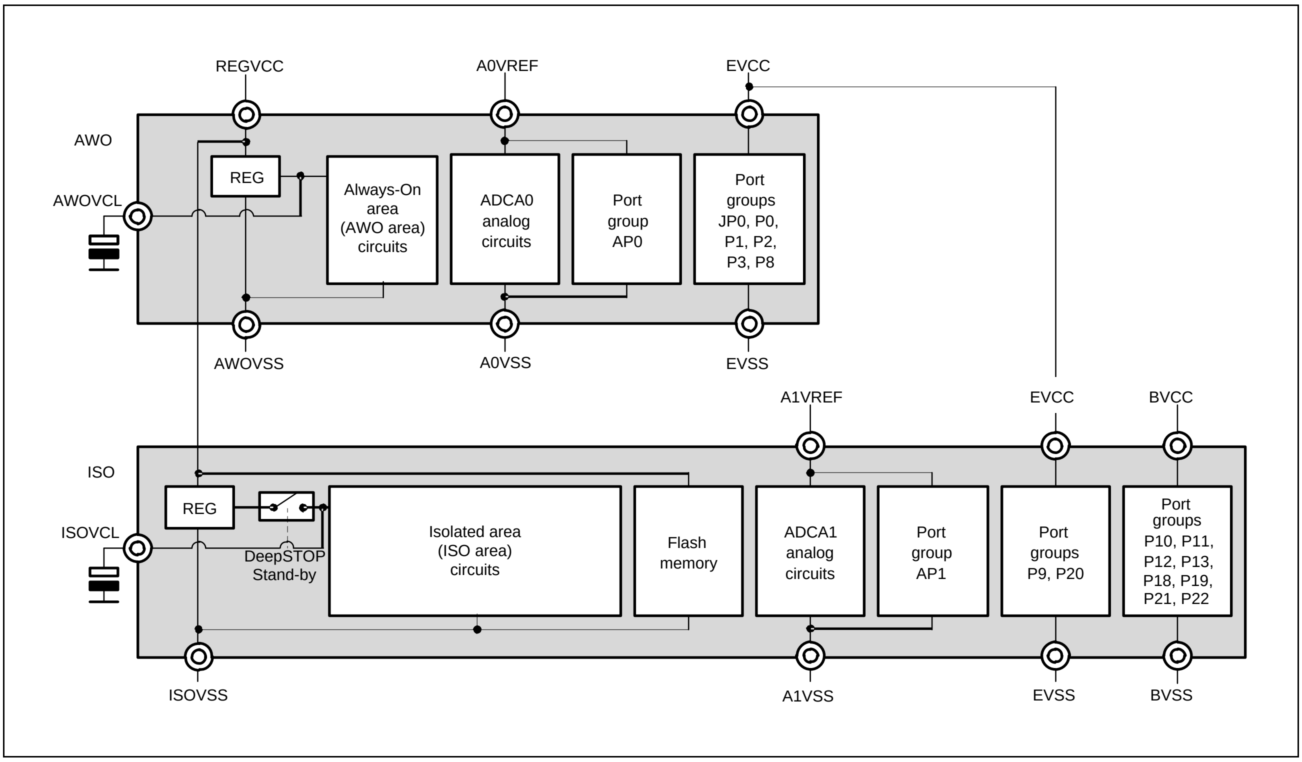


Figure 10B.5 Overview of Power Supply Circuit (272 Pins)

10B.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 10B.2 Functional Modules and Power Domain

Power Domain	Functions
Always-On area (AWO area)	• STBC, Reset controller • Retention RAM • MainOSC, SubOSC, LS IntOSC, HS IntOSC, CLMA0, CLMA1 • WDTA0, RTCAn, TAUJ0, TAUJ2, ADCA0, LPS0 • Port groups JP0, P0, P1, P2, P3, P8, AP0, IP0
Isolated area (ISO area)	• CPU subsystem • Code flash, Data flash, Local RAM, Global RAM • PLL0, PLL1, CLMA2, CLMA3 • WDTA1, DCRAn, TAUDn, TAUBn, TAUJ1, TAUJ3, OSTMn, PWM-Diag, CSIGn, CSIHn, RCFDCn, RLIN24n, RLIN3n, RIICn, ADCA1, Motor Control, ENCAn, KRn, MEMCn, SFMAn, FLXAn, ETNBn, RSENTn • Port groups P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, AP1

Section 10C Power Supply Circuit of RH850/F1KM-S1

This section describes the power supply and power domains of the RH850/F1KM.

10C.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO area) and the Isolated area (ISO area).

The power supply of the Always-On area (AWO area) is always on in all operating modes and stand-by modes.

The power supply of the Isolated area (ISO area) can be turned off to reduce the overall power consumption depending on the type of stand-by mode.

For each power domain, a dedicated on-chip voltage regulator generates the internal supply voltage.

For operation of the device, the following voltages are required:

- Power supply voltage REGVCC for the on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in each power domain.
- Power supply voltage EVCC for the I/O ports.
- Power supply voltage A0VREF for the A/D converters and the separated I/O ports.

10C.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

Table 10C.1 Power Supply Pins

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	REGVCC	• On-chip voltage regulators for the Always-On area (AWO area) and Isolated area (ISO area) • MainOSC • POC / LVI
	AWOVCL*[1]	
	AWOVSS	
	ISOVCL*[1]	
	ISOVSS	
Power supply for I/O port	EVCC	• $\overline{\text{RESET}}$ • FLMD0
	EVSS	(100/80-pin devices) • Port groups JP0, P0, P8, P9, P10, P11 (64/48-pin devices) • Port groups JP0, P0, P8, P9, P10
Power supply for A/D converters	A0VREF	• Analog circuits of ADCA0, port group AP0
	A0VSS	

Note: See **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for the voltage range of each power supply.

Note 1. Pin to connect a stabilization capacitor for on-chip voltage regulator.

10C.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

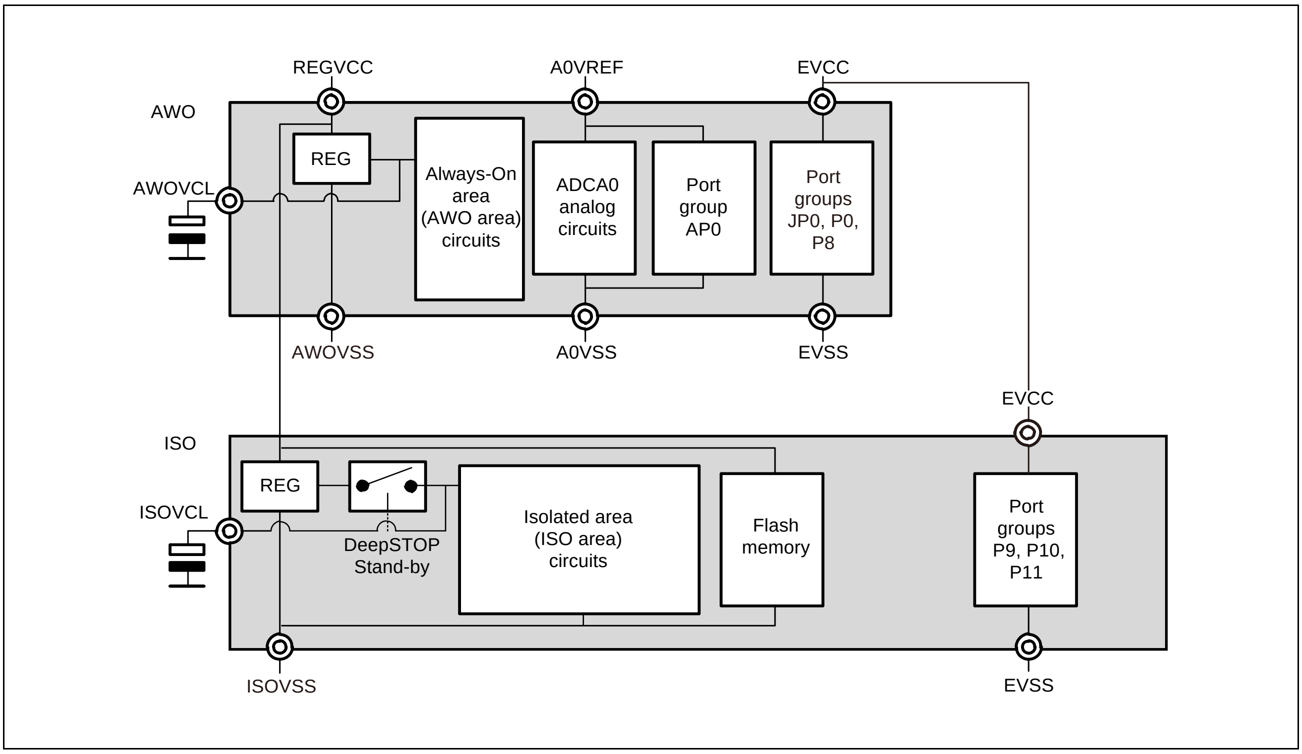


Figure 10C.1 Overview of Power Supply Circuit (100/80 Pins)

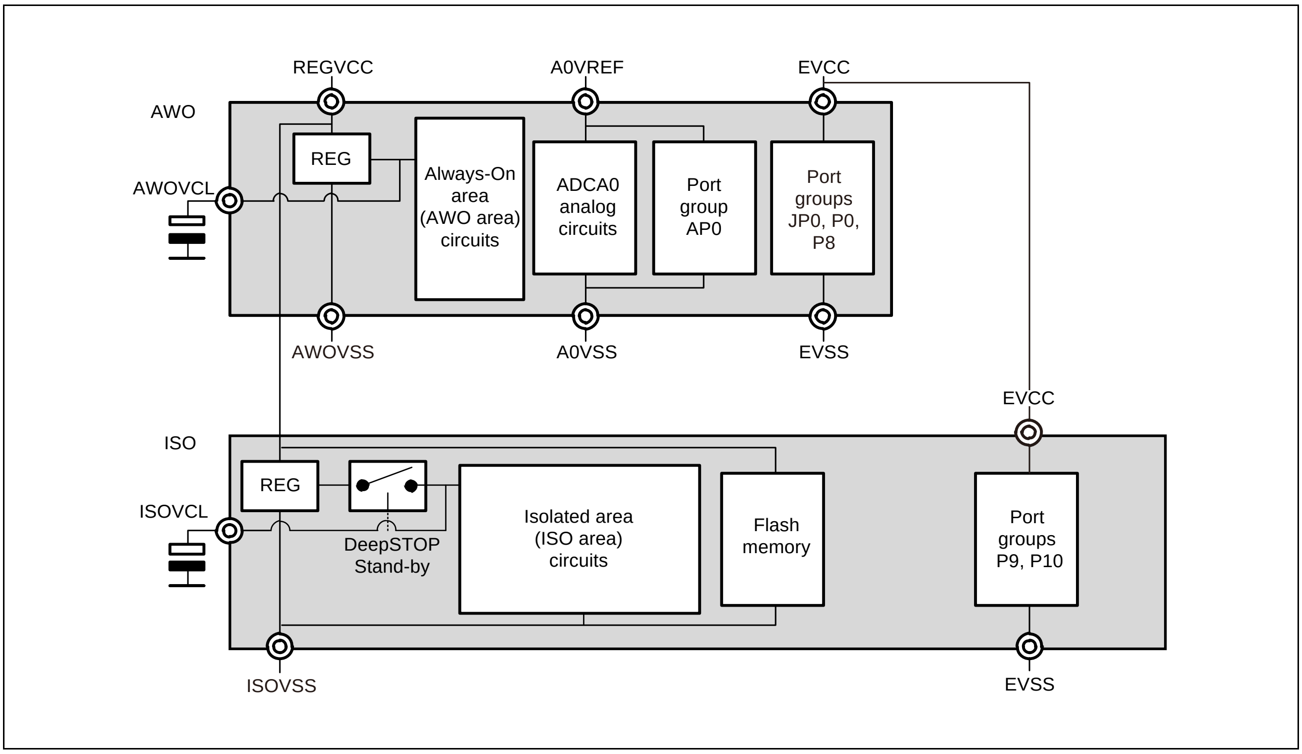


Figure 10C.2 Overview of Power Supply Circuit (64/48 Pins)

10C.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 10C.2 Functional Modules and Power Domain

Power Domain	Functions
Always-On area (AWO area)	• STBC, Reset controller • Retention RAM • MainOSC, LS IntOSC, HS IntOSC, CLMA0, CLMA1 • WDTA0, RTCAn, TAUJ0, TAUJ2, ADCA0, LPS0 • Port groups JP0, P0, P8, AP0
Isolated area (ISO area)	• CPU subsystem • Code flash, Data flash, Local RAM • PLL1, CLMA3 • WDTA1, DCRAn, TAUDn, TAUBn, TAUJ1, TAUJ3, OSTMn, PWM-Diag, CSIGn, CSIHn, RCFDCn, RLIN24n, RLIN3n, RIICn, Motor Control, ENCAn, KRn, RSENTn • Port groups P9, P10, P11

Section 11A Supply Voltage Monitor of RH850/F1KH-D8

This section explains in general about the supply voltage monitor.

The first part in this section describes the supply voltage monitor function, and the ensuing sections describe the registers.

This supply voltage monitor is for detecting and control of a power supply failure. However, the supply voltage monitor does not detect all of the possible failures.

Therefore, a power supply monitoring with an external device is required for the following terminals, in case that the customer's system requires an appropriate failure detection and control for Functional Safety.

- REG0VCC
- REG1VCC
- EVCC
- BVCC
- A0VREF
- A1VREF
- AWOVCL
- ISOVCL

The required power supply specification for power supply monitoring with external device is shown at **Section 47A.4.5, Power Management Characteristics**.

11A.1 Overview

11A.1.1 Functional Overview

The supply voltage monitor continuously monitors multiple external and internal supply voltages in order to ensure that the device operates with a supply voltage within the specified range. If the voltage drops below the reference voltage or comparison voltage, an interrupt request signal or internal reset signal is generated. The following table lists the supply voltage monitor functions.

Table 11A.1 Supply Voltage Monitor Functions

Function Name	Monitor Voltage	Signal Generated when Voltage Drops below Certain Level
Power-On Clear (POC)	REG0VCC REG1VCC	Internal reset signal
Low-Voltage Indicator Circuit (LVI)	REG0VCC	Internal reset signal, interrupt request signal
Core Voltage Monitor (CVM)	Isolated area (ISO area) voltage	Internal reset signal
RAM Retention Voltage Indicator (VLVI)	REG0VCC	—

NOTES

1. The RAM Retention Voltage Indicator sets the very-low voltage detection flag (VLVF) when the voltage drops below the RAM retention voltage.
2. When the internal core voltage monitor (CVM) is used for customer's system as the functional safety measure, the voltage of the Always-On area (AWO area) shall be monitored by the external voltage monitor.

11A.1.2 Power-On Clear (POC)

The POC continuously monitors the external power supply voltage REG0VCC and REG1VCC. This ensures that the microcontroller only operates at or above power-on clear detection voltage (VPOC).

If REG0VCC or REG1VCC falls below the POC detection voltage (REG0VCC < VPOC, REG1VCC < VPOC), the internal reset signal (POCRES) is generated.

For details, see **Section 9A.4.2, Power-On Clear (POC) Reset**.

11A.1.3 Low Voltage Indicator Circuit (LVI)

The LVI continuously compares the external power supply voltage REG0VCC with the LVI reference voltage VLVIn.

If REG0VCC falls below the reference voltage (REG0VCC < VLVIn), an internal reset signal or interrupt request signal is generated.

11A.1.3.1 LVI Reference Voltage

The LVI reference voltage VLVIn can be selected from three different levels by LVICNT.LVICNT[1:0].

If LVICNT.LVICNT[1:0] is set to 00_B, the LVI is disabled.

For the specification of the reference voltage level VLVIn, see **Section 11A.2.2.1, LVICNT — LVI Control Register**.

11A.1.3.2 LVI Reset ($\overline{\text{LVIRES}}$)

When the LVI detection voltage is set and LVIRESMK is cleared, if REG0VCC falls below the reference voltage (REG0VCC < VLVIn), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

For the specification of $\overline{\text{LVIRES}}$ generation, see **Section 9A.4.3, Low-Voltage Indicator (LVI) Reset**.

11A.1.3.3 LVI Interrupt (INTLVIL/INTLVIH)

After the LVI detection voltage is set to LVICNT.LVICNT[1:0] and LVICNT.LVIRESMK is set to 1, if REG0VCC falls below the reference voltage (REG0VCC (MIN) < VLVIn), the LVI interrupt INTLVIL is generated.

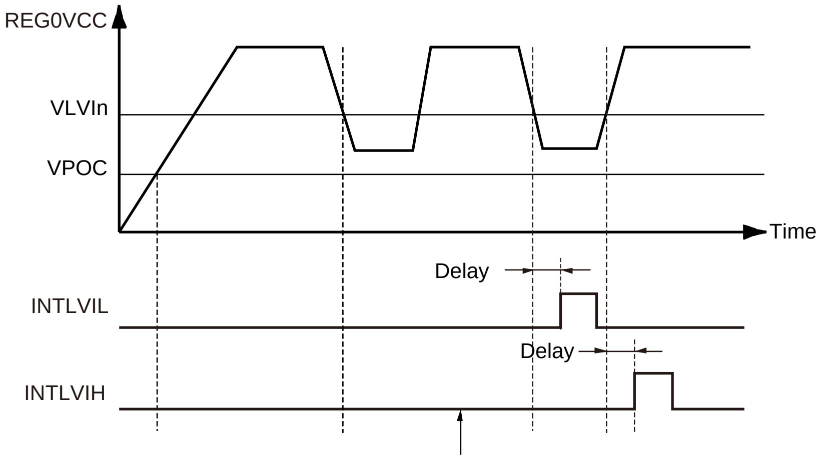
To use the LVI as an interrupt source, the INTLVIL interrupt must be unmasked.

INTLVIL interrupt can be used as wake-up source from all of standby modes. For details, see **Section 14, Stand-By Controller (STBC)**.

After the LVI detection voltage is set to LVICNT.LVICNT[1:0] and LVICNT.LVIRESMK is set to 1, if REG0VCC exceeds the reference voltage (REG0VCC (MIN) > VLVIn), LVI interrupt INTLVIH is generated.

When LVI is used as an interrupt source, INTLVIH interrupt must be unmasked.

The following figure illustrates the timing of INTLVIL/INTLVIH.



Set LVICNT.LVICNT[1:0] to a value other than 00_B.
Set FEINTFMSK.LVILFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0
(For detailed procedure, see **Section 11A.1.3.4, LVI Setting Procedure**.)

Note: Delay:
When REG0VCC falls below VLVIn, INTLVIL is generated after a certain delay.
After that, when REG0VCC exceeds VLVIn, INTLVIH is generated after a certain delay.
See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for delay time.

Figure 11A.1 INTLVIL/INTLVIH Generation Timing

11A.1.3.4 LVI Setting Procedure

The setting procedures for LVI are shown below.

(1) Using LVI as the Reset Source

a. Mask LVI reset. (LVICNT.LVIRESMK = 1)*[1]

b. Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

c. Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*[1]

d. Insert ample wait time by software (see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**).

e. Unmask LVI reset. (LVICNT.LVIRESMK = 0)*[1]

(2) Using LVI as the Interrupt Source (FEINT)

a. Mask LVI reset. (LVICNT.LVIRESMK = 1)*[1]

b. Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

c. Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*[1]

d. Insert ample wait time by software (see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**).

e. Unmask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0)

Note 1. Follow the register protection sequence to set LVICNT register because it is a write-protected register. For details on the write-protected registers, see **Section 5, Write-Protected Registers**.

CAUTION

If REG0VCC is not stable around the LVI detection level (VLVIn), correct judgment of whether INTLVIH or INTLVIL interrupt processing should proceed may not be possible.

For example, if multiple interrupts consisting of both INTLVIH and INTLVIL occur during INTLVIL interrupt processing due to REG0VCC being unstable, the software cannot detect which type of interrupt was generated last.

Consequently, if the last interrupt generated was an INTLVIL interrupt, regardless of REG0VCC (min.) being greater than VLVIn, the software erroneously judges that REG0VCC (min.) < VLVIn.

Accordingly, take measures such as programming the software so that LVI detection interrupt processing is completed before a next LVI interrupt. Also, consider control of REG0VCC.

11A.1.3.5 Clock Supply to the LVI

The clock supply to the LVI is shown in the following table.

Table 11A.2 Clock Supply to the LVI

Unit Name	Unit Clock Name	Supply Clock Name
LVI	Register access clock	CPUCLK_UL

11A.1.4 Core Voltage Monitor (CVM)

The core voltage monitor (CVM) monitors the Isolated area (ISO area) voltage (referred to as “core voltage” below) in the microcontroller.

If the regulator output voltage is outside of the specified range, the internal reset signal ($\overline{\text{CVMRES}}$) is generated.

If the CVM detects an abnormal high voltage, the power supply to the Isolated area (ISO area) is switched off in addition to a reset being generated.

When operation shifts to diagnostic mode (DIAG mode), the CVM enters the abnormal core voltage detection state. An abnormal core voltage detected state can be intentionally created by using the DIAG mode so that the CVM abnormal voltage detected flag can be checked for failures.

CAUTION

The CVM cannot detect drifts in the voltage of the Isolated area (ISO area) caused by the characteristics of the regulator, use conditions or the increase or decrease of voltage.

11A.1.4.1 CVM Reset ($\overline{\text{CVMRES}}$)

If the core voltage exceeds the specified level while high-voltage monitor is enabled (CVMDE.H_D_E = 1), then $\overline{\text{CVMRES}}$ is generated and the power supply to the Isolated area (ISO area) is stopped.

If the core voltage falls below the specified level while low voltage monitor is enabled (CVMDE.L_D_E = 1), $\overline{\text{CVMRES}}$ is generated.

For the specification of $\overline{\text{CVMRES}}$ generation, see **Section 9A.4.4, Core Voltage Monitor (CVM) Reset**.

11A.1.4.2 CVM Setting

Use the option byte to enable the high-voltage monitor and the low-voltage monitor.
For details, see **Section 44.9, Option Bytes**.

11A.1.4.3 Diagnostic (DIAG) Mode

This product supports diagnostic mode.

In diagnostic (DIAG) mode, whether the CVM abnormal voltage detection flag is set to 1 can be checked.

In diagnostic mode, $\overline{\text{CVMRES}}$ is not output.

The setting procedure for diagnostic mode is described below.

Set the registers according to this procedure. Otherwise the operation is not guaranteed.

1. Set CVMDIAG.CVM_DIAG_MASK.*1
2. Set CVMDIAG.CVM_DIAG.*1
3. Wait for 12 µs.*2
4. Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 1 (if these bits are 0, the CVM does not operate normally, requiring error handling).
5. Clear CVMDIAG.CVM_DIAG.*1
6. Clear the CVMF register.*1
7. Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 0 (if these bits are 1, go back to step 5 again).
8. Clear CVMDIAG.CVM_DIAG_MASK.*1

Note 1. Follow the register protection sequence to set CVMF and CVMDIAG registers because these are write-protected registers. For details, see **Section 5, Write-Protected Registers**.

Note 2. At least 50 µs must elapse after the following conditions are fulfilled before step (4) is started.

- Release from HALT state
- Release from STOP mode
- Release from the reset state, when a reset other than a reset due to the CVM is generated in RUN mode (include HALT state)
- The CPU clock is switched
- Operation of the MainOSC is started or stopped
- Operation of the PLL is started or stopped

11A.1.4.4 Clock Supply to the CVM

The clock supply to the CVM is shown in the following table.

Table 11A.3 Clock Supply to the CVM

Unit Name	Unit Clock Name	Supply Clock Name
CVM	Register access clock	CPUCLK_UL

11A.1.5 RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit, VLVI)

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage REG0VCC with the RAM retention voltage VVLVI.

See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for the specification of the RAM retention voltage level VVLVI.

11A.1.5.1 Clock Supply to the VLVI

The clock supply to the VLVI is shown in the following table.

Table 11A.4 Clock Supply to the VLVI

Unit Name	Unit Clock Name	Supply Clock Name
VLVI	Register access clock	CPUCLK_UL

11A.1.5.2 Retention RAM Content Retention

If the power supply voltage REG0VCC does not fall below VVLVI, the content of the retention RAM (RRAM) is retained. See **Section 9A.1.1, Reset Sources** for retention during reset.

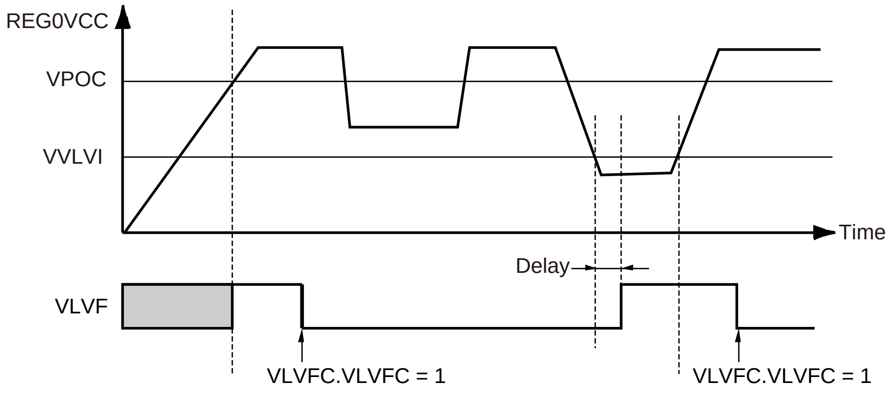
If REG0VCC falls below VVLVI, the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

If REG0VCC falls below the RAM retention voltage (REG0VCC < VVLVI), the VLVF.VLVF bit is set.

After that, even if REG0VCC exceeds VVLVI, the VLVF.VLVF bit is not cleared automatically. It is cleared by

- setting VLVFC.VLVFC bit to 1.

The following figure illustrates the timing of VLVF.



Note: Delay:
When REG0VCC falls below VVLVI, the VLVF bit is set to 1 after a certain delay.
See **Section 47A, Electrical Characteristics of RH850/F1KH-D8** for delay time.

Figure 11A.2 VLVF Operation Timing

11A.1.6 Block Diagram

The block diagram of the supply voltage monitor is shown below.

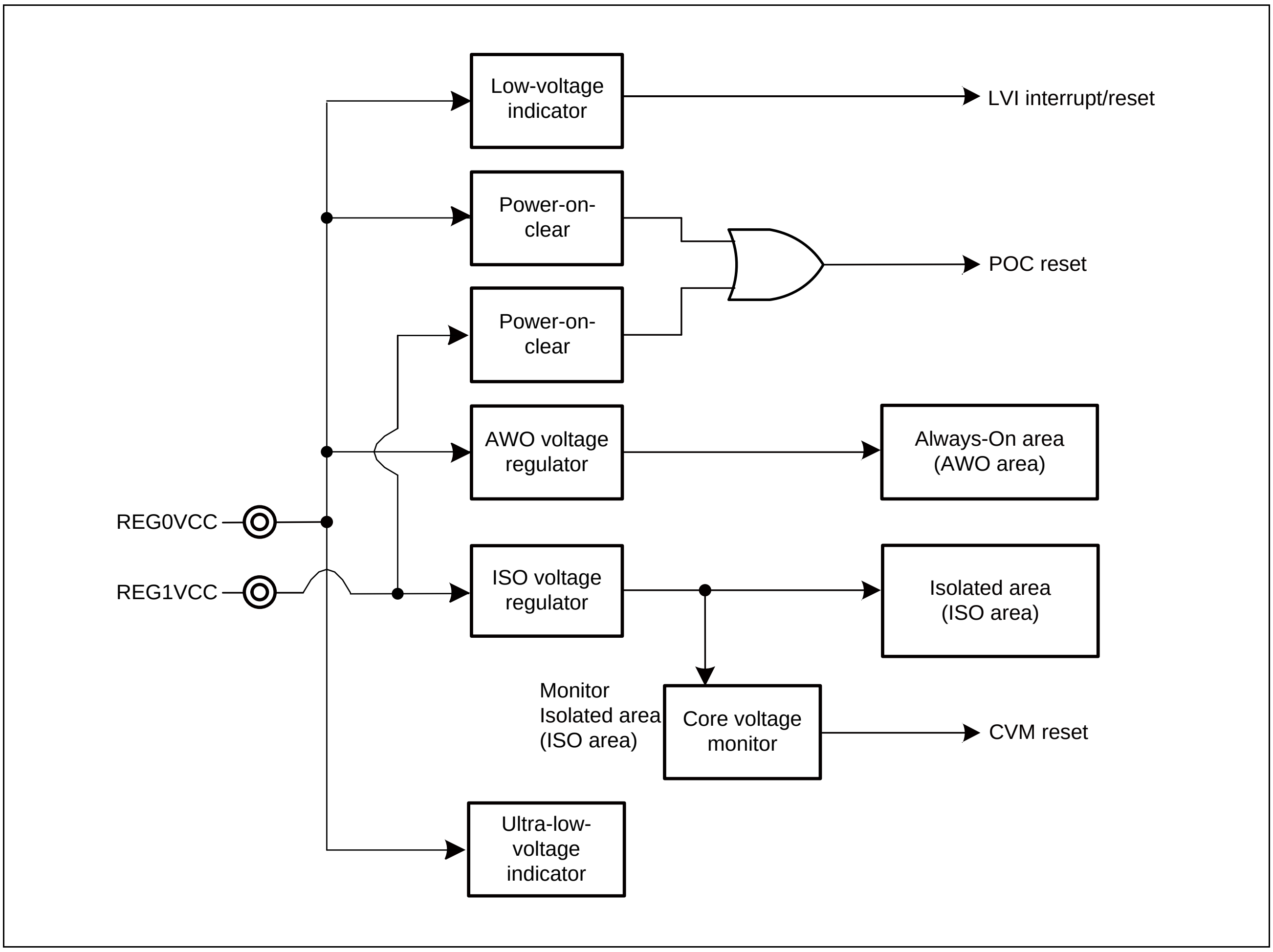


Figure 11A.3 Supply Voltage Monitor

11A.2 Registers

11A.2.1 List of Registers

The following table lists the supply voltage monitor registers.

Table 11A.5 List of Registers

Module Name	Register Name	Symbol	Address
SVM	Low-voltage indicator reset control register		
	LVI control register	LVICNT	FFF8 0A00$_H$
	Core voltage monitor control registers		
	CVM factor register	CVMF	FFF8 3100$_H$
	CVM detection enable register	CVMDE	FFF8 3104$_H$
	CVM diagnostic mode setting register	CVMDIAG	FFF8 3114$_H$
	Very-low-voltage detection control registers		
	Very-low-voltage detection register	VLVF	FFF8 0980$_H$
	Very-low-voltage detection clear register	VLVFC	FFF8 0988$_H$

11A.2.2 Low-Voltage Indicator Reset Control Register

11A.2.2.1 LVICNT — LVI Control Register

This register is used to control the Low-Voltage Indicator and to select the LVI detection level.

This register is initialized by power-up reset PURES.

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 0A00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LVIRESMK	LVICNT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11A.6 LVICNT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LVIRESMK	Mask LVI Reset 0: LVI reset is not masked 1: LVI reset is masked
1, 0	LVICNT[1:0]	Detection Level 0 0: LVI is ignored 0 1: VLVI0 = 4.0 ± 0.1 V (fall), 4.0 ± 0.13 V (rise) 1 0: VLVI1 = 3.7 ± 0.1 V (fall), 3.7 ± 0.13 V (rise) 1 1: VLVI2 = 3.5 ± 0.1 V (fall), 3.5 ± 0.13 V (rise)

NOTE

To use an LVI interrupt, LVI reset must be masked (LVIRESMK = 1) by LVIRESMK.

11A.2.3 Core Voltage Monitor Control Registers

11A.2.3.1 CVMF — CVM Factor Register

This register records the core voltage failure state generated after the last power-on clear reset POCRES.

The L_V_F bit and the H_V_F bit are set to 1 by hardware when the CVM detects core voltage failure.

If the L_V_F or H_V_F bit of this register is set to 1, that bit is not updated until it is initialized by a power-on clear reset POCRES or by writing 0 to the CVMF.L_V_F or CVMF.H_V_F bit. However, it continuously monitors an error signal from the core voltage monitoring circuit in diagnostic mode.

The correct write sequence using the PROTCMDCVM register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3100_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_V_F	L_V_F
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11A.7 CVMF Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	H_V_F	High-Voltage Failure Detection of the Core Voltage by the CVM Read access 0: No high-voltage failure state is detected 1: High-voltage failure state is detected Write access 0: Clear the H_V_F bit 1: Invalid
0	L_V_F	Low-Voltage Failure Detection of the Core Voltage by the CVM Read access 0: No low-voltage failure state is detected 1: Low-voltage failure state is detected Write access 0: Clear the L_V_F bit 1: Invalid

11A.2.3.2 CVMDE — CVM Detection Enable Register

This register is used to indicate the voltage detection enabled or disabled state.

This register is initialized only by power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 3104_H

Value after reset: The value after reset depends on the option byte setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_D_E	L_D_E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1[*1]	0/1[*2]
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The setting of the option byte OPBT0.CVM_HD_EN is reflected.

Note 2. The setting of the option byte OPBT0.CVM_LD_EN is reflected.
For details on the option byte, see **Section 44.9, Option Bytes**.

Table 11A.8 CVMDE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	H_D_E	High-Voltage Monitor Enable 0: High-voltage detection is disabled. 1: High-voltage detection is enabled.
0	L_D_E	Low-Voltage Monitor Enable 0: Low-voltage detection is disabled. 1: Low-voltage detection is enabled.

11A.2.3.3 CVMDIAG — CVM Diagnostic Mode Setting Register

This register sets the CVM diagnostic mode.

This register is initialized only by power-up reset PURES.

For details on the register settings in diagnostic mode, see **Section 11A.1.4.3, Diagnostic (DIAG) Mode**.

The correct write sequence using the PROTCMDCVM register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3114_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVM_ DIAG_ MASK	CVM_ DIAG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11A.9 CVMDIAG Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CVM_DIAG_MASK	$\overline{\text{CVMRES}}$ Mask Control 0: $\overline{\text{CVMRES}}$ is not masked. 1: $\overline{\text{CVMRES}}$ output is masked.
0	CVM_DIAG	CVM Diagnostic Mode Setting 0: Normal mode 1: Diagnostic mode

11A.2.4 Very-Low-Voltage Detection Control Registers

11A.2.4.1 VLVF — Very-Low-Voltage Detection Register

The very-low-voltage detection register (VLVF) shows the state of the RAM retention voltage detection.

This register is set upon detection of a voltage below the RAM retention voltage (VVLVI).

If VLVF is set, the retention RAM content cannot be guaranteed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0980_H

Value after power on: 0000 0001_H
This register is not initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11A.10 VLVF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	VLVF	Very-Low-Voltage Detection Flag 0: Very-low-voltage is not detected. 1: Very-low-voltage is detected. **NOTE:** Very-low-voltage is the voltage status of REG0VCC < RAM retention voltage (VVLVI). For details, See **Section 11A.1.5.2, Retention RAM Content Retention**.

11A.2.4.2 VLVFC — Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0988$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 11A.11 VLVFC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

Section 11BC Supply Voltage Monitor of RH850/F1KM

This section explains in general about the supply voltage monitor.

The first part in this section describes the supply voltage monitor function, and the ensuing sections describe the registers.

This supply voltage monitor is for detecting and control of a power supply failure. However, the supply voltage monitor does not detect all of the possible failures.

Therefore, a power supply monitoring with an external device is required for the following terminals, in case that the customer's system requires an appropriate failure detection and control for Functional Safety.

- REGVCC
- EVCC
- BVCC*1
- A0VREF
- A1VREF*1
- AWOVCL
- ISOVCL

Note 1. This is not supported for the 100-pin devices with RH850/F1KM-S4 and RH850/F1KM-S1.

The required power supply specification for power supply monitoring with external device is shown at **Section 47B.4.5, Power Management Characteristics**, **Section 47C.4.5, Power Management Characteristics**.

11BC.1 Overview

11BC.1.1 Functional Overview

The supply voltage monitor continuously monitors multiple external and internal supply voltages in order to ensure that the device operates with a supply voltage within the specified range. If the voltage drops below the reference voltage or comparison voltage, an interrupt request signal or internal reset signal is generated. The following table lists the supply voltage monitor functions.

Table 11BC.1 Supply Voltage Monitor Functions

Function Name	Monitor Voltage	Signal Generated when Voltage Drops below Certain Level
Power-On Clear (POC)	REGVCC	Internal reset signal
Low-Voltage Indicator Circuit (LVI)	REGVCC	Internal reset signal, interrupt request signal
Core Voltage Monitor (CVM)	Isolated area (ISO area) voltage	Internal reset signal
RAM Retention Voltage Indicator (VLVI)	REGVCC	—

NOTES

1. The RAM Retention Voltage Indicator sets the very-low voltage detection flag (VLVF) when the voltage drops below the RAM retention voltage.
2. When the internal core voltage monitor (CVM) is used for customer's system as the functional safety measure, the voltage of the Always-On area (AWO area) shall be monitored by the external voltage monitor.

11BC.1.2 Power-On Clear (POC)

The POC continuously monitors the external power supply voltage REGVCC. This ensures that the microcontroller only operates at or above power-on clear detection voltage (VPOC).

If REGVCC falls below the POC detection voltage (REGVCC < VPOC), the internal reset signal (POCRES) is generated.

For details, see **Section 9BC.4.2, Power-On Clear (POC) Reset**.

11BC.1.3 Low Voltage Indicator Circuit (LVI)

The LVI continuously compares the external power supply voltage REGVCC with the LVI reference voltage VLVIn.

If REGVCC falls below the reference voltage (REGVCC < VLVIn), an internal reset signal or interrupt request signal is generated.

11BC.1.3.1 LVI Reference Voltage

The LVI reference voltage VLVIn can be selected from three different levels by LVICNT.LVICNT[1:0].

If LVICNT.LVICNT[1:0] is set to 00_B, the LVI is disabled.

For the specification of the reference voltage level VLVIn, see **Section 11BC.2.2.1, LVICNT — LVI Control Register**.

11BC.1.3.2 LVI Reset ($\overline{\text{LVIRES}}$)

When the LVI detection voltage is set and LVIRESMK is cleared, if REGVCC falls below the reference voltage (REGVCC < VLVIn), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

For the specification of $\overline{\text{LVIRES}}$ generation, see **Section 9BC.4.3, Low-Voltage Indicator (LVI) Reset**.

11BC.1.3.3 LVI Interrupt (INTLVIL/INTLVIH)

After the LVI detection voltage is set to LVICNT.LVICNT[1:0] and LVICNT.LVIRESMK is set to 1, if REGVCC falls below the reference voltage (REGVCC (MIN) < VLVIn), the LVI interrupt INTLVIL is generated.

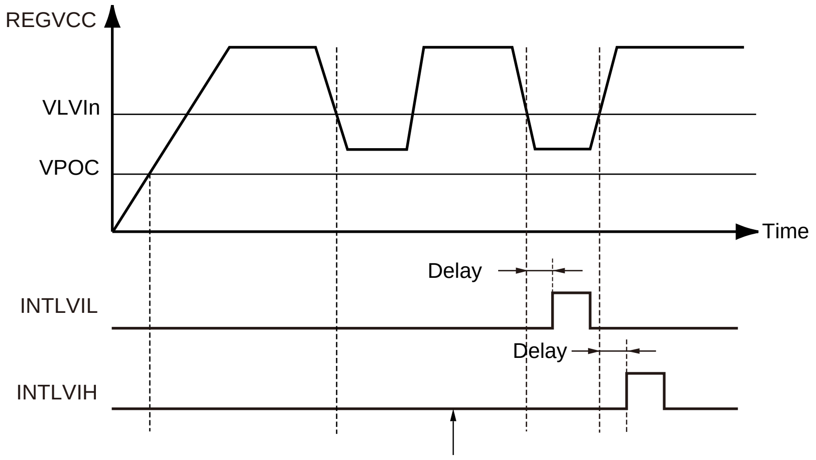
To use the LVI as an interrupt source, the INTLVIL interrupt must be unmasked.

INTLVIL interrupt can be used as wake-up source from all of standby modes. For details, see **Section 14, Stand-By Controller (STBC)**.

After the LVI detection voltage is set to LVICNT.LVICNT[1:0] and LVICNT.LVIRESMK is set to 1, if REGVCC exceeds the reference voltage (REGVCC (MIN) > VLVIn), LVI interrupt INTLVIH is generated.

When LVI is used as an interrupt source, INTLVIH interrupt must be unmasked.

The following figure illustrates the timing of INTLVIL/INTLVIH.



Set LVICNT.LVICNT[1:0] to a value other than 00_B.
Set FEINTFMSK.LVILFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0
(For detailed procedure, see **Section 11BC.1.3.4, LVI Setting Procedure**.)

Note: Delay:
When REGVCC falls below VLVIn, INTLVIL is generated after a certain delay.
After that, when REGVCC exceeds VLVIn, INTLVIH is generated after a certain delay.
See **Section 47B, Electrical Characteristics of RH850/F1KM-S4, Section 47C, Electrical Characteristics of RH850/F1KM-S1** for delay time.

Figure 11BC.1 INTLVIL/INTLVIH Generation Timing

11BC.1.3.4 LVI Setting Procedure

The setting procedures for LVI are shown below.

(1) Using LVI as the Reset Source

a. Mask LVI reset. (LVICNT.LVIRESMK = 1)*1

b. Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

c. Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*1

d. Insert ample wait time by software (see **Section 47B, Electrical Characteristics of RH850/F1KM-S4, Section 47C, Electrical Characteristics of RH850/F1KM-S1**).

e. Unmask LVI reset. (LVICNT.LVIRESMK = 0)*1

(2) Using LVI as the Interrupt Source (FEINT)

a. Mask LVI reset. (LVICNT.LVIRESMK = 1)*1

b. Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

c. Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*1

d. Insert ample wait time by software (see **Section 47B, Electrical Characteristics of RH850/F1KM-S4, Section 47C, Electrical Characteristics of RH850/F1KM-S1**).

e. Unmask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0)

Note 1. Follow the register protection sequence to set LVICNT register because it is a write-protected register. For details on the write-protected registers, see **Section 5, Write-Protected Registers.**

CAUTION

If REGVCC is not stable around the LVI detection level (VLVIn), correct judgment of whether INTLVIH or INTLVIL interrupt processing should proceed may not be possible.

For example, if multiple interrupts consisting of both INTLVIH and INTLVIL occur during INTLVIL interrupt processing due to REGVCC being unstable, the software cannot detect which type of interrupt was generated last.

Consequently, if the last interrupt generated was an INTLVIL interrupt, regardless of REGVCC (min.) being greater than VLVIn, the software erroneously judges that REGVCC (min.) < VLVIn.

Accordingly, take measures such as programming the software so that LVI detection interrupt processing is completed before a next LVI interrupt. Also, consider control of REGVCC.

11BC.1.3.5 Clock Supply to the LVI

The clock supply to the LVI is shown in the following table.

Table 11BC.2 Clock Supply to the LVI (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
LVI	Register access clock	CPUCLK_UL

11BC.1.4 Core Voltage Monitor (CVM)

The core voltage monitor (CVM) monitors the Isolated area (ISO area) voltage (referred to as “core voltage” below) in the microcontroller.

If the regulator output voltage is outside of the specified range, the internal reset signal ($\overline{\text{CVMRES}}$) is generated.

If the CVM detects an abnormal high voltage, the power supply to the Isolated area (ISO area) is switched off in addition to a reset being generated.

When operation shifts to diagnostic mode (DIAG mode), the CVM enters the abnormal core voltage detection state. An abnormal core voltage detected state can be intentionally created by using the DIAG mode so that the CVM abnormal voltage detected flag can be checked for failures.

CAUTION

The CVM cannot detect drifts in the voltage of the Isolated area (ISO area) caused by the characteristics of the regulator, use conditions or the increase or decrease of voltage.

11BC.1.4.1 CVM Reset ($\overline{\text{CVMRES}}$)

If the core voltage exceeds the specified level while high-voltage monitor is enabled (CVMDE.H_D_E = 1), then $\overline{\text{CVMRES}}$ is generated and the power supply to the Isolated area (ISO area) is stopped.

If the core voltage falls below the specified level while low voltage monitor is enabled (CVMDE.L_D_E = 1), $\overline{\text{CVMRES}}$ is generated.

For the specification of $\overline{\text{CVMRES}}$ generation, see **Section 9BC.4.4, Core Voltage Monitor (CVM) Reset**.

11BC.1.4.2 CVM Setting

Use the option byte to enable the high-voltage monitor and the low-voltage monitor.
For details, see **Section 44.9, Option Bytes**.

11BC.1.4.3 Diagnostic (DIAG) Mode

This product supports diagnostic mode.

In diagnostic (DIAG) mode, whether the CVM abnormal voltage detection flag is set to 1 can be checked.

In diagnostic mode, $\overline{\text{CVMRES}}$ is not output.

The setting procedure for diagnostic mode is described below.

Set the registers according to this procedure. Otherwise the operation is not guaranteed.

1. Set CVMDIAG.CVM_DIAG_MASK.*[1]
2. Set CVMDIAG.CVM_DIAG.*[1]
3. Wait for 12 µs.*[2]
4. Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 1 (if these bits are 0, the CVM does not operate normally, requiring error handling).
5. Clear CVMDIAG.CVM_DIAG.*[1]
6. Clear the CVMF register.*[1]
7. Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 0 (if these bits are 1, go back to step 5 again).
8. Clear CVMDIAG.CVM_DIAG_MASK.*[1]

Note 1. Follow the register protection sequence to set CVMF and CVMDIAG registers because these are write-protected registers. For details, see **Section 5, Write-Protected Registers**.

Note 2. At least 50 µs must elapse after the following conditions are fulfilled before step (4) is started.

- Release from HALT state
- Release from STOP mode
- Release from the reset state, when a reset other than a reset due to the CVM is generated in RUN mode (include HALT state)
- The CPU clock is switched
- Operation of the MainOSC is started or stopped
- Operation of the PLL is started or stopped

11BC.1.4.4 Clock Supply to the CVM

The clock supply to the CVM is shown in the following table.

Table 11BC.3 Clock Supply to the CVM (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
CVM	Register access clock	CPUCLK_UL

11BC.1.5 RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit, VLVI)

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage REGVCC with the RAM retention voltage VVLVI.

See **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for the specification of the RAM retention voltage level VVLVI.

11BC.1.5.1 Clock Supply to the VLVI

The clock supply to the VLVI is shown in the following table.

Table 11BC.4 Clock Supply to the VLVI (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
VLVI	Register access clock	CPUCLK_UL

11BC.1.5.2 Retention RAM Content Retention

If the power supply voltage REGVCC does not fall below VVLVI, the content of the retention RAM (RRAM) is retained. See **Section 9BC.1.1, Reset Sources** for retention during reset.

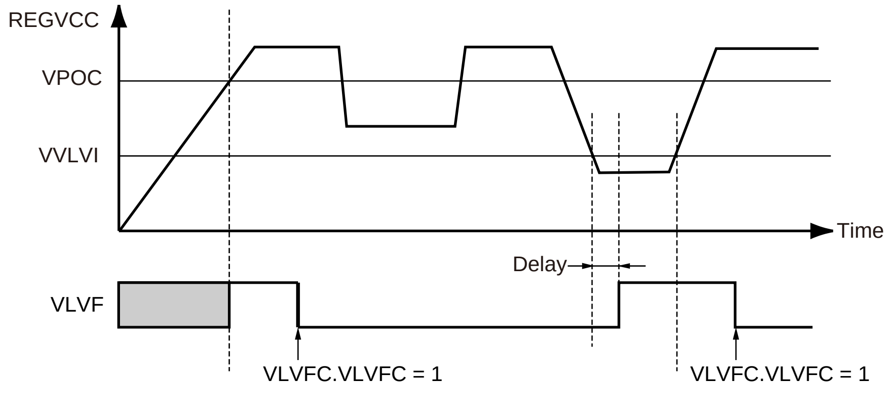
If REGVCC falls below VVLVI, the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

If REGVCC falls below the RAM retention voltage (REGVCC < VVLVI), the VLVF.VLVF bit is set.

After that, even if REGVCC exceeds VVLVI, the VLVF.VLVF bit is not cleared automatically. It is cleared by

- setting VLVFC.VLVFC bit to 1.

The following figure illustrates the timing of VLVF.



Note: Delay:
When REGVCC falls below VVLVI, the VLVF bit is set to 1 after a certain delay.
See **Section 47B, Electrical Characteristics of RH850/F1KM-S4, Section 47C, Electrical Characteristics of RH850/F1KM-S1** for delay time.

Figure 11BC.2 VLVF Operation Timing

11BC.1.6 Block Diagram

The block diagram of the supply voltage monitor is shown below.

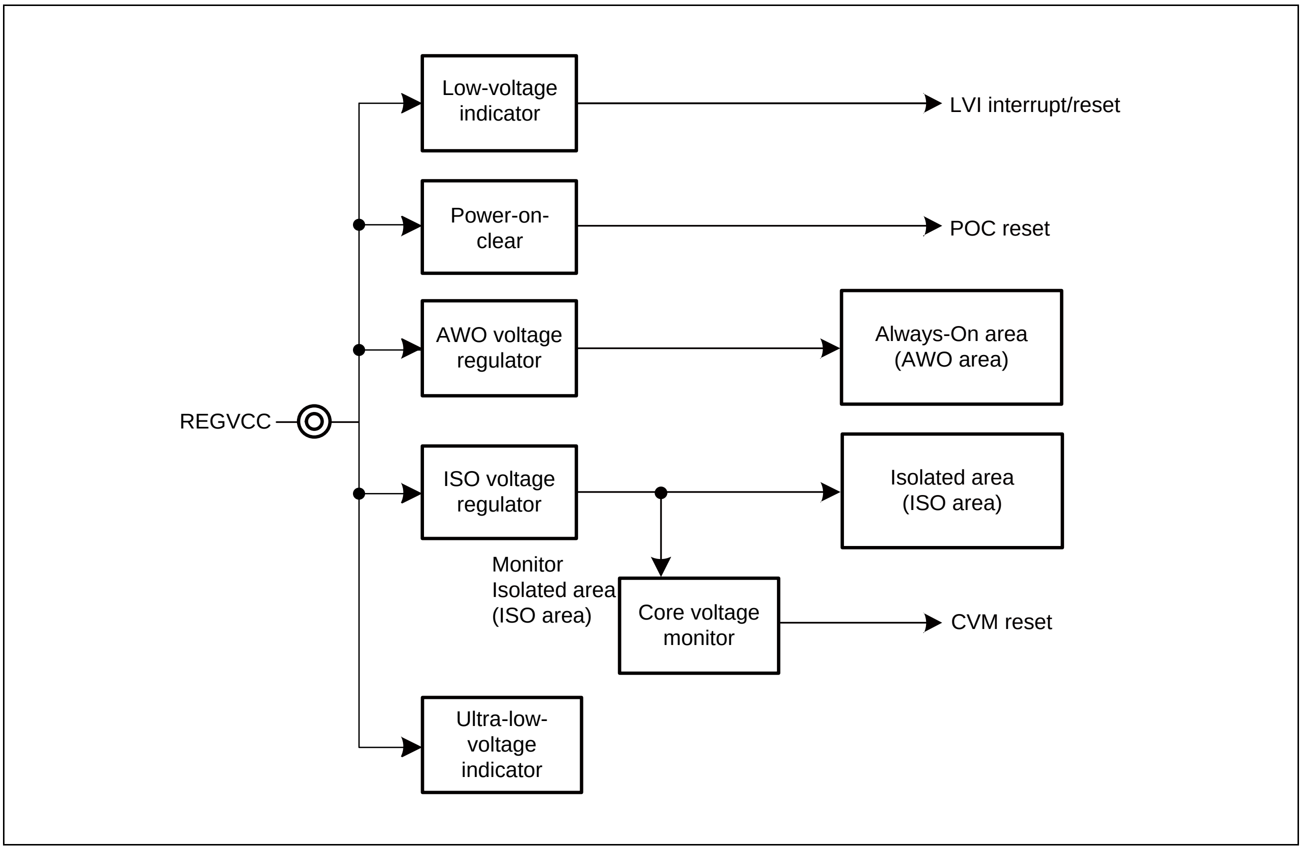


Figure 11BC.3 Supply Voltage Monitor (RH850/F1KM-S4, RH850/F1KM-S1)

11BC.2 Registers

11BC.2.1 List of Registers

The following table lists the supply voltage monitor registers.

Table 11BC.5 List of Registers

Module Name	Register Name	Symbol	Address
SVM	Low-voltage indicator reset control register		
	LVI control register	LVICNT	FFF8 0A00$_H$
	Core voltage monitor control registers		
	CVM factor register	CVMF	FFF8 3100$_H$
	CVM detection enable register	CVMDE	FFF8 3104$_H$
	CVM diagnostic mode setting register	CVMDIAG	FFF8 3114$_H$
	Very-low-voltage detection control registers		
	Very-low-voltage detection register	VLVF	FFF8 0980$_H$
	Very-low-voltage detection clear register	VLVFC	FFF8 0988$_H$

11BC.2.2 Low-Voltage Indicator Reset Control Register

11BC.2.2.1 LVICNT — LVI Control Register

This register is used to control the Low-Voltage Indicator and to select the LVI detection level.

This register is initialized by power-up reset PURES.

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 0A00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LVIRESMK	LVICNT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11BC.6 LVICNT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LVIRESMK	Mask LVI Reset 0: LVI reset is not masked 1: LVI reset is masked
1, 0	LVICNT[1:0]	Detection Level 0 0: LVI is ignored 0 1: VLVI0 = 4.0 ± 0.1 V (fall), 4.0 ± 0.13 V (rise) 1 0: VLVI1 = 3.7 ± 0.1 V (fall), 3.7 ± 0.13 V (rise) 1 1: VLVI2 = 3.5 ± 0.1 V (fall), 3.5 ± 0.13 V (rise)

NOTE

To use an LVI interrupt, LVI reset must be masked (LVIRESMK = 1) by LVIRESMK.

11BC.2.3 Core Voltage Monitor Control Registers

11BC.2.3.1 CVMF — CVM Factor Register

This register records the core voltage failure state generated after the last power-on clear reset POCRES.

The L_V_F bit and the H_V_F bit are set to 1 by hardware when the CVM detects core voltage failure.

If the L_V_F or H_V_F bit of this register is set to 1, that bit is not updated until it is initialized by a power-on clear reset POCRES or by writing 0 to the CVMF.L_V_F or CVMF.H_V_F bit. However, it continuously monitors an error signal from the core voltage monitoring circuit in diagnostic mode.

The correct write sequence using the PROTCMDCVM register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3100_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_V_F	L_V_F
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11BC.7 CVMF Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	H_V_F	High-Voltage Failure Detection of the Core Voltage by the CVM Read access 0: No high-voltage failure state is detected 1: High-voltage failure state is detected Write access 0: Clear the H_V_F bit 1: Invalid
0	L_V_F	Low-Voltage Failure Detection of the Core Voltage by the CVM Read access 0: No low-voltage failure state is detected 1: Low-voltage failure state is detected Write access 0: Clear the L_V_F bit 1: Invalid

11BC.2.3.2 CVMDE — CVM Detection Enable Register

This register is used to indicate the voltage detection enabled or disabled state.

This register is initialized only by power-up reset PURES.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 3104_H
Value after reset:	The value after reset depends on the option byte setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_D_E	L_D_E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1	0/1*2
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The setting of the option byte OPBT0.CVM_HD_EN is reflected.

Note 2. The setting of the option byte OPBT0.CVM_LD_EN is reflected.
For details on the option byte, see **Section 44.9, Option Bytes**.

Table 11BC.8 CVMDE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	H_D_E	High-Voltage Monitor Enable 0: High-voltage detection is disabled. 1: High-voltage detection is enabled.
0	L_D_E	Low-Voltage Monitor Enable 0: Low-voltage detection is disabled. 1: Low-voltage detection is enabled.

11BC.2.3.3 CVMDIAG — CVM Diagnostic Mode Setting Register

This register sets the CVM diagnostic mode.

This register is initialized only by power-up reset PURES.

For details on the register settings in diagnostic mode, see **Section 11BC.1.4.3, Diagnostic (DIAG) Mode**.

The correct write sequence using the PROTCMDCVM register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3114$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVM_DIAG_MASK	CVM_DIAG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11BC.9 CVMDIAG Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CVM_DIAG_MASK	$\overline{\text{CVMRES}}$ Mask Control 0: $\overline{\text{CVMRES}}$ is not masked. 1: $\overline{\text{CVMRES}}$ output is masked.
0	CVM_DIAG	CVM Diagnostic Mode Setting 0: Normal mode 1: Diagnostic mode

11BC.2.4 Very-Low-Voltage Detection Control Registers

11BC.2.4.1 VLVF — Very-Low-Voltage Detection Register

The very-low-voltage detection register (VLVF) shows the state of the RAM retention voltage detection.

This register is set upon detection of a voltage below the RAM retention voltage (VVLVI).

If VLVF is set, the retention RAM content cannot be guaranteed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0980$_H$

Value after power on: 0000 0001$_H$
This register is not initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11BC.10 VLVF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	VLVF	Very-Low-Voltage Detection Flag 0: Very-low-voltage is not detected. 1: Very-low-voltage is detected. **NOTE:** Very-low-voltage is the voltage status of REGVCC < RAM retention voltage (VVLVI). For details, See **Section 11BC.1.5.2, Retention RAM Content Retention**.

11BC.2.4.2 VLVFC — Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0988$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 11BC.11 VLVFC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

Section 12AB Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4

This section explains in general about the clock controller.

The first part in this section describes the specific features of the clock controller of the RH850/F1KH, RH850/F1KM microcontrollers. The ensuing sections describe the clock oscillation circuit, clock selectors, and clock output function that make up the clock controller.

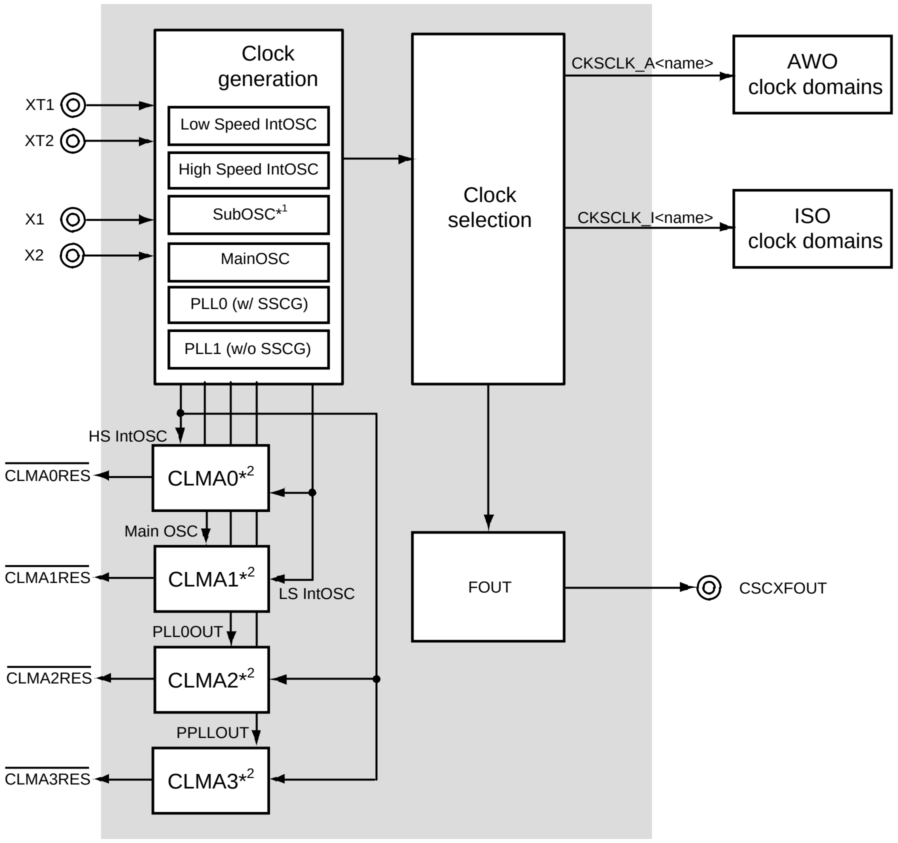
12AB.1 Features of RH850/F1KH, RH850/F1KM Clock Controller

The clock controller of the RH850/F1KH, RH850/F1KM microcontrollers has the following features.

- Six on-chip clock oscillators
  - Main Oscillator (MainOSC) with an oscillation frequency of 8, 16, 20, and 24 MHz
  - Sub Oscillator (SubOSC) with an oscillation frequency of 32.768 kHz*[1]
  - High Speed Internal Oscillator (HS IntOSC) with a nominal frequency of 8 MHz (Typ.)
  - Low Speed Internal Oscillator (LS IntOSC) with a nominal frequency of 240 kHz (Typ.)
  - PLL with SSCG (PLL0)
  - PLL without SSCG (PLL1)
- Fine management of clock supply to peripheral modules through clock domains
- On-chip clock monitor that detects clock anomalies when the Main Oscillator, High Speed Internal Oscillator, or PLL are in use. See **Section 13, Clock Monitor (CLMA)**.
- Clock output (FOUT)

Note 1. It isn’t supported with 100-pin product.

Figure 12AB.1, Clock Controller Overview shows the schematic diagram of the clock controller.



Note 1. It isn't supported with 100-pin product.

Note 2. See **Section 13, Clock Monitor (CLMA)** for details of clock monitors.

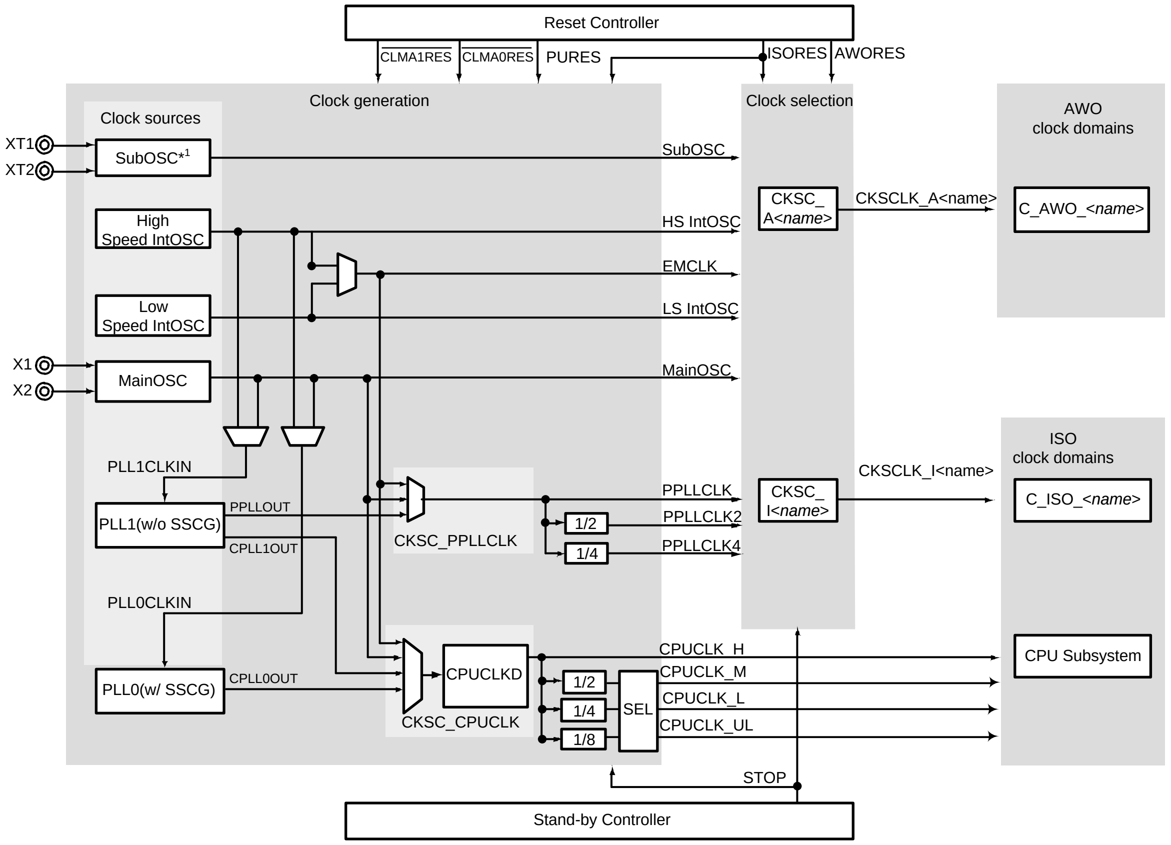
Figure 12AB.1 Clock Controller Overview

12AB.2 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is composed of clock oscillators and clock generation circuits that generate the clocks for the CPU and the peripheral modules, a clock selector for selecting the optimum clock, and clock domains for the CPU and the peripheral modules.

Figure 12AB.2, Clock Controller Configuration shows the configuration of the clock controller.



Note 1. It isn't supported with 100-pin product.

Figure 12AB.2 Clock Controller Configuration

NOTE

Clock domain and clock control register naming conventions

The clock signals and their control registers, etc., described in this section are named according to the following naming conventions to reflect the power domain or clock domain to which they belong. The placeholder “<*name*>” is used to identify the target module in the clock domain:

- Clock domain names:
  - C_AWO_<*name*>: Always-On area (AWO area)*[1] clock domain
  - C_ISO_<*name*>: Isolated area (ISO area)*[1] clock domain
- Domain clock names:
  - CKSCLK_A<*name*>: Always-On area (AWO area) domain clock
  - CKSCLK_I<*name*>: Isolated area (ISO area) domain clock
- Clock selector names:
  - CKSC_A<*name*>: Always-On area (AWO area) clock selector
  - CKSC_I<*name*>: Isolated area (ISO area) clock selector
- Clock selector registers:
  - CKSC_A<*name*>S_CTL: Always-On area (AWO area) source clock selector register
  - CKSC_A<*name*>D_CTL: Always-On area (AWO area) source clock divider register
  - CKSC_I<*name*>S_CTL: Isolated area (ISO area) source clock selector register
  - CKSC_I<*name*>D_CTL: Isolated area (ISO area) source clock divider register

Example

The clock signal CKSCLK_AADCA (placeholder <*name*> = ADCA) is the clock supplied to the clock domain C_AWO_ADCA in the Always-On area (AWO area). This clock is selected by the clock selector register CKSC_AADCAS_CTL.

Note 1. Always-On area (AWO area) and Isolated area (ISO area) refer to the power supply domains. Always-On area (AWO area) is an always-on power supply, and Isolated area (ISO area) is an isolated power supply that is switched on or off by the operation mode.
For details, see **Section 10A, Power Supply Circuit of RH850/F1KH-D8**, **Section 10B, Power Supply Circuit of RH850/F1KM-S4**.

12AB.2.1 Clock Generation Circuits

Six clock oscillators are provided:

Four clock oscillators are located on the Always-On area (AWO area) and two PLL (PLL0 and PLL1) are located on the Isolated area (ISO area).

Main Oscillator (MainOSC)

The MainOSC generates the main clock X.
Generation of the clock X requires the connection of an external resonator to X1 and X2.
The clock X is used as the reference clock for the PLL.

Sub Oscillator (SubOSC)

The SubOSC generates the sub-clock XT, which runs at a frequency of 32.768 kHz (Typ.). Generation of the sub clock XT requires the connection of an external resonator to XT1 and XT2.
This clock is mainly used for real-time clock applications.

High Speed Internal Oscillator (HS IntOSC)

The HS IntOSC generates the clock RH, which runs at a frequency of 8 MHz (Typ.).

Low Speed Internal Oscillator (LS IntOSC)

The LS IntOSC generates the clock RL, which runs at a frequency of 240 kHz (Typ.). It starts operation at power up and cannot be stopped, hence it is always operating.

PLL

The PLL circuits generate high speed operation clocks CPLL0OUT, CPLL1OUT and PPLLOUT for normal operation of the microcontroller.

The clocks supplied by the clock oscillators (X, XT, RH, RL, CPLL0OUT, CPLL1OUT, PPLLOUT) and their divided clocks (CPUCLK_H, CPUCLK_M, CPUCLK_L, CPUCLK_UL, PPLLCLK, PPLLCLK2, and PPLLCLK4) are all generated in the clock generation circuit.

12AB.2.2 Clock Selection

The clocks generated by the clock oscillators are input to the clock selectors CKSC_A<*name*>/CKSC_I<*name*>.

Domain clocks CKSCLK_A<*name*>/CKSCLK_I<*name*> are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

- CKSC_I<*name*>S_CTL/CKSC_I<*name*>D_CTL registers: determine the clock for the Isolated area (ISO area) clock domains.
- CKSC_A<*name*>S_CTL/CKSC_A<*name*>D_CTL registers: determine the clock for the Always-On area (AWO area) clock domains.

Note that not all available clocks generated by the clock oscillators are input to each clock selector.

The following clocks are supplied to the CPU and related modules from the clock generation circuit.

Emergency Clock (EMCLK)

The emergency clock EMCLK is supplied by the

- HS IntOSC, if it is active
- LS IntOSC, if HS IntOSC is inactive

The selection is done automatically after CLMA0 reset is occurred, so if the HS IntOSC becomes lower than the limit for any reason, vital modules of the microcontroller are still in operation, since the LS IntOSC does not stop.

CPU Subsystem Clock (CPUCLK)

The CPU Subsystem clock CPUCLK is derived from PLL0 clock CPLL0OUT, PLL1 clock CPLL1OUT, MainOSC, and EMCLK. The CPU clock selector CKSC_CPUCLK incorporates the selector CPUCLKS, followed by the clock divider CPUCLKD.

The CPUCLK clock divider provides the frequency-divided CPUCLK_H, CPUCLK_M, CPUCLK_L clock signal and CPUCLK_UL clock signal derived from CPUCLK. CPUCLK settings, see **Table 12AB.1, CPUCLK Divide Table**.

Table 12AB.1 CPUCLK Divide Table

	Products of CPU Frequency		
	240 MHz Max.		160 MHz Max.
CKDIVMD*[1]	1	0*[2]	1*[3]
CPUCLK_H	CPUCLK	CPUCLK	CPUCLK
CPUCLK_M	CPUCLK × 1/2	CPUCLK	CPUCLK × 1/2
CPUCLK_L	CPUCLK × 1/4	CPUCLK × 1/2	CPUCLK × 1/4
CPUCLK_UL	CPUCLK × 1/8	CPUCLK × 1/4	CPUCLK × 1/8

Note 1. For details, see **Section 44, Flash Memory**, **44.9.2, OPBT0 — Option Byte 0**

Note 2. When setting “0”, please set CPUCLK to 120 MHz or less.

Note 3. When writing, write “1” for products of CPU frequency 160 MHz max.

12AB.2.3 Clock Domains

The clock controller allows selection of the respective clocks for the CPU and peripheral modules. The clock control scope is called the clock domain. For the correspondence between the CPU and peripheral modules and clock domains, see **Section 12AB.5.3, Clock Domain Settings**.

12AB.2.4 Resetting Clock Oscillators

The clock oscillators on the Always-On area (AWO area) are reset by the PURES signal.

The HS IntOSC is reset when $\overline{\text{CLMA0RES}}$ is generated and the MainOSC is reset when $\overline{\text{CLMA1RES}}$ is generated.

The clock oscillator on the Isolated area (ISO area) is reset by the ISORES signal.

For further details on the clock oscillators, see **Section 12AB.3, Clock Oscillators**.

CAUTION

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4**.

12AB.3 Clock Oscillators

12AB.3.1 Main Oscillator (MainOSC)

The Main Oscillator generates the clock X. X is also used as the PLL0 and PLL1 input clock PLL0CLKIN and PLL1CLKIN.

Figure 12AB.3, Main Oscillator (MainOSC) shows the basic configuration and signals of the MainOSC.

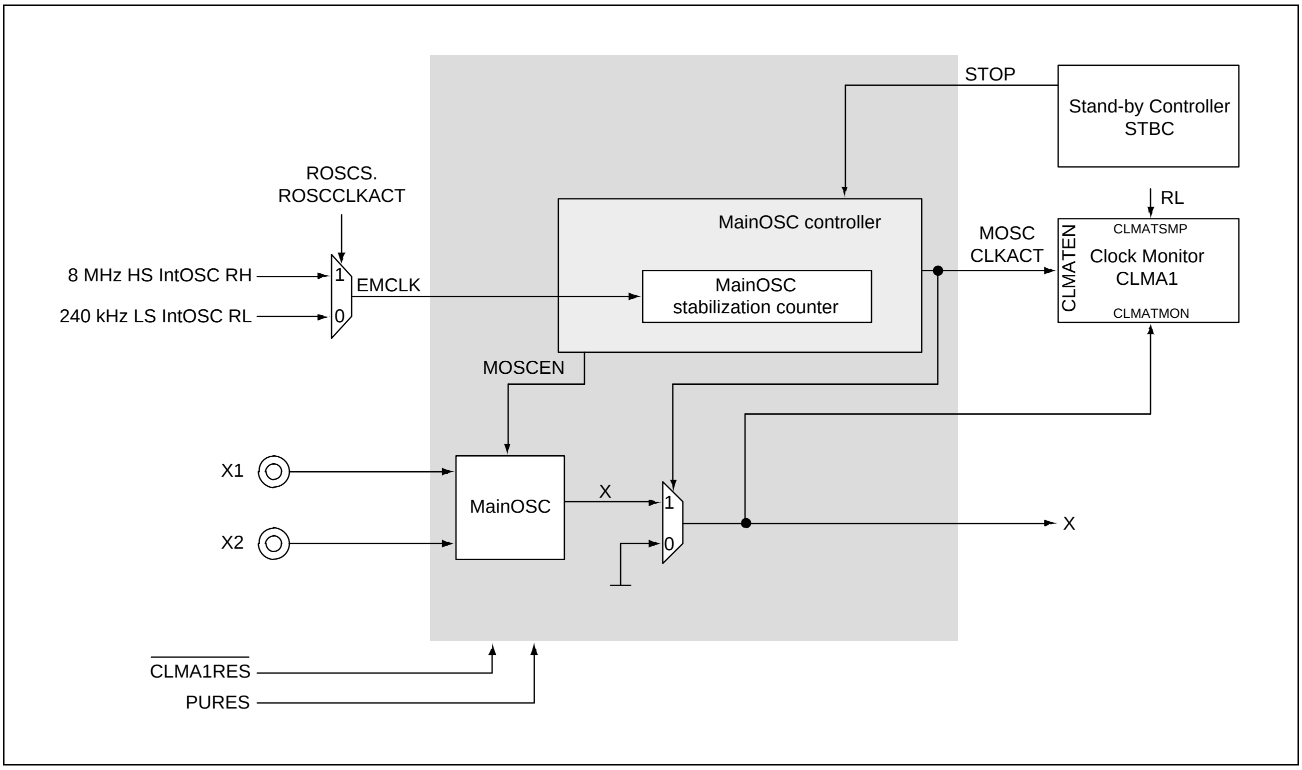


Figure 12AB.3 Main Oscillator (MainOSC)

MainOSC

The MainOSC stops operating after reset is released. To use the MainOSC, set the MainOSC enable trigger bit (MOSCE.MOSCENTRG) to 1 to start the MainOSC.

MainOSC Stabilization

The MOSCST.MOSCCLKST[16:0] bits set the MainOSC oscillation stabilization time.
The MainOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. The oscillation stabilization time can be set to up to $2^{17} – 1$ EMCLK cycles.
As long as the MainOSC is not stable, the MOSCCLKACT signal disables the X output.
When the MainOSC stabilization counter reaches the value specified in MOSCST.MOSCCLKST[16:0], X is assumed to be stable and MOSCCLKACT switches from 0 to 1 to enable output of X when a waveform is output from MainOSC.

Stable and active X clock is indicated by MOSCS.MOSCCLKACT = 1.

MainOSC Amplification Gain

By using MOSCC.MOSCAMPSEL[1:0], the MainOSC’s input frequency, determined by the external resonator, can be selected from 8 MHz, 16 MHz, 20 MHz, and 24 MHz.

MainOSC STOP Requests in Stand-by Mode

The STOP signal from the Stand-by Controller requests the MainOSC Controller to switch off the X clock in stand-by modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).

The stop request mask bit MOSCSTPM.MOSCSTPMSK controls whether the MainOSC is stopped during stand-by or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0:
  The STOP request signal is not masked, so the MainOSC is stopped in stand-by.
  If the MainOSC is in operation before stand-by, it is automatically re-started after wake-up from stand-by, and the MainOSC stabilization counter counts the oscillation stabilization time.
  However, the STOP request is masked under the following conditions, even if MOSCSTPM.MOSCSTPMSK = 0. Therefore, the MainOSC will continue to operate even in stand-by mode.
  - If the stop mask is set (CKSC_xxxx_STPM = 0000 0003_H) for a clock domain for which the MainOSC is selected.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is masked, so the MainOSC continues to operate in stand-by.

Clock Monitor Control

The MainOSC activity signal MOSCCLKACT enables or disables supervision by the Clock Monitor CLMA1. In case the MainOSC is inactive (MOSCCLKACT = 0), supervision of its output clock X by CLMA1 is also disabled.

MainOSC Enable/Disable Trigger

The MainOSC can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger MOSCE.MOSCENTRG = 1 starts the MainOSC.
  Note that setting the enable trigger is only effective if the MainOSC is inactive, i.e. if MOSCS.MOSCCLKACT = 0.
- Disable trigger MOSCE.MOSCDISTRG = 1 stops the MainOSC.
  Note that setting the disable trigger is only effective if the MainOSC is active (MOSCCLKACT = 1) and the MainOSC stop requests are not masked (MOSCSTPM.MOSCSTPMSK = 0).

Direct Clock Input to X1 (EXCLK Mode)

A clock waveform from an external clock source can be supplied to X1 pin. In this case, set the MOSCM bit of MOSCM register to 1 before clock input to X1 pin is supplied.

12AB.3.2 Sub Oscillator (SubOSC)

The Sub Oscillator generates the sub clock XT. XT has usually a frequency of 32.768 kHz and is used for the Real-time Clock.

Figure 12AB.4, Sub Oscillator (SubOSC) shows the basic structure and signals of the SubOSC.

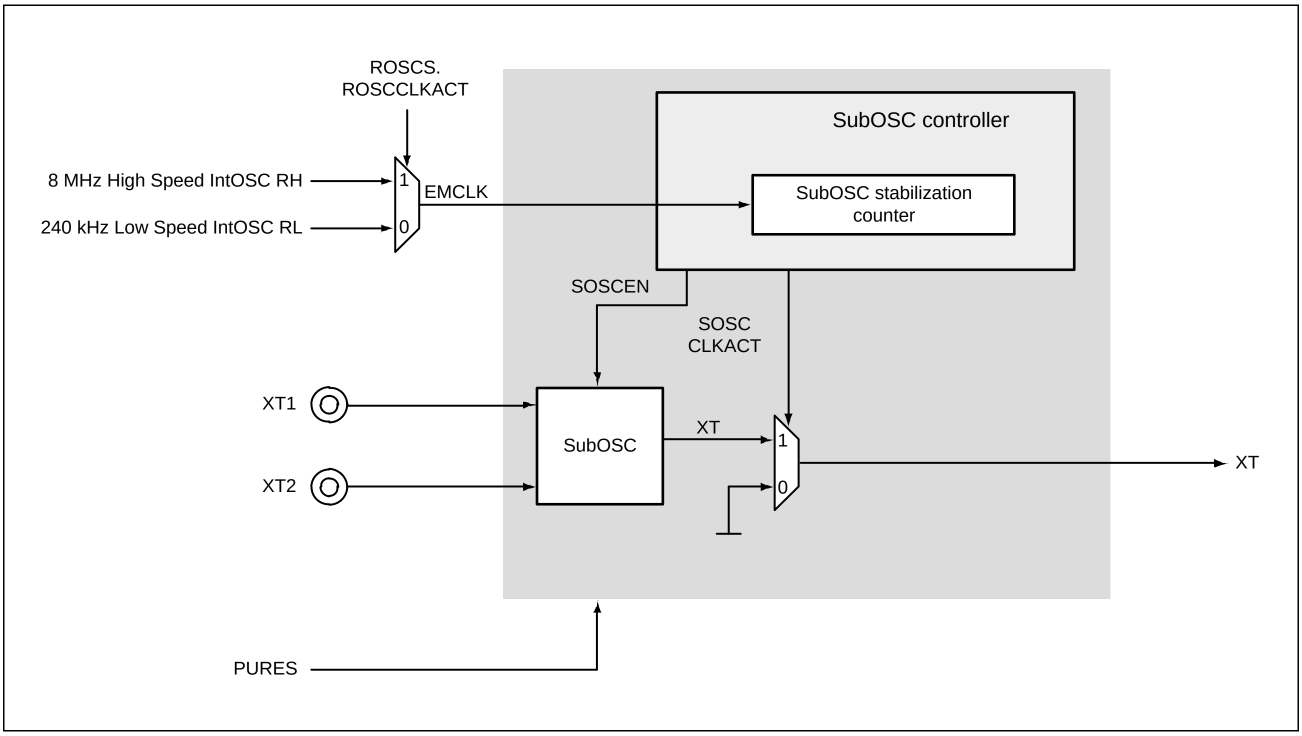


Figure 12AB.4 Sub Oscillator (SubOSC)

SubOSC Enable

The SubOSC stops operating after reset is released. To use the SubOSC, set SubOSC enable trigger bit (SOSCE.SOSCENTRG) to 1 to start the SubOSC.

SubOSC Stabilization

The SOSCST.SOSCCLKST[29:0] bits set the SubOSC oscillation stabilization time.
The SubOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting.
As long as the SubOSC is not stable, the SOSCCLKACT signal disables the XT output.
When the SubOSC stabilization counter reaches the value specified in SOSCST.SOSCCLKST[29:0], XT is assumed to be stable and SOSCCLKACT switches from 0 to 1 to enable output of XT.
Secure the stabilization time longer than 2 seconds.
Stable and active XT clock is indicated by SOSCS.SOSCCLKACT = 1.

SubOSC Input Frequencies

The SubOSC input frequency is 32.768 kHz (Typ.).

SubOSC Enable/Disable Trigger

SubOSC can be enabled or disabled by using enable/disable trigger control bit.

- Enable trigger SOSCE.SOSCENTRG = 1 starts the SubOSC.
  Note that setting the enable trigger is only effective if the SubOSC is inactive, i.e. if SOSCS.SOSCCLKACT = 0.
- Disable trigger SOSCE.SOSCDISTRG = 1 stops the SubOSC. Note that setting the disable trigger is only effective if the SubOSC is active; that is, if SOSCS.SOSCCLKACT = 1.

12AB.3.3 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock RH. RH has a nominal frequency of 8 MHz.

Figure 12AB.5, High Speed Internal Oscillator (HS IntOSC) shows the basic configuration and signals of the HS IntOSC.

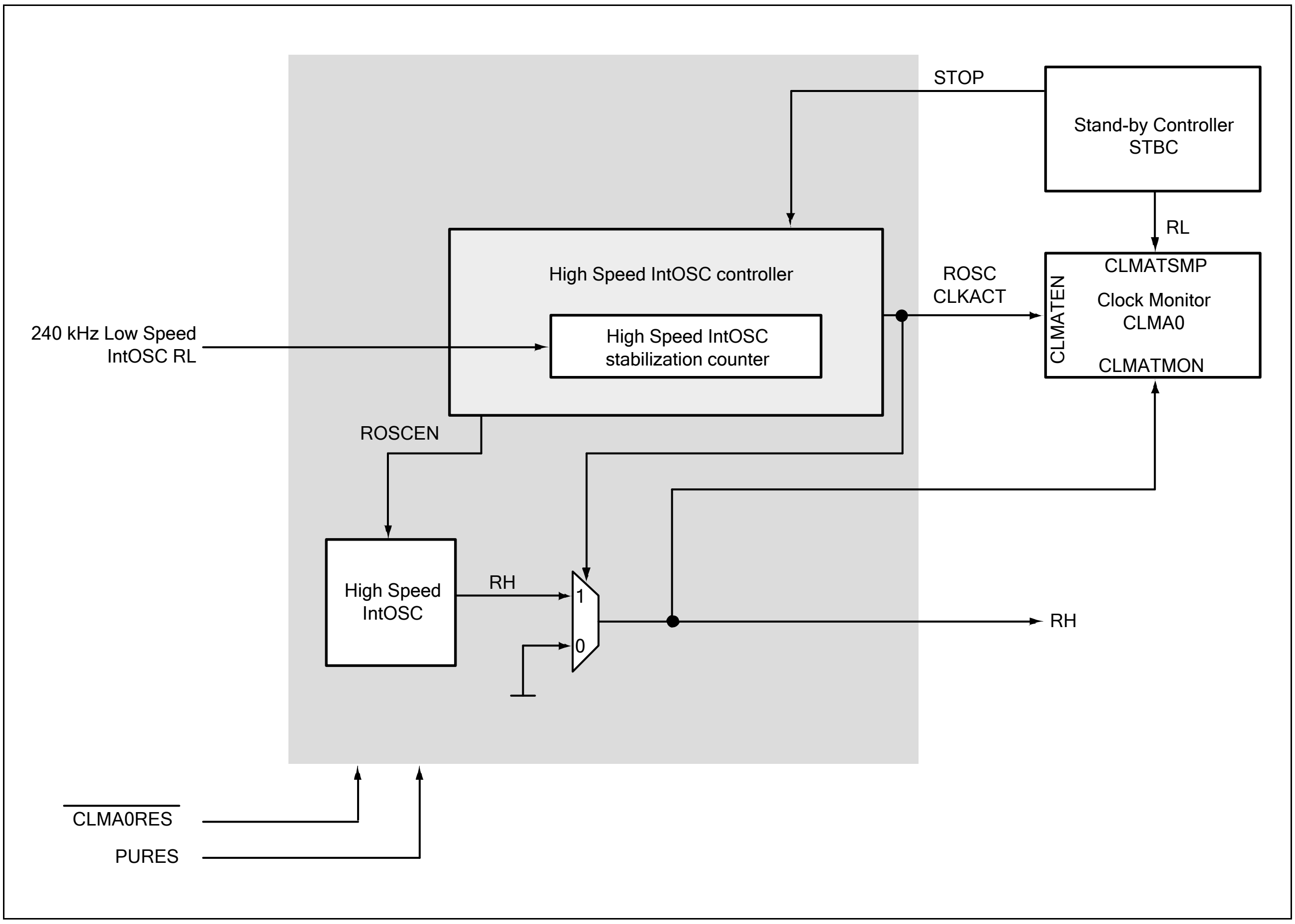


Figure 12AB.5 High Speed Internal Oscillator (HS IntOSC)

After reset release the HS IntOSC starts operation.

NOTE

The HS IntOSC can neither be stopped nor started by software. It can only be stopped in stand-by mode. On the other hand, when CLMA0 is reset, the HS IntOSC can be enabled to stop by software.

HS IntOSC Stabilization

HS IntOSC outputs RH when it is stabilized.
Stable and active RH clock is indicated by ROSCS.ROSCCLKACT = 1.

HS IntOSC STOP Requests in Stand-by Mode

The STOP signal from the Stand-By Controller requests the HS IntOSC Controller to switch off the RH clock in stand-by modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).
The stop request mask bit ROSCSTPM.ROSCSTPMSK controls whether the HS IntOSC is stopped during stand-by or continues operation:

- ROSCSTPM.ROSCSTPMSK = 0:
  The STOP request signal is not masked, so the HS IntOSC is stopped during stand-by and automatically restarted after wake-up from stand-by.
  However, the STOP request is masked under the following conditions, even if ROSCSTPM.ROSCSTPMSK = 0. Therefore, the HS IntOSC will continue to operate even in stand-by mode.
  - If the stop mask is set (CKSC_xxxx_STPM = 0000 0003_H) for a clock domain for which the HS IntOSC is selected.
  - If the low power sampler (LPS) is operating
- ROSCSTPM.ROSCSTPMSK = 1:
  The STOP request signal is masked, so the HS IntOSC continues to operate during stand-by.

Clock Monitor Control

The HS IntOSC activity signal ROSCCLKACT enables or disables supervision by the Clock Monitor CLMA0. In case the HS IntOSC is inactive (ROSCCLKACT = 0), supervision of its output clock by CLMA0 is also deactivated.

The HS IntOSC clock RH is used as the sampling clock for Clock Monitor CLMA2 and CLMA3.

HS IntOSC Disable Trigger

The disable trigger, ROSCE.ROSCDISTRG = 1 stops the HS IntOSC.

The setting of the disable trigger is enabled when HS IntOSC is active (ROSCS.ROSCCLKACT = 1) and HS IntOSC stop requests are not masked (ROSCSTPM.ROSCSTPMSK = 0).

HS IntOSC User Calibration Function

The HS IntOSC user trimming register (ROSCUT) enables adjustment of the HS IntOSC frequency. The initial value of ROSCUT is the preset value of the HS IntOSC trimming data. Overwrite the value with “read value + 1” or “read value – 1” until the HS IntOSC frequency reaches the target frequency range.

12AB.3.4 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock RL. RL has a nominal frequency of 240 kHz.

Figure 12AB.6, Low Speed Internal Oscillator (LS IntOSC) shows the basic configuration and signals of the LS IntOSC.

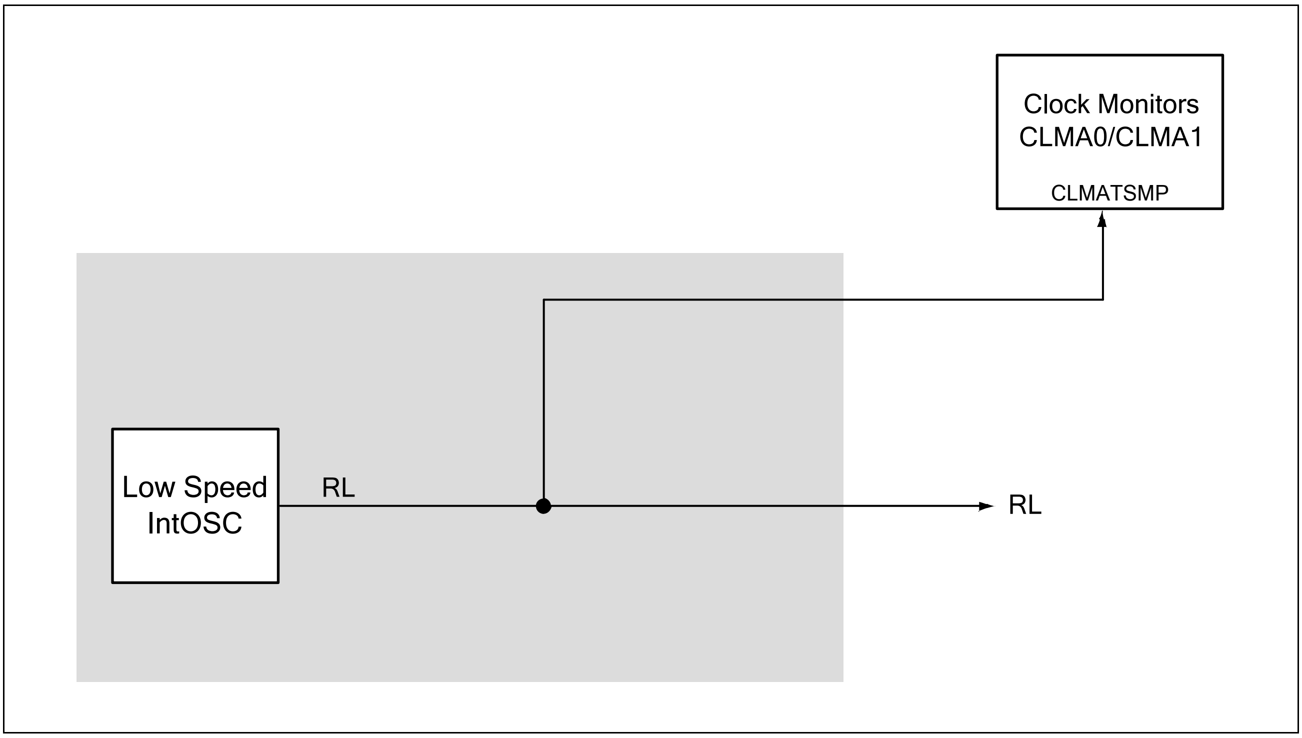


Figure 12AB.6 Low Speed Internal Oscillator (LS IntOSC)

After reset release the LS IntOSC starts operation. It cannot be stopped.

The LS IntOSC clock RL is used as the sampling clock for the Clock Monitors CLMA0 and CLMA1.

12AB.3.5 PLL

MainOSC or HS IntOSC is input to a phase-locked loop (PLL0 and PLL1) clock oscillator as PLL0CLKIN and PLL1CLKIN.
The PLL0 output clock CPLL0OUT, PLL1 output clocks CPLL1OUT and PPLLOUT serve as the main operation clocks for the microcontroller.

Figure 12AB.7, PLL shows the basic configuration and signals of the PLL.

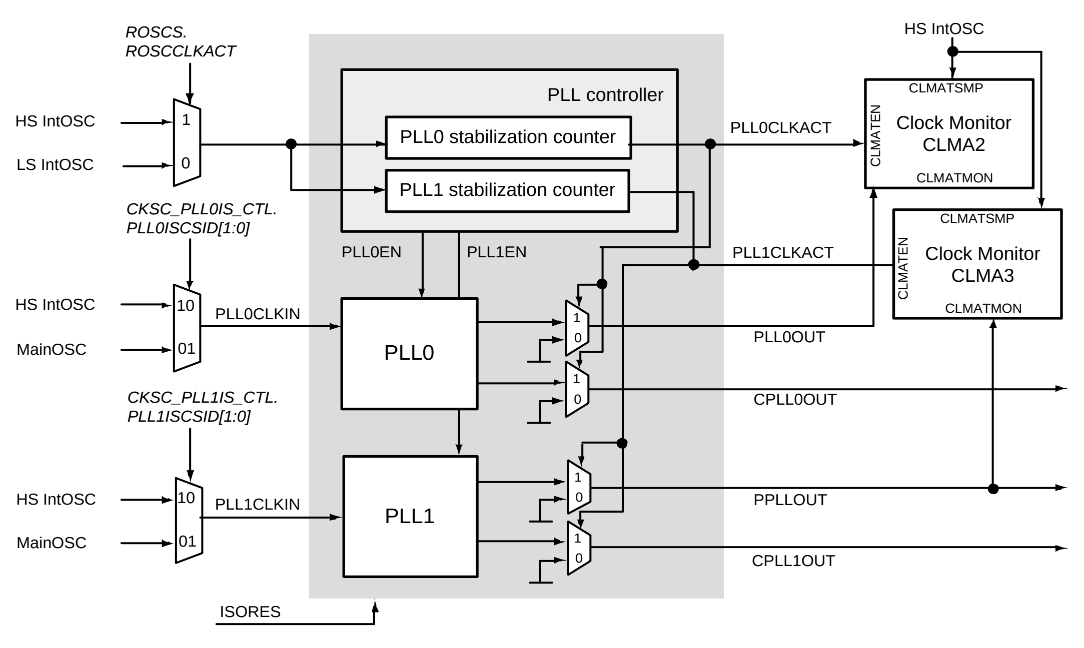


Figure 12AB.7 PLL

PLL Enable

The PLL0 stops operating after reset is released. To use the PLL0, set the PLL0 enable trigger bit (PLL0E.PLL0ENTRG) to 1 to start the PLL0.
The PLL1 stops operating after reset is released. To use the PLL1, set the PLL1 enable trigger bit (PLL1E.PLL1ENTRG) to 1 to start the PLL1.

PLL Stabilization

The PLL0 stabilization counter starts counting the stabilization time, after PLL0 enabled. As long as the PLL0 is not stable, the PLL0CLKACT signal disables the PLL0OUT and CPLL0OUT outputs.
When the PLL0 stabilization counter reaches the value set in PLL0ST.PLL0CLKST[12:0], PLL0OUT and CPLL0OUT are assumed to be stable and PLL0CLKACT switches from 0 to 1 to enable output of PLL0OUT and CPLL0OUT.
The stable and active state of the PLL0OUT and CPLL0OUT clocks is indicated by PLLS.PLL0CLKACT = 1.

The PLL1 stabilization counter starts counting the stabilization time, after PLL1 enabled. As long as the PLL1 is not stable, the PLL1CLKACT signal disables the PPLLOUT and CPLL1OUT outputs. When the PLL1 stabilization counter reaches the predefined value, PPLLOUT and CPLL1OUT are assumed to be stable and PLL1CLKACT switches from 0 to 1 to enable output of PPLLOUT and CPLL1OUT.

The stable and active state of the PPLLOUT and CPLL1OUT clocks is indicated by PLL1S.PLL1CLKACT = 1.

PLL in Stand-by Modes

In STOP mode, the PLL0 and PLL1 are automatically disabled and resumes operation after wake-up from STOP mode, if it was operating before entering STOP mode.

The PLL0 and PLL1 are also automatically disabled when transitioning to DeepSTOP mode. However, after restoring from DeepSTOP mode, the PLL0 and PLL1 need to be reconfigured.

In Cyclic RUN and Cyclic STOP mode, the PLL0 and PLL1 are not available. Do not enable the PLL0 and PLL1 in Cyclic RUN mode.

Clock Monitor Control

The PLL0 activity signal PLL0CLKACT enables or disables supervision by the Clock Monitor CLMA2. In case the PLL0 is inactive (PLL0CLKACT = 0), supervision of the output clock PLL0OUT by CLMA2 is also deactivated.

The PLL1 activity signal PLL1CLKACT enables or disables supervision by the Clock Monitor CLMA3. In case the PLL1 is inactive (PLL1CLKACT = 0), supervision of the output clock PPLLOUT by CLMA3 is also deactivated.

PLL Enable/Disable Trigger

The PLL0 and PLL1 can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLL0E.PLL0ENTRG = 1 starts the PLL0
  Note that setting the enable trigger is only effective if the PLL0 is inactive, i.e. if PLL0S.PLL0CLKACT = 0.
- Disable trigger PLL0E.PLL0DISTRG = 1 stops the PLL0
  Note that setting the disable trigger is only effective if the PLL0 is active, i.e. if PLL0S.PLL0CLKACT = 1.
- Enable trigger PLL1E.PLL1ENTRG = 1 starts the PLL1
  Note that setting the enable trigger is only effective if the PLL1 is inactive, i.e. if PLL1S.PLL1CLKACT = 0.
- Disable trigger PLL1E.PLL1DISTRG = 1 stops the PLL1
  Note that setting the disable trigger is only effective if the PLL1 is active, i.e. if PLL1S.PLL1CLKACT = 1.

PLL Input Clock Selection

The PLL0 input clock (PLL0CLKIN) can be selectable from MainOSC and HS IntOSC by using the CKSC_PLL0IS_CTL register.

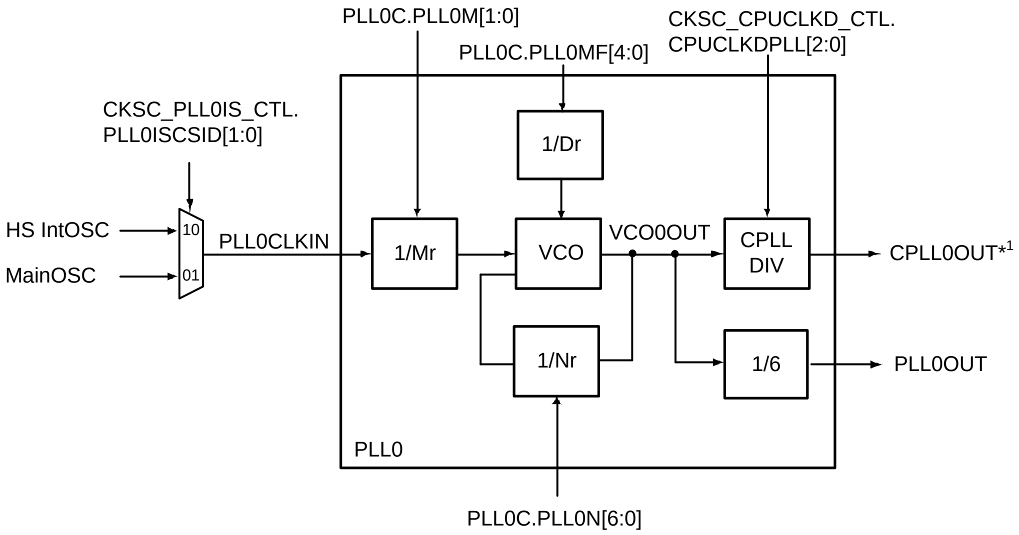
The maximum frequency of CPLL0OUT and PLL0OUT is limited in the case the HS IntOSC is selected as PLL0 input clock.

The PLL1 input clock (PLL1CLKIN) can be selectable from MainOSC and HS IntOSC by using the CKSC_PLL1IS_CTL register.

The maximum frequency of CPLL1OUT and PPLLOUT is limited in the case the HS IntOSC is selected as PLL1 input clock.

12AB.3.5.1 PLL0 Parameters

The PLL0 is configured by a set of parameters, loaded from the control register PLL0C and CKSC_CPUCLKD_CTL.



Note 1. When CPLL0OUT is selected as a source clock of C_ISO_CPUCLK with CKSC_CPUCLKS_CTL, this is valid. When it is not selected, this is invalid.

Figure 12AB.8 PLL0 Circuit

CPLL0OUT and PLL0OUT

The PLL0 has two clock outputs “CPLL0OUT” and “PLL0OUT”. The CPLL0OUT is one of the clock sources of the CPU subsystem, and the PLL0OUT is supervised by the Clock Monitor CLMA2. The CPLL0OUT and PLL0OUT shares the same clock source “VCO0OUT”, which is the output of Voltage Controlled Oscillator (VCO). The clock frequency of the VCO0OUT is calculated by the following formula:

$f_{\mathrm{VCO0OUT}} = f_{\mathrm{PLL0CLKIN}} \times (\mathrm{Nr} / \mathrm{Mr})$

The clock frequency of CPLL0OUT “f_{CPLL0OUT}” and that of PLL0OUT “f_{PLL0OUT}” are integer fractions of the VCO output frequency f_{VCO0OUT}. f_{CPLL0OUT} and f_{PLL0OUT} are calculated by the following formulas:

$f_{\mathrm{CPLL0OUT}} = f_{\mathrm{VCO0OUT}} \times 1/\mathrm{Pr} = f_{\mathrm{PLL0CLKIN}} \times (\mathrm{Nr} / \mathrm{Mr}) \times 1/\mathrm{Pr}$

$f_{\mathrm{PLL0OUT}} = f_{\mathrm{VCO0OUT}} \times 1/6 = f_{\mathrm{PLL0CLKIN}} \times (\mathrm{Nr} / \mathrm{Mr}) \times 1/6$

The values Nr and Mr are derived from PLL0C register bits:

Nr = PLL0C.PLL0N[6:0] + 1

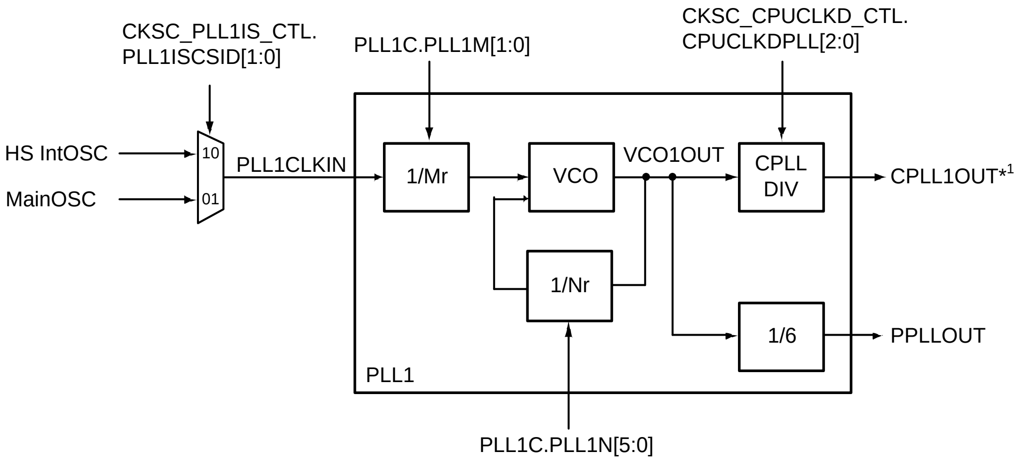
Mr = PLL0C.PLL0M[1:0] + 1

The setting range of Mr: $1 \leq \mathrm{Mr} \leq 3$

The value Pr is derived from CKSC_CPUCLKD_CTL. CPUCLKDPLL[2:0], and the value is 2, 3, 4, or 6.

12AB.3.5.2 PLL1 Parameters

The PLL1 is configured by a set of parameters, loaded from the control register PLL1C and CKSC_CPUCLKD_CTL.



Note 1. When this is selected as a source clock of C_ISO_CPUCLK with CKSC_CPUCLKS_CTL, this is valid.
When this is not selected, this is invalid.

Figure 12AB.9 PLL1 Circuit

CPLL1OUT and PPLLOUT

The PLL1 has two clock outputs “CPLL1OUT” and “PPLLOUT”. The CPLL1OUT is one of the clock sources of the CPU subsystem, and the PPLLOUT is one of the clock sources of the peripheral modules. CPLL1OUT and PPLLOUT shares the same clock source “VCO1OUT”, which is the output of the voltage controlled oscillator (VCO). The clock frequency of VCO1OUT is calculated by the following formula:

$f_{VCO1OUT} = f_{PLL1CLKIN} \times (Nr / Mr)$

The clock frequency of CPLL1OUT “$f_{CPLL1OUT}$” and that of PPLLOUT “$f_{PPLLOUT}$” are integer fractions of the VCO output frequency $f_{VCO1OUT}$. $f_{CPLL1OUT}$ and $f_{PPLLOUT}$ are calculated by the following formulas:

$f_{CPLL1OUT} = f_{VCO1OUT} \times 1/Pr = f_{PLL1CLKIN} \times (Nr / Mr) \times 1/Pr$

$f_{PPLLOUT} = f_{VCO1OUT} \times 1/6 = f_{PLL1CLKIN} \times (Nr / Mr) \times 1/6$

The values Nr and Mr are derived from PLL1C register bits:

Nr = PLL1C.PLL1N[5:0] + 1

Mr = PLL1C.PLL1M[1:0] + 1

The setting range of Mr: $1 \leq Mr \leq 3$

The value Pr is derived from CKSC_CPUCLKD_CTL. CPUCLKDPLL[2:0], and the value is 4 or 6.

12AB.4 Registers

12AB.4.1 List of Registers

The registers of the clock controller are listed below.

Table 12AB.2 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	Clock oscillator registers:		
	MainOSC enable register	MOSCE	FFF8 1100_H
	MainOSC status register	MOSCS	FFF8 1104_H
	MainOSC control register	MOSCC	FFF8 1108_H
	MainOSC stabilization time register	MOSCST	FFF8 $110C_H$
	MainOSC stop mask register	MOSCSTPM	FFF8 1118_H
	MainOSC mode control register	MOSCM	FFF8 $111C_H$
	SubOSC enable register	SOSCE*1	FFF8 1200_H
	SubOSC status register	SOSCS*1	FFF8 1204_H
	SubOSC stabilization time register	SOSCST*1	FFF8 $120C_H$
	HS IntOSC enable register	ROSCE	FFF8 1000_H
	HS IntOSC status register	ROSCS	FFF8 1004_H
	HS IntOSC stop mask register	ROSCSTPM	FFF8 1018_H
	HS IntOSC user trimming register	ROSCUT	FFF8 $101C_H$
	PLL0 enable register	PLL0E	FFF8 9000_H
	PLL0 status register	PLL0S	FFF8 9004_H
	PLL0 control register	PLL0C	FFF8 9008_H
	PLL0 stabilization time register	PLL0ST	FFF8 $900C_H$
	PLL1 enable register	PLL1E	FFF8 9100_H
	PLL1 status register	PLL1S	FFF8 9104_H
	PLL1 control register	PLL1C	FFF8 9108_H
	PLL0 input clock selection register	CKSC_PLL0IS_CTL	FFF8 $A700_H$
	PLL0 input clock active register	CKSC_PLL0IS_ACT	FFF8 $A708_H$
	PLL1 input clock selection register	CKSC_PLL1IS_CTL	FFF8 $A710_H$
	PLL1 input clock active register	CKSC_PLL1IS_ACT	FFF8 $A718_H$
	PPLLCLK source clock selection register	CKSC_PPLLCLKS_CTL	FFF8 $A010_H$
	PPLLCLK source clock active register	CKSC_PPLLCLKS_ACT	FFF8 $A018_H$
	Clock selector control register:		
	C_AWO_WDTA clock divider selection register	CKSC_AWDTAD_CTL	FFF8 2000_H
	C_AWO_WDTA clock divider active register	CKSC_AWDTAD_ACT	FFF8 2008_H
	C_AWO_WDTA stop mask register	CKSC_AWDTAD_STPM	FFF8 2018_H
	C_AWO_TAUJ source clock selection register	CKSC_ATAUJS_CTL	FFF8 2100_H
	C_AWO_TAUJ source clock active register	CKSC_ATAUJS_ACT	FFF8 2108_H
	C_AWO_TAUJ clock divider selection register	CKSC_ATAUJD_CTL	FFF8 2200_H
	C_AWO_TAUJ clock divider active register	CKSC_ATAUJD_ACT	FFF8 2208_H
	C_AWO_TAUJ stop mask register	CKSC_ATAUJD_STPM	FFF8 2218_H
	C_AWO_RTCA source clock selection register	CKSC_ARTCAS_CTL	FFF8 2300_H
	C_AWO_RTCA source clock active register	CKSC_ARTCAS_ACT	FFF8 2308_H
	C_AWO_RTCA clock divider selection register	CKSC_ARTCAD_CTL	FFF8 2400_H

Table 12AB.2 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	C_AWO_RTCA clock divider active register	CKSC_ARTCAD_ACT	FFF8 2408_H
	C_AWO_RTCA stop mask register	CKSC_ARTCAD_STPM	FFF8 2418_H
	C_AWO_ADCA source clock selection register	CKSC_AADCAS_CTL	FFF8 2500_H
	C_AWO_ADCA source clock active register	CKSC_AADCAS_ACT	FFF8 2508_H
	C_AWO_ADCA clock divider selection register	CKSC_AADCAD_CTL	FFF8 2600_H
	C_AWO_ADCA clock divider active register	CKSC_AADCAD_ACT	FFF8 2608_H
	C_AWO_ADCA stop mask register	CKSC_AADCAD_STPM	FFF8 2618_H
	C_AWO_FOUT source clock selection register	CKSC_AFOUTS_CTL	FFF8 2700_H
	C_AWO_FOUT source clock active register	CKSC_AFOUTS_ACT	FFF8 2708_H
	C_AWO_FOUT stop mask register	CKSC_AFOUTS_STPM	FFF8 2718_H
	C_ISO_CPUCLK source clock selection register	CKSC_CPUCLKS_CTL	FFF8 $A000_H$
	C_ISO_CPUCLK source clock active register	CKSC_CPUCLKS_ACT	FFF8 $A008_H$
	C_ISO_CPUCLK clock divider selection register	CKSC_CPUCLKD_CTL	FFF8 $A100_H$
	C_ISO_CPUCLK clock divider active register	CKSC_CPUCLKD_ACT	FFF8 $A108_H$
	C_ISO_PERI1 source clock selection register	CKSC_IPERI1S_CTL	FFF8 $A200_H$
	C_ISO_PERI1 source clock active register	CKSC_IPERI1S_ACT	FFF8 $A208_H$
	C_ISO_PERI2 source clock selection register	CKSC_IPERI2S_CTL	FFF8 $A300_H$
	C_ISO_PERI2 source clock active register	CKSC_IPERI2S_ACT	FFF8 $A308_H$
	C_ISO_LIN source clock selection register	CKSC_ILINS_CTL	FFF8 $A400_H$
	C_ISO_LIN source clock active register	CKSC_ILINS_ACT	FFF8 $A408_H$
	C_ISO_ADCA source clock selection register	CKSC_IADCAS_CTL*1	FFF8 $A500_H$
	C_ISO_ADCA source clock active register	CKSC_IADCAS_ACT*1	FFF8 $A508_H$
	C_ISO_ADCA clock divider selection register	CKSC_IADCAD_CTL*1	FFF8 $A600_H$
	C_ISO_ADCA clock divider active register	CKSC_IADCAD_ACT*1	FFF8 $A608_H$
	C_ISO_LIN clock divider selection register	CKSC_ILIND_CTL	FFF8 $A800_H$
	C_ISO_LIN clock divider active register	CKSC_ILIND_ACT	FFF8 $A808_H$
	C_ISO_LIN stop mask register	CKSC_ILIND_STPM	FFF8 $A818_H$
	C_ISO_CAN source clock selection register	CKSC_ICANS_CTL	FFF8 $A900_H$
	C_ISO_CAN source clock active register	CKSC_ICANS_ACT	FFF8 $A908_H$
	C_ISO_CAN stop mask register	CKSC_ICANS_STPM	FFF8 $A918_H$
	C_ISO_CANOSC clock divider selection register	CKSC_ICANOSCD_CTL	FFF8 $AA00_H$
	C_ISO_CANOSC clock divider active register	CKSC_ICANOSCD_ACT	FFF8 $AA08_H$
	C_ISO_CANOSC stop mask register	CKSC_ICANOSCD_STPM	FFF8 $AA18_H$
	C_ISO_CSI source clock selection register	CKSC_ICSIS_CTL	FFF8 $AB00_H$
	C_ISO_CSI source clock active register	CKSC_ICSIS_ACT	FFF8 $AB08_H$
	C_ISO_IIC source clock selection register	CKSC_IIICS_CTL	FFF8 $AC00_H$
	C_ISO_IIC source clock active register	CKSC_IIICS_ACT	FFF8 $AC08_H$

Note 1. It isn't supported with 100-pin product.

12AB.4.2 Clock Oscillator Registers

12AB.4.2.1 MOSCE — MainOSC Enable Register

This register is used to start and stop the MainOSC.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1100$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCDISTRG	MOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.3 MOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MOSCDISTRG	MainOSC Disable Trigger*1,*3 MOSCSTPM.MOSCSTPMSK = 0 0: No function 1: Stops MainOSC MOSCSTPM.MOSCSTPMSK = 1 When writing this bit while MOSCSTPM.MOSCSTPMSK = 1, the write value should be 0. 0: No function 1: Setting Prohibited This bit is automatically cleared to 0 by hardware after MainOSC is disabled.
0	MOSCENTRG	MainOSC Enable Trigger*2,*3 0: No function 1: Starts MainOSC This bit is automatically cleared to 0 by hardware after MainOSC is enabled.

Note 1. Follow the procedure given below for stopping the MainOSC by using MOSCDISTRG.

1. Confirm that the MainOSC is neither waiting for being enabled nor stopped (both MOSCE.MOSCDISTRG = 0 and MOSCE.MOSCENTRG = 0).
2. Confirm that the MainOSC is active (MOSCS.MOSCCLKACT = 1).
3. Check that there is no clock domain for which the MainOSC is selected. If the MainOSC is selected for a clock domain, disable the setting or select a clock source other than MainOSC.
4. Confirm that the MainOSC stop mask register (MOSCSTPM) is NOT set to “MainOSC continues operation in standby mode” (MOSCSTPM.MOSCSTPMSK = 1). Otherwise, set the register to “MainOSC stops operation in stand-by mode” (MOSCSTPM.MOSCSTPMSK = 0).
5. Stop the MainOSC (MOSCE.MOSCDISTRG = 1).
6. Confirm that the MainOSC has been stopped (MOSCS.MOSCCLKACT = 0).

Note 2. Follow the procedure given below for starting the MainOSC by using MOSCENTRG

1. Confirm that the MainOSC is neither waiting for being enabled nor stopped (both MOSCE.MOSCDISTRG = 0 and MOSCE.MOSCENTRG = 0).
2. Confirm that the MainOSC is inactive (MOSCS.MOSCCLKACT = 0).
3. Start the MainOSC (MOSCE.MOSCENTRG = 1).
4. Confirm that the MainOSC has been started (MOSCS.MOSCCLKACT = 1).

Note 3. Starting and stopping the MainOSC at the same time by using the start and stop bits, i.e., by setting MOSCE.MOSCENTRG = 1 and MOSCE.MOSCDISTRG = 1 at the same time is not allowed.

12AB.4.2.2 MOSCS — MainOSC Status Register

This register provides active status information about the MainOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 1104$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1^{*1}	0^{*1}
	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12AB.4 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	MOSCCLKACT	MainOSC Active Status 0: MainOSC is inactive 1: MainOSC is active
1, 0	Reserved	When read, an undefined value is returned.

12AB.4.2.3 MOSCC — MainOSC Control Register

This register is used to specify amplification gain of the MainOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1108_H

Value after reset: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCAMPSEL [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.5 MOSCC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MOSCAMPSEL[1:0]	MainOSC Amplification Gain Selection 00: Amplification gain for fx = 24 MHz 01: Amplification gain for fx = 20 MHz 10: Amplification gain for fx = 16 MHz 11: Amplification gain for fx = 8 MHz

CAUTION

Set this register when MainOSC is stopped.

12AB.4.2.4 MOSCST — MainOSC Stabilization Time Register

This register determines the MainOSC stabilization time.

This register is initialized by the power-up reset signal PURES and $\overline{\mathrm{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 110C$_\text{H}$

Value after reset: 0000 44C0$_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKST16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSCCLKST[15:0]															
Value after reset	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.6 MOSCST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16 to 0	MOSCCLKST[16:0]	The MOSCCLKST[16:0] bits specify the count value for the MainOSC stabilization counter. ● If HS IntOSC active (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f_{RH} ● If HS IntOSC inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = MOSCCLKST[16:0] / f_{RL}

NOTE

See **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for information about the MainOSC stabilization time.

CAUTION

Set this register when MainOSC is stopped.

12AB.4.2.5 MOSCSTPM — MainOSC Stop Mask Register

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1118$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12AB.7 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCSTPMSK	MainOSC Stop Request Mask 0: MainOSC stops operation in stand-by mode. MainOSC stops operation in the case the MainOSC disable trigger MOSCE.MOSCDISTRG is set to 1. 1: MainOSC continues operation in stand-by mode.

12AB.4.2.6 MOSCM — MainOSC Mode Control Register

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 111C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12AB.8 MOSCM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCM	MainOSC Mode control 0: OSC mode. (default) 1: EXCLK mode. MainOSC amplifier is disabled.

CAUTION

Set this register when MainOSC is stopped.

NOTE

EXCLK mode is a mode to directly input clock to X1. For details, see **Section 12AB.3.1, Main Oscillator (MainOSC)**.

12AB.4.2.7 SOSCE — SubOSC Enable Register

This register is used to start and stop the SubOSC.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1200_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCDISTRG	SOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.9 SOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SOSCDISTRG	SubOSC Disable Trigger*[1,*3] 0: No function 1: Stops SubOSC This bit is automatically cleared to 0 by hardware after SubOSC is disabled.
0	SOSCENTRG	SubOSC Enable Trigger*[2,*3] 0: No function 1: Starts SubOSC This bit is automatically cleared to 0 by hardware after SubOSC is enabled.

Note 1. Follow the procedure given below for stopping the SubOSC by using SOSCDISTRG.

1. Confirm that the SubOSC is neither waiting for being enabled nor stopped (SOSCE.SOSCDISTRG = 0 and SOSCE.SOSCENTRG = 0).
2. Confirm that the SubOSC is active (SOSCS.SOSCCLKACT = 1).
3. Check that there is no clock domain for which the SubOSC is selected. If the SubOSC is selected for a clock domain, disable the setting or select a clock source other than the SubOSC.
4. Stop the SubOSC (SOSCE.SOSCDISTRG = 1).
5. Confirm that the SubOSC has been stopped (SOSCS.SOSCCLKACT = 0).

Note 2. Follow the procedure given below for starting the SubOSC by using SOSCENTRG.

1. Confirm that the SubOSC is neither waiting for being enabled nor stopped (SOSCE.SOSCDISTRG = 0 and SOSCE.SOSCENTRG = 0).
2. Confirm that the SubOSC is inactive (SOSCS.SOSCCLKACT = 0).
3. Start the SubOSC (SOSCE.SOSCENTRG = 1).
4. Confirm that the SubOSC has been started (SOSCS.SOSCCLKACT = 1).

Note 3. Starting and stopping the SubOSC at the same time by using the start and stop bits, i.e., by setting SOSCE.SOSCENTRG = 1 and SOSCE.SOSCDISTRG = 1 at the same time is not allowed.

12AB.4.2.8 SOSCS — SubOSC Status Register

This register provides active status information about the SubOSC.

This register is initialized by the power-up reset signal PURES.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 1204_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1^{*1}	0^{*1}
	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12AB.10 SOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SOSCCLKACT	SubOSC Activation Status 0: SubOSC is inactive 1: SubOSC is active
1, 0	Reserved	When read, an undefined value is returned.

12AB.4.2.9 SOSCST — SubOSC Stabilization Time Register

This register determines the SubOSC stabilization time.

This register is initialized by the power-up reset signal PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 $120C_H$

Value after reset: $010C\ 8E00_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SOSCCLKST[29:16]													
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOSCCLKST[15:0]															
Value after reset	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.11 SOSCST Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 0	SOSCCLKST[29:0]	The SOSCCLKST[29:0] bits specify the count value for the SubOSC stabilization time counter. ● If the HS IntOSC is active (ROSCS.ROSCCLKACT = 1): Stabilization time = SOSCCLKST[29:0] / f_{RH} ● If the HS IntOSC is inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = SOSCCLKST[29:0] / f_{RL}

NOTE

See **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** for information about the SubOSC stabilization time.

CAUTION

Set this register when SubOSC is stopped.

12AB.4.2.10 ROSCE — HS IntOSC Enable Register

This register is used to stop the HS IntOSC operation.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

CAUTION

Set the ROSCE.ROSCDISTRG bit only when the $\overline{\text{CLMA0RES}}$ has occurred. In other cases, setting this bit is prohibited.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCDISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 12AB.12 ROSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ROSCDISTRG	HS IntOSC Disable Trigger ROSCSTPM.ROSCSTPMSK = 0 0: No function 1: Stops HS IntOSC ROSCSTPM.ROSCSTPMSK = 1 Setting prohibited
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

12AB.4.2.11 ROSCS — HS IntOSC Status Register

This register provides active status information about the HS IntOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 1004_H
Value after reset:	0000 0007_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1^{*1}	0^{*1}
	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12AB.13 ROSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	ROSCCLKACT	HS IntOSC Active Status 0: HS IntOSC is inactive 1: HS IntOSC is active
1, 0	Reserved	When read, an undefined value is returned.

12AB.4.2.12 **ROSCSTPM — HS IntOSC Stop Mask Register**

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1018$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12AB.14 ROSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ROSCSTPMSK	HS IntOSC Stop Request Mask 0: HS IntOSC stops operation in stand-by mode 1: HS IntOSC continues operation in stand-by mode Do not set the HS IntOSC disable trigger ROSCE.ROSCDISTRG to 1 while ROSCSTPMSK bit is set to 1.

12AB.4.2.13 PLL0E — PLL0 Enable Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 9000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0DISTRG	PLL0ENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.15 PLL0E Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLL0DISTRG	PLL0 Disable Trigger[*1,*4] 0: No function 1: Stops PLL0 This bit is automatically cleared to 0 by hardware after PLL0 is disabled.
0	PLL0ENTRG	PLL0 Enable Trigger[*2,*3,*4] 0: No function 1: Starts PLL0 This bit is automatically cleared to 0 by hardware after PLL0 is activated.

Note 1. Follow the procedure given below for stopping the PLL0 by using PLL0DISTRG.

1. Confirm that the PLL0 is neither waiting for being enabled nor stopped (both PLL0E.PLL0DISTRG = 0 and PLL0E.PLL0ENTRG = 0).
2. Confirm that the PLL0 is active (PLL0S.PLL0CLKACT = 1).
3. Check that there is no clock domain for which the PLL0 is selected. If the PLL0 is selected for a clock domain, disable the setting or select a clock source other than the PLL0.
4. Stop the PLL0 (PLL0E.PLL0DISTRG = 1).
5. Confirm that the PLL0 has been stopped (PLL0S.PLL0CLKACT = 0).

Note 2. Before starting PLL0 using PLL0ENTRG, confirm that the PLL0 input clock (MainOSC or HS IntOSC, selected by the CKSC_PLL0IS_CTL) is operating.

Note 3. Follow the procedure given below for starting the PLL0 by using PLL0ENTRG.

1. Confirm that the PLL0 is neither waiting for being enabled nor stopped (both PLL0E.PLL0DISTRG = 0 and PLL0E.PLL0ENTRG = 0).
2. Confirm that the PLL0 is inactive (PLL0S.PLL0CLKACT = 0).
3. Start the PLL0 (PLL0E.PLL0ENTRG = 1).
4. Confirm that the PLL0 has been started (PLL0S.PLL0CLKACT = 1).

Note 4. Starting and stopping the PLL0 at the same time by using the start and stop bits, i.e., by setting PLL0E.PLL0ENTRG = 1 and PLL0E.PLL0DISTRG = 1 at the same time is not allowed.

12AB.4.2.14 PLL0S — PLL0 Status Register

This register provides active status information about the PLL0.

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 9004_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1[*1]	0[*1]
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0CLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12AB.16 PLL0S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLL0CLKACT	PLL0 Active Status 0: PLL0 is inactive 1: PLL0 is active
1, 0	Reserved	When read, an undefined value is returned.

12AB.4.2.15 PLL0C — PLL0 Control Register

This register is used to set the PLL0 VCO output clock frequency $f_{VCO0OUT}$, shown in **Section 12AB.3.5.1, PLL0 Parameters**.

This register can only be written, if the PLL0 is disabled.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 9008_H

Value after reset: 6000 $503B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLL0FVV[1:0]		PLL0MF[4:0]					—	PLL0ADJ[2:0]			—	—	—	—
Value after reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PLL0MD	PLL0SMD	PLL0M[1:0]		—	—	—	—	PLL0N[6:0]						
Value after reset	0	1	0	1	0	0	0	0	0	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.17 PLL0C Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>30, 29</td><td>PLL0FVV[1:0]</td><td>VCO Output Frequency Range Setting.
<table>
<tr><th>PLL0FVV1</th><th>PLL0FVV0</th><th>VCO Output Frequency Range</th></tr>
<tr><td>1</td><td>1</td><td>440 MHz to 480 MHz.</td></tr>
<tr><td>—</td><td>—</td><td>Settings are prohibited</td></tr>
</table></td></tr>
</table>

Table 12AB.17 PLL0C Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>28 to 24</td><td>PLL0MF[4:0]</td><td>Frequency Modulation Cycle Setting
The modulation frequency is calculated as:
Modulation frequency = (fx / Mr) / (MFD × 4)
Example: When fX = 8 MHz, Mr = 1, and MFD = 00011_B = 20,
fX = (8/1) / (20 × 4) = 100 [KHz]
For the modulation frequency setting range, see the electrical characteristics.
<table>
<tr><th>PLL0 MF4</th><th>PLL0 MF3</th><th>PLL0 MF2</th><th>PLL0 MF1</th><th>PLL0 MF0</th><th>Modulation Frequency Division Ratio MFD</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>10</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>12</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>18</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>20</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>22</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>26</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>28</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>30</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>34</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>38</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>40</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>44</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>50</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>56</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>58</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>60</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>62</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>66</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>72</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>76</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>80</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>84</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>86</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>100</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>120</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>126</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>134</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>150</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>166</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>200</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>250</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>300</td></tr>
</table>
</td></tr>
<tr><td>23</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 12AB.17 PLL0C Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>22 to 20</td><td>PLL0ADJ[2:0]</td><td>Frequency Modulation Range Setting
<table>
<tr><th>PLL0ADJ2</th><th>PLL0ADJ1</th><th>PLL0ADJ0</th><th>Frequency Modulation Range</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1%</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2%</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3%</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4%</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5%</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6%</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>8%</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>10%</td></tr>
</table></td></tr>
<tr><td>19 to 15</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>14</td><td>PLL0MD</td><td>Operating Mode Setting
0: Setting prohibited
1: SSCG mode (modulation frequency)</td></tr>
<tr><td>13</td><td>PLL0SMD</td><td>Modulation Mode Setting in SSCG Mode
1: Setting prohibited
0: Down spread modulation</td></tr>
<tr><td>12, 11</td><td>PLL0M[1:0]</td><td>Division ratio Mr is set.
For PLL0M[1:0] settings, see **Table 12AB.18, PLL0 Output Table**.</td></tr>
<tr><td>10 to 7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>6 to 0</td><td>PLL0N[6:0]</td><td>Division ratio Nr is set.
For PLL0N[6:0] settings, see **Table 12AB.18, PLL0 Output Table**.</td></tr>
</table>

CAUTION

Set this register when PLL0 is stopped.

Table 12AB.18 PLL0 Output Table

<table>
<tr><th rowspan="2">PLL0CLKIN frequency $f_{PLL0CLKIN}$ (MHz)</th><th rowspan="2">PLL0C. PLL0M[1:0] (Mr)*3</th><th rowspan="2">PLL0C. PLL0N[6:0] (Nr)*3</th><th rowspan="2">VCO0OUT frequency $f_{VCO0OUT}$ (MHz)</th><th colspan="4">CPLL0OUT frequency $f_{CPLL0OUT}$ (MHz)*1</th></tr>
<tr><th>VCO0OUT × 1/2 (Products of CPU frequency 240 MHz max. at CKDIVMD “1”)</th><th>VCO0OUT × 1/3 (Products of CPU frequency 240 MHz max. at CKDIVMD “1” /Products of CPU frequency 160 MHz max.)</th><th>VCO0OUT × 1/4 (Products of CPU frequency 240 MHz max. / Products of CPU frequency 160 MHz max.)</th><th>VCO0OUT × 1/6 (Products of CPU frequency 240 MHz max. / Products of CPU frequency 160 MHz max.)</th></tr>
<tr><td>8 (MainOSC)</td><td>00_B (Mr = 1)</td><td>$3B_H$ (Nr = 60)</td><td>480.0</td><td>240.0</td><td>160.0</td><td>120.0</td><td>N/A</td></tr>
<tr><td>16 (MainOSC)</td><td>01_B (Mr = 2)</td><td>$3B_H$ (Nr = 60)</td><td>480.0</td><td>240.0</td><td>160.0</td><td>120.0</td><td>N/A</td></tr>
<tr><td>20 (MainOSC)</td><td>01_B (Mr = 2)</td><td>$2F_H$ (Nr = 48)</td><td>480.0</td><td>240.0</td><td>160.0</td><td>120.0</td><td>N/A</td></tr>
<tr><td>24 (MainOSC)</td><td>01_B (Mr = 2)</td><td>27_H (Nr = 40)</td><td>480.0</td><td>240.0</td><td>160.0</td><td>120.0</td><td>N/A</td></tr>
<tr><td>24 (MainOSC)</td><td>10_B (Mr = 3)</td><td>$3B_H$ (Nr = 60)</td><td>480.0</td><td>240.0</td><td>160.0</td><td>120.0</td><td>N/A</td></tr>
<tr><td>8 (HS IntOSC)</td><td>00_B (Mr = 1)</td><td>$3B_H$ (Nr = 60)</td><td>480.0</td><td>N/A</td><td>N/A</td><td>N/A</td><td>80.0*2</td></tr>
</table>

Note 1. The CPLL0OUT frequency is defined by CKSC_CPUCLKD_CTL.CPUCLKDPLL[2:0]. Refer to the CKSC_CPUCLKD_CTL register description.

Note 2. The limit of CPLL0OUT frequency is 80 MHz (typ.) when HS IntOSC is selected as clock source of PLL0.

Note 3. Settings other than those shown in this table are prohibited.

12AB.4.2.16 PLL0ST — PLL0 Stabilization Time Register

This register specifies stabilization time of the PLL0.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 900C$_{\text{H}}$
Value after reset:	0000 1B80$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PLL0CLKST[12:0]												
Value after reset	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.19 PLL0ST Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 13</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>12 to 0</td><td>PLL0CLKST[12:0]</td><td>PLL0 Stabilization Time Setting
The PLL0 stabilization counter counts cycles of the HS IntOSC.
Set the following value as the PLL0 operating mode settings.
<table>
<tr><th>Mode</th><th>PLL0CLKST[12:0]</th></tr>
<tr><td>SSCG mode</td><td>1B80$_{\text{H}}$</td></tr>
</table></td></tr>
</table>

12AB.4.2.17 **PLL1E — PLL1 Enable Register**

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 9100$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1DISTRG	PLL1ENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.20 PLL1E Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLL1DISTRG	PLL1 Disable Trigger[*1,*4] 0: No function 1: Stops PLL1 This bit is automatically cleared to 0 by hardware after PLL1 is disabled.
0	PLL1ENTRG	PLL1 Enable Trigger[*2,*3,*4] 0: No function 1: Starts PLL1 This bit is automatically cleared to 0 by hardware after PLL1 is activated.

Note 1. Follow the procedure given below for stopping the PLL1 by using PLL1DISTRG.

1. Confirm that the PLL1 is neither waiting for being enabled nor stopped (both PLL1E.PLL1DISTRG = 0 and PLL1E.PLL1ENTRG = 0).
2. Confirm that the PLL1 is active (PLL1S.PLL1CLKACT = 1).
3. Check that there is no clock domain for which the PLL1 is selected. If the PLL1 is selected for a clock domain, disable the setting or select a clock source other than the PLL1.
4. Stop the PLL1 (PLL1E.PLL1DISTRG = 1).
5. Confirm that the PLL1 has been stopped (PLL1S.PLL1CLKACT = 0).

Note 2. Before starting PLL1 using PLL1ENTRG, confirm that the PLL1 input clock (MainOSC or HS IntOSC, selected by the CKSC_PLL1IS_CTL) is operating.

Note 3. Follow the procedure given below for starting the PLL1 by using PLL1ENTRG.

1. Confirm that the PLL1 is neither waiting for being enabled nor stopped (both PLL1E.PLL1DISTRG = 0 and PLL1E.PLL1ENTRG = 0).
2. Confirm that the PLL1 is inactive (PLL1S.PLL1CLKACT = 0).
3. Start the PLL1 (PLL1E.PLL1ENTRG = 1).
4. Confirm that the PLL1 has been started (PLL1S.PLL1CLKACT = 1).

Note 4. Starting and stopping the PLL1 at the same time by using the start and stop bits, i.e., by setting PLL1E.PLL1ENTRG = 1 and PLL1E.PLL1DISTRG = 1 at the same time is not allowed.

12AB.4.2.18 PLL1S — PLL1 Status Register

This register provides active status information about the PLL1.

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 9104$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1[*1]	0[*1]
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1CLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12AB.21 PLL1S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLL1CLKACT	PLL1 Active Status 0: PLL1 is inactive 1: PLL1 is active
1, 0	Reserved	When read, an undefined value is returned.

12AB.4.2.19 PLL1C — PLL1 Control Register

This register is used to set the PLL1 VCO output clock frequency $f_{VCO1OUT}$, shown in **Section 12AB.3.5.2, PLL1 Parameters**.

This register can only be written, if the PLL1 is disabled.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 9108_H

Value after reset: 0001 $133B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PLL1M[1:0]		—	—	—	—	—	PLL1N[5:0]					
Value after reset	0	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.22 PLL1C Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12, 11	PLL1M[1:0]	Division ratio Mr is set. For PLL1M[1:0] settings, see **Table 12AB.23, PLL1 Output Table**.
10 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	PLL1N[5:0]	Division ratio Nr is set. For PLL1N[5:0] settings, see **Table 12AB.23, PLL1 Output Table**.

CAUTION

Set this register when PLL1 is stopped.

Table 12AB.23 PLL1 Output Table

PLL1CLKIN frequency $f_{PLL1CLKIN}$ (MHz)	PLL1C. PLL1M[1:0] (Mr)*3	PLL1C. PLL1N[5:0] (Nr)*3	VCO1OUT frequency $f_{VCO1OUT}$ (MHz)	CPLL1OUT frequency $f_{CPLL1OUT}$ (MHz)*1		PPLLOUT frequency $f_{PPLLOUT}$ (MHz)
				VCO1OUT × 1/4	VCO1OUT × 1/6	
8 (Main OSC)	00_B (Mr = 1)	$3B_H$ (Nr = 60)	480.0	120.0	80.0	80.0
16 (MainOSC)	01_B (Mr = 2)	$3B_H$ (Nr = 60)	480.0	120.0	80.0	80.0
20 (MainOSC)	01_B (Mr = 2)	$2F_H$ (Nr = 48)	480.0	120.0	80.0	80.0
24 (MainOSC)	01_B (Mr = 2)	27_H (Nr = 40)	480.0	120.0	80.0	80.0
24 (MainOSC)	10_B (Mr = 3)	$3B_H$ (Nr = 60)	480.0	120.0	80.0	80.0
8 (HS IntOSC)	00_B (Mr = 1)	$3B_H$ (Nr = 60)	480.0	N/A	80.0*2	80.0

Note 1. The CPLL1OUT frequency is defined by CKSC_CPUCLKD_CTL.CPUCLKDPLL[2:0]. Refer to the CKSC_CPUCLKD_CTL register description.

Note 2. The limit of CPLL1OUT frequency is 80 MHz (typ.) when HS IntOSC is selected as clock source of PLL1.

Note 3. Settings other than those shown in this table are prohibited.

12AB.4.2.20 PLL0 Input Clock Selection

(1) CKSC_PLL0IS_CTL — PLL0 Input Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A700$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0ISCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.24 CKSC_PLL0IS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PLL0ISCSID[1:0]	Source Clock Setting for PLL0 input clock 01$_B$: MainOSC (Default) 10$_B$: HS IntOSC*[1] Other than above: Setting prohibited

Note 1. The maximum frequency of CPLL0OUT and PLL0OUT is limited when the HS IntOSC is selected as the PLL0 input clock.

CAUTION

Set this register when PLL0 is stopped.

(2) CKSC_PLL0IS_ACT — PLL0 Input Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A708$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0ISACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.25 CKSC_PLL0IS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	PLL0ISACT[1:0]	Source clock for currently active PLL0 input clock

12AB.4.2.21 PLL1 Input Clock Selection

(1) CKSC_PLL1IS_CTL — PLL1 Input Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 $A710_H$
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1ISCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.26 CKSC_PLL1IS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PLL1ISCSID[1:0]	Source Clock Setting for PLL1 input clock 01_B: MainOSC (Default) 10_B: HS IntOSC*[1] Other than above: Setting prohibited

Note 1. The maximum frequency of CPLL1OUT and PPLLOUT is limited when the HS IntOSC is selected as the PLL1 input clock.

CAUTION

Set this register when PLL1 is stopped.

(2) CKSC_PLL1IS_ACT — PLL1 Input Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A718$_{\text{H}}$
Value after reset:	0000 0001$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1ISACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.27 CKSC_PLL1IS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	PLL1ISACT[1:0]	Source clock for currently active PLL1 input clock

12AB.4.2.22 PPLLCLK Source Clock Selection

(1) CKSC_PPLLCLKS_CTL — PPLLCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 $A010_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPLLCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.28 CKSC_PPLLCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PPLLCLKSCSID[1:0]	Source Clock Setting for PPLLCLK 00_B: Setting prohibited 01_B: EMCLK (default) 10_B: MainOSC 11_B: PPLLOUT

(2) CKSC_PPLLCLKS_ACT — PPLLCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: $\text{FFF8 A018}_{\text{H}}$

Value after reset: $\text{0000 0001}_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPLLCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.29 CKSC_PPLLCLKS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	PPLLCLKSACT[1:0]	Source clock for currently active PPLLCLK^{*1}

Note 1. The data read from this register is 00_B when the following conditions is satisfied.

- The selected source clock for all of the following clock domains is other than PPLLCLK (or PPLLCLK2): C_ISO_PERI1, C_ISO_PERI2, C_ISO_LIN, C_ISO_ADCA, C_ISO_CAN, C_ISO_CSI, C_ISO_IIC, C_ISO_ADCA
- The OPBT0.FLXA0EN bit is set to 0 (FlexRay is disabled).

12AB.4.2.23 ROSCUT — HS IntOSC User Trimming Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 $101C_H$
Value after reset:	0000 $0XXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FADJUST[8:0]								
Value after reset	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.30 ROSCUT Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	FADJUST[8:0]	Frequency Adjustment parameters of HS IntOSC. The value after reset of this register is pre-set value of HS IntOSC trimming data. Overwrite this register with “read value + 1” or “read value – 1” until the HS IntOSC frequency becomes target frequency range.

12AB.4.3 Clock Selector Control Register

12AB.4.3.1 WDTA0 Clock Domain C_AWO_WDTA

(1) CKSC_AWDTAD_CTL — C_AWO_WDTA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2000_H
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.31 CKSC_AWDTAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AWDTADCSID[1:0]	Clock Divider Setting for C_AWO_WDTA 00_B: Setting prohibited 01_B: LS IntOSC / 128 (default) 10_B: LS IntOSC / 1 11_B: Setting prohibited

(2) CKSC_AWDTAD_ACT — C_AWO_WDTA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2008$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.32 CKSC_AWDTAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AWDTADACT[1:0]	Clock divider for currently active C_AWO_WDTA

(3) CKSC_AWDTAD_STPM — C_AWO_WDTA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2018_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.33 CKSC_AWDTAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AWDTADSTPMSK*[1]	0: Clock domain C_AWO_WDTA is stopped in stand-by mode. 1: Clock domain C_AWO_WDTA is not stopped in stand-by mode.

Note 1. The return time from stand-by mode can be shortened by setting this bit to 1.

12AB.4.3.2 TAUJ Clock Domain C_AWO_TAUJ

(1) CKSC_ATAUJS_CTL — C_AWO_TAUJ Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2100$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.34 CKSC_ATAUJS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJSCSID[2:0]	Source Clock Setting for C_AWO_TAUJ*[1] 000$_B$: Disabled 001$_B$: HS IntOSC (default) 010$_B$: MainOSC 011$_B$: LS IntOSC 100$_B$: PPLLCLK2 Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ATAUJS_ACT — C_AWO_TAUJ Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2108_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.35 CKSC_ATAUJS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJSACT[2:0]	Source clock for currently active C_AWO_TAUJ

(3) CKSC_ATAUJD_CTL — C_AWO_TAUJ Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2200$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.36 CKSC_ATAUJD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJDCSID[2:0]	Clock Divider Setting for C_AWO_TAUJ 000$_B$: Setting prohibited 001$_B$: CKSC_ATAUJS_CTL selection /1 (default) 010$_B$: CKSC_ATAUJS_CTL selection /2 011$_B$: CKSC_ATAUJS_CTL selection /4 100$_B$: CKSC_ATAUJS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ATAUJD_ACT — C_AWO_TAUJ Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2208_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.37 CKSC_ATAUJD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJDACT[2:0]	Clock divider for currently active C_AWO_TAUJ

(5) CKSC_ATAUJD_STPM — C_AWO_TAUJ Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2218_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJ DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.38 CKSC_ATAUJD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ATAUJDSTPMSK	0: Clock domain C_AWO_TAUJ is stopped in stand-by mode. 1: Clock domain C_AWO_TAUJ is not stopped in stand-by mode.

12AB.4.3.3 RTCA Clock Domain C_AWO_RTCA

(1) CKSC_ARTCAS_CTL — C_AWO_RTCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2300_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.39 CKSC_ARTCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ARTCASCSID[1:0]	Source Clock Setting for C_AWO_RTCA 00_B: Disabled (default) 01_B: SubOSC*[1] 10_B: MainOSC*[2] 11_B: LS IntOSC

Note 1. It isn't supported with 100-pin product.

Note 2. To avoid supplying a clock signal equal to or higher than 4 MHz to the C_AWO_RTCA clock domain, check that CKSC_ARTCAD_ACT = 0000 0000_H (disabled) before setting CKSC_ARTCAS_CTL to 10_B (MainOSC).

(2) CKSC_ARTCAS_ACT — C_AWO_RTCA Source Clock Active Register

This register is initialized by a power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2308$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.40 CKSC_ARTCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ARTCASACT[1:0]	Source clock for currently active C_AWO_RTCA

(3) CKSC_ARTCAD_CTL — C_AWO_RTCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2400_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.41 CKSC_ARTCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ARTCADCSID[2:0]	Clock Divider Setting for C_AWO_RTCA 000_B: Disabled (default) 001_B: CKSC_ARTCAS_CTL selection /1 010_B: CKSC_ARTCAS_CTL selection /2 011_B: CKSC_ARTCAS_CTL selection /4 100_B: CKSC_ARTCAS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ARTCAD_ACT — C_AWO_RTCA Clock Divider Active Register

This register is initialized by a power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2408_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.42 CKSC_ARTCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ARTCADACT[2:0]	Clock divider for currently active C_AWO_RTCA

(5) CKSC_ARTCAD_STPM — C_AWO_RTCA Stop Mask Register

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2418_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.43 CKSC_ARTCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARTCADSTPMSK	0: Clock domain C_AWO_RTCA is stopped in stand-by mode. 1: Clock domain C_AWO_RTCA is not stopped in stand-by mode.

12AB.4.3.4 ADCA0 Clock Domain C_AWO_ADCA

(1) CKSC_AADCAS_CTL — C_AWO_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2500$_{H}$

Value after reset: 0000 0001$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.44 CKSC_AADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCASCSID[1:0]	Source Clock Setting for C_AWO_ADCA*1 00$_{B}$: Disabled 01$_{B}$: HS IntOSC (default) 10$_{B}$: MainOSC 11$_{B}$: PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_AADCAS_ACT — C_AWO_ADCA Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2508$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.45 CKSC_AADCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCASACT[1:0]	Source clock for currently active C_AWO_ADCA

(3) CKSC_AADCAD_CTL — C_AWO_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2600$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.46 CKSC_AADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCADCSID[1:0]	Clock Divider Setting for C_AWO_ADCA 00$_B$: Setting prohibited 01$_B$: CKSC_AADCAS_CTL selection /1 (default) 10$_B$: CKSC_AADCAS_CTL selection /2*[1] 11$_B$: Setting prohibited

Note 1. Make sure that the frequency of CKSC_AADCA is no less than 8 MHz after division by 2.

(4) CKSC_AADCAD_ACT — C_AWO_ADCA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2608$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.47 CKSC_AADCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCADACT[1:0]	Clock divider for currently active for C_AWO_ADCA

(5) CKSC_AADCAD_STPM — C_AWO_ADCA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2618$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCA DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.48 CKSC_AADCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AADCADSTPMSK	0: Clock domain C_AWO_ADCA is stopped in stand-by mode. 1: Clock domain C_AWO_ADCA is not stopped in stand-by mode.

12AB.4.3.5 FOUT Clock Domain C_AWO_FOUT

(1) CKSC_AFOUTS_CTL — C_AWO_FOUT Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2700_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.49 CKSC_AFOUTS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	AFOUTSCSID[2:0]	Source Clock Setting for C_AWO_FOUT*[1] 000_B: Disabled (default) 001_B: MainOSC 010_B: HS IntOSC 011_B: LS IntOSC 100_B: SubOSC*[2] 101_B: PPLLCLK4 110_B: PPLLCLK4 111_B: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK4.

Note 2. It isn't supported with 100-pin product.

(2) CKSC_AFOUTS_ACT — C_AWO_FOUT Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2708_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.50 CKSC_AFOUTS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	AFOUTSACT[2:0]	Source clock for currently active C_AWO_FOUT

(3) CKSC_AFOUTS_STPM — C_AWO_FOUT Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2718_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.51 CKSC_AFOUTS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AFOUTSSTPMSK	0: Clock domain C_AWO_FOUT is stopped in stand-by mode. 1: Clock domain C_AWO_FOUT is not stopped in stand-by mode.

12AB.4.3.6 CPU Clock Domain C_ISO_CPUCLK

(1) CKSC_CPUCLKS_CTL — C_ISO_CPUCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 $A000_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.52 CKSC_CPUCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CPUCLKSCSID[1:0]	Source Clock Setting for C_ISO_CPUCLK 00_B: CPLL0OUT 01_B: EMCLK (default) 10_B: MainOSC 11_B: CPLL1OUT

CAUTION

The clock source selected for the C_ISO_CPUCLK clock domain should not be stopped by software.

For the setting procedure of this register, refer to **Section 12AB.5.4, CPUCLK Source and Divided Clock Selection Method**.

(2) CKSC_CPUCLKS_ACT — C_ISO_CPUCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A008$_{\text{H}}$

Value after reset: 0000 0001$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.53 CKSC_CPUCLKS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	CPUCLKSACT[1:0]	Source clock for currently active C_ISO_CPUCLK

(3) CKSC_CPUCLKD_CTL — C_ISO_CPUCLK Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 $A100_H$
Value after reset:	0000 0029_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CPUCLKDPLL[2:0]			CPUCLKDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.54 CKSC_CPUCLKD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 3	CPUCLKDPLL[2:0] *1, *2	Clock Divider CPLLDIV Setting Specifies the CPLLDIV divisor, which determines maximum clock frequency of C_ISO_CPUCLK. 000_B: CPLL0/1OUT = VCO0/1OUT × 1/6 (80 MHz) (PLL0/1) 010_B: CPLL0/1OUT = VCO0/1OUT × 1/4 (120 MHz) (PLL0/1) 110_B: CPLL0OUT = VCO0OUT × 1/3 (160 MHz) (PLL0) 101_B: CPLL0OUT = VCO0OUT × 1/2 (240 MHz) (PLL0) Other than above: Setting prohibited
2 to 0	CPUCLKDCSID[2:0]	Clock Divider Setting for C_ISO_CPUCLK 000_B: Setting prohibited 001_B: CKSC_CPUCLKS_CTL selection /1 (Default) 010_B: CKSC_CPUCLKS_CTL selection /2 011_B: CKSC_CPUCLKS_CTL selection /4 100_B: CKSC_CPUCLKS_CTL selection /8 Other than above: Setting prohibited

Note 1. Setting of CKSC_CPUCLKD_CTL.CPUCLKDPLL[2:0] is effective only to the PLL selected by CKSC_CPUCLKS_CTL.

Note 2. For the supported settings, refer to the **Table 12AB.18, PLL0 Output Table and Table 12AB.23, PLL1 Output Table**.

CAUTION

The clock source selected for the C_ISO_CPUCLK clock domain should not be stopped by software.

For the setting procedure of this register, refer to **Section 12AB.5.4, CPUCLK Source and Divided Clock Selection Method**.

(4) CKSC_CPUCLKD_ACT — C_ISO_CPUCLK Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A108$_H$

Value after reset: 0000 0029$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CPUCLKDPLLACT[2:0]			CPUCLKDACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.55 CKSC_CPUCLKD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5 to 3	CPUCLKDPLLACT [2:0]	Clock divider for currently active CPLLDIV.
2 to 0	CPUCLKDACT [2:0]	Clock divider for currently active C_ISO_CPUCLK

12AB.4.3.7 Peripheral Clock Domains C_ISO_PERI1 and C_ISO_PERI2

(1) CKSC_IPERI1S_CTL — C_ISO_PERI1 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A200$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.56 CKSC_IPERI1S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI1SCSID[1:0]	Source Clock Setting for C_ISO_PERI1 00_B: Disabled 01_B: PPLLCLK (default) 10_B: PPLLCLK 11_B: Setting prohibited

NOTE

When CKSCLK_IPERI1 is disabled, SFMA0 and SFMA0_MEM shall be set to access disabled, by the corresponding HFSGD01PROT0 and HFSGD01PROT1 registers.

(2) CKSC_IPERI1S_ACT — C_ISO_PERI1 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A208$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.57 CKSC_IPERI1S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI1SACT[1:0]	Source clock for currently active C_ISO_PERI1

(3) CKSC_IPERI2S_CTL — C_ISO_PERI2 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A300$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.58 CKSC_IPERI2S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI2SCSID[1:0]	Source Clock Setting for C_ISO_PERI2 00$_B$: Disabled 01$_B$: PPLLCLK2 (default) 10$_B$: PPLLCLK2 11$_B$: Setting prohibited

(4) CKSC_IPERI2S_ACT — C_ISO_PERI2 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A308$_{\text{H}}$
Value after reset:	0000 0001$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.59 CKSC_IPERI2S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI2SACT[1:0]	Source clock for currently active C_ISO_PERI2

12AB.4.3.8 RLIN Clock Domains C_ISO_LIN

(1) CKSC_ILINS_CTL — C_ISO_LIN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A400$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.60 CKSC_ILINS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ILINSCSID[2:0]	Source Clock Setting for C_ISO_LIN*[1] 000$_B$: Disabled 001$_B$: PPLLCLK2 (default) 010$_B$: MainOSC 011$_B$: PPLLCLK2 100$_B$: HS IntOSC Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ILINS_ACT — C_ISO_LIN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A408$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.61 CKSC_ILINS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ILINSACT[2:0]	Source clock for currently active C_ISO_LIN

(3) CKSC_ILIND_CTL — C_ISO_LIN Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A800$_{H}$
Value after reset:	0000 0001$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.62 CKSC_ILIND_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINDCSID[1:0]	Clock Divider Setting for C_ISO_LIN 00$_{B}$: Setting prohibited 01$_{B}$: CKSC_ILINS_CTL selection /1 (default) 10$_{B}$: CKSC_ILINS_CTL selection /4 11$_{B}$: CKSC_ILINS_CTL selection /8

NOTE

The setting of this register is only applicable to RLIN30. The settings 10$_{B}$ (CKSC_ILINS_CTL selection /4) and 11$_{B}$ (CKSC_ILINS_CTL selection /8) are only available in UART mode.

(4) CKSC_ILIND_ACT — C_ISO_LIN Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A808$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.63 CKSC_ILIND_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINDACT[1:0]	Clock divider for currently active C_ISO_LIN

(5) CKSC_ILIND_STPM — C_ISO_LIN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A818$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILIND STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.64 CKSC_ILIND_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ILINDSTPMSK	0: Clock domain C_ISO_LIN is stopped in stand-by mode. 1: Clock domain C_ISO_LIN is not stopped in stand-by mode.

12AB.4.3.9 ADCA1 Clock Domain C_ISO_ADCA

(1) CKSC_IADCAS_CTL — C_ISO_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A500$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.65 CKSC_IADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCASCSID[1:0]	Source Clock Setting for C_ISO_ADCA 00$_B$: Disabled 01$_B$: HS IntOSC (default) 10$_B$: MainOSC 11$_B$: PPLLCLK2

CAUTION

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 4 to 9.6” at CKDIVMD = 1[*1].

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 2 to 4.8” at CKDIVMD = 0[*1].

(1) Frequency [MHz] specified by the C_ISO_CPUCLK source clock selection register (CKSC_CPUCLKS_CTL) and C_ISO_CPUCLK clock divider selection register (CKSC_CPUCLKD_CTL)

(2) Frequency [MHz] specified by the C_ISO_ADCA source clock selection register (CKSC_IADCAS_CTL) and the C_ISO_ADCA clock divider selection register (CKSC_IADCAD_CTL)

Note 1. For details, see **Section 44, Flash Memory, 44.9.2, OPBT0 — Option Byte 0**.

(2) CKSC_IADCAS_ACT — C_ISO_ADCA Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 $A508_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.66 CKSC_IADCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCASACT[1:0]	Source clock for currently active C_ISO_ADCA

(3) CKSC_IADCAD_CTL — C_ISO_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A600$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.67 CKSC_IADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCADCSID[1:0]	Clock Divider Setting for C_ISO_ADCA 00$_B$: Setting prohibited 01$_B$: CKSC_IADCAS_CTL selection /1 (default) 10$_B$: CKSC_IADCAS_CTL selection /2*[1] 11$_B$: Setting prohibited

Note 1. Make sure that the frequency of CKSC_IADCA is no less than 8 MHz after division by 2.

CAUTION

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 4 to 9.6” at CKDIVMD = 1*[1].

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 2 to 4.8” at CKDIVMD = 0*[1].

(1) Frequency [MHz] specified by the C_ISO_CPUCLK source clock selection register (CKSC_CPUCLKS_CTL) and C_ISO_CPUCLK clock divider selection register (CKSC_CPUCLKD_CTL)

(2) Frequency [MHz] specified by the C_ISO_ADCA source clock selection register (CKSC_IADCAS_CTL) and the C_ISO_ADCA clock divider selection register (CKSC_IADCAD_CTL)

Note 1. For details, see **Section 44, Flash Memory**, **44.9.2, OPBT0 — Option Byte 0**.

(4) CKSC_IADCAD_ACT — C_ISO_ADCA Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 A608$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12B.68 CKSC_IADCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCADACT[1:0]	Clock divider for currently active C_ISO_ADCA

12AB.4.3.10 RS-CANFD Clock Domains C_ISO_CAN and C_ISO_CANOSC

(1) CKSC_ICANS_CTL — C_ISO_CAN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 $A900_H$

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.69 CKSC_ICANS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANSCSID[1:0]	Source Clock Setting for C_ISO_CAN*[1] 00_B: Disabled 01_B: MainOSC 10_B: PPLLCLK 11_B: PPLLCLK (default)

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK.

(2) CKSC_ICANS_ACT — C_ISO_CAN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A908$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.70 CKSC_ICANS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANSACT[1:0]	Source clock for currently active C_ISO_CAN

(3) CKSC_ICANS_STPM — C_ISO_CAN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A918$_H$
Value after reset:	0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANS STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.71 CKSC_ICANS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANSSTPMSK	0: Clock domain C_ISO_CAN is stopped in stand-by mode. 1: Clock domain C_ISO_CAN is not stopped in stand-by mode.

(4) CKSC_ICANOSCD_CTL — C_ISO_CANOSC Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCD CSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.72 CKSC_ICANOSCD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANOSCDCSID[1:0]	Clock Divider Setting for C_ISO_CANOSC*[1] 00$_B$: Disabled (default) 01$_B$: MainOSC/1 10$_B$: MainOSC/2 11$_B$: Setting prohibited

Note 1. Select MainOSC/2 when the source clock setting of C_ISO_CAN is MainOSC.

(5) CKSC_ICANOSCD_ACT — C_ISO_CANOSC Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AA08$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.73 CKSC_ICANOSCD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANOSCDACT[1:0]	Clock divider for currently active C_ISO_CANOSC

(6) CKSC_ICANOSCD_STPM — C_ISO_CANOSC Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA18$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12AB.74 CKSC_ICANOSCD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANOSCDSTPMSK	0: Clock domain C_ISO_CANOSC is stopped in stand-by mode. 1: Clock domain C_ISO_CANOSC is not stopped in stand-by mode.

12AB.4.3.11 CSI Clock Domain C_ISO_CSI

(1) CKSC_ICSIS_CTL — C_ISO_CSI Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AB00$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12AB.75 CKSC_ICSIS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ICSISCSID[2:0]	Source Clock Setting for C_ISO_CSI 000$_B$: Disabled 001$_B$: PPLLCLK (default) 010$_B$: PPLLCLK 011$_B$: MainOSC 100$_B$: HS IntOSC Other than above: Setting prohibited

(2) CKSC_ICSIS_ACT — C_ISO_CSI Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 AB08$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.76 CKSC_ICSIS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ICSISACT[2:0]	Source clock for currently active C_ISO_CSI

12AB.4.3.12 RIIC Clock Domain C_ISO_IIC

(1) CKSC_IIICS_CTL — C_ISO_IIC Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AC00$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12AB.77 CKSC_IIICS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IIICSCSID[1:0]	Source Clock Setting for C_ISO_IIC 00$_B$: Disabled 01$_B$: PPLLCLK2 (default) 10$_B$: PPLLCLK2 11$_B$: Setting prohibited

(2) CKSC_IIICS_ACT — C_ISO_IIC Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 AC08$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.78 CKSC_IIICS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IIICSACT[1:0]	Source clock for currently active C_ISO_IIC

12AB.5 Clock Domain Setting Method

12AB.5.1 Clock Domain Setting

12AB.5.1.1 Overview of Clock Selector Register

The clock selector for a clock domain C_AWO_<*name*>/C_ISO_<*name*> can be controlled by the following registers:

- Source clock selection registers
  These registers select the clock to be used as the domain clock from the available source clocks.
  - AWO source clock selection: CKSC_A<*name*>S_CTL
  - ISO source clock selection: CKSC_I<*name*>S_CTL
- Clock divider selection registers
  These registers specify the clock division ratio for the selected source clock.
  - AWO clock divider: CKSC_A<*name*>D_CTL
  - ISO clock divider: CKSC_I<*name*>D_CTL
- Source clock active registers and clock divider active registers
  These registers return the currently active source clock selection and division ratio, respectively.
  - AWO source clock active register/clock divider active register: CKSC_A<*name*>S_ACT/CKSC_A<*name*>D_ACT
  - ISO source clock active register/clock divider active register: CKSC_I<*name*>S_ACT/CKSC_I<*name*>D_ACT

NOTE

- Not all clock selectors provide all the control functions described above.
- The symbol “I”, which indicates the power domain, is not added to the names of registers within clock domain C_ISO_CPUCLK.

12AB.5.1.2 Setting Procedure for Clock Domain

Procedure of setting up clock domain is described as below:

1. Set up a source clock
   - Select a source clock. (CKSC_A<*name*>S_CTL, CKSC_I<*name*>S_CTL)
   - Confirm completion of selection. (CKSC_A<*name*>S_ACT, CKSC_I<*name*>S_ACT)*[1]
2. Setting a clock divider
   - Select a clock divider. (CKSC_A<*name*>D_CTL, CKSC_I<*name*>D_CTL)
   - Confirm completion of selection. (CKSC_A<*name*>D_ACT, CKSC_I<*name*>D_ACT)*[2]

Note 1. Continue processing after CKSC_A<*name*>S_ACT and CKSC_I<*name*>S_ACT are updated with the new values written to CKSC_A<*name*>S_CTL and CKSC_I<*name*>S_CTL.

Note 2. Continue processing after CKSC_A<*name*>D_ACT and CKSC_I<*name*>D_ACT are updated with the new values written to CKSC_A<*name*>D_CTL and CKSC_I<*name*>D_CTL.

CAUTION

The source clock to be selected must be operating before performing these settings.
The behavior and performance are not guaranteed if setup is performed while the source clock is stopped.
Access to a peripheral module is prohibited while the clock is not supplied to the module.

12AB.5.2 Stopping the Clock in Stand-by Mode

In stand-by mode (STOP mode, DeepSTOP mode, and Cyclic STOP mode), clock domain C_AWO_<*name*>/C_ISO_<*name*> can be configured to stop or continue its clock CKSCLK_A<*name*>/CKSCLK_I<*name*> in response to clock stop requests from the stand-by controller.

The clock stop mask registers are used to determine the operation status of the clock in stand-by mode:

- CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK = 0:
  The STOP request signal is not masked, so the domain clock CKSCLK_A<*name*>/CKSCLK_I<*name*> is stopped during stand-by mode.
  If the domain clock was in operation before transition to stand-by mode, it is automatically re-started after wake-up from stand-by mode.
  If there is another clock domain which the same source clock is selected and its stop mask setting is set to 1 (CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK=1), the source clock will continue operation in stand-by mode.
  The CPU clock domain C_ISO_CPUCLK is always stopped in stand-by mode.
- CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK = 1:
  The STOP request signal is masked, so CKSCLK_A<*name*>/CKSCLK_I<*name*> continues to operate during stand-by.
  The source clock selected for the target clock domain will also continue to operate in stand-by mode.
  Supply of a clock signal to the clock domains in the Isolated area (ISO area) will be stopped in DeepSTOP mode.

12AB.5.3 Clock Domain Settings

The following table shows a selectable source clock, a frequency division ratio, and a register to be used for each clock domain.

Table 12AB.79 List of Selectable Clocks (RH850/F1KH-D8)

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_AWO_WDTA	CKSCLK_AWDTA	—	**LS IntOSC**	CKSC_AWDTAD_CTL	1/1	240 kHz	WDTA0
					1/128		
C_AWO_TAUJ	CKSCLK_ATAUJ	CKSC_ATAUJS_CTL	MainOSC	CKSC_ATAUJD_CTL	**1/1**	40 MHz	TAUJ0
			HS IntOSC		1/2		TAUJ2
			LS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable		—		
C_AWO_RTCA	CKSCLK_ARTCA	CKSC_ARTCAS_CTL	MainOSC	CKSC_ARTCAD_CTL	1/1	4 MHz	RTCA0
			LS IntOSC		1/2		
			SubOSC		1/4		
			Disable		1/8		
			—		**Disable**		
C_AWO_ADCA	CKSCLK_AADCA	CKSC_AADCAS_CTL	MainOSC	CKSC_AADCAD_CTL	**1/1**	40 MHz	ADCA0
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_AWO_FOUT	CKSCLK_AFOUT	CKSC_AFOUTS_CTL	MainOSC	—	**1/1**	24 MHz	FOUT
			HS IntOSC				
			LS IntOSC				
			SubOSC				
			PPLLCLK4				
			Disable				
C_ISO_CPUCLK	CPUCLK	CKSC_CPUCLKS_CTL*3	MainOSC	CKSC_CPUCLKD_CTL	**1/1**	240 / 160 MHz*1	CPU subsystem
			CPLL0OUT (VCO0OUT × 1/2)		1/2		
			CPLL0OUT (VCO0OUT × 1/3)		1/4		
			CPLL0/1OUT (VCO0/1OUT × 1/4)		1/8		
			CPLL0/1OUT (VCO0/1OUT × 1/6)				
			EMCLK		—		
C_ISO_PERI1	CKSCLK_IPERI1	CKSC_IPERI1S_CTL	**PPLLCLK**	—	**1/1**	80 MHz	TAUD0
			Disable				TAUJ1
							TAUJ3
							ENCA0
							TAPA0
							PIC0
							SFMAn
C_ISO_PERI2	CKSCLK_IPERI2	CKSC_IPERI2S_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	TAUBn
			Disable				PWBAn
							PWGAn
							PWSAn
							RCFDCn (clkc)
							RSENTn
							MMCA0

Table 12AB.79 List of Selectable Clocks (RH850/F1KH-D8)

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_ISO_LIN	CKSCLK_ILIN	CKSC_ILINS_CTL	MainOSC	CKSC_ILIND_CTL*2	**1/1**	40 MHz	RLIN24n
			HS IntOSC		1/4		RLIN3n
			PPLLCLK2		1/8		
			Disable				
C_ISO_ADCA	CKSCLK_IADCA	CKSC_IADCAS_CTL	MainOSC	CKSC_IADCAD_CTL	**1/1**	40MHz	ADCA1
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	**1/1**	80 MHz	RCFDCn (PCLK)
			PPLLCLK				
			Disable				
C_ISO_CANOSC	CKSCLK_ICANOSC	—	**MainOSC**	CKSC_ICANOSCD_CTL	1/1	24 MHz	RCFDCn (clk_xincan)
					1/2		
					Disable		
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSIS_CTL	**PPLLCLK**	—	**1/1**	80 MHz	CSIGn
			MainOSC				CSIHn
			HS IntOSC				
			Disable				
C_ISO_IIC	CKSCLK_IIIC	CKSC_IIICS_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	RIICn
			Disable				

Note: The items written in bold are the initial setting clocks for each register.

Note 1. For the supported settings, refer to the **Table 12AB.18, PLL0 Output Table** and **Table 12AB.23, PLL1 Output Table**.

Note 2. The setting of this register only applies to RLIN30. The settings 1/4 and 1/8 are only available in UART mode.

Note 3. CKSC_CPUCLKS_CTL selects selection of Main OSC, CPLL0OUT, CPLL1OUT and EMCLK. CKSC_CPUCLKD_CTL.CPUCLKDPLL[2:0] selects clock frequency of CPLL0OUT or CPLL1OUT which is selected as a source clock of C_ISO_CPUCLK with CKSC_CPUCLKS_CTL.

Table 12AB.80 List of Selectable Clocks (RH850/F1KM-S4)

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_AWO_WDTA	CKSCLK_AWDTA	—	**LS IntOSC**	CKSC_AWDTAD_CTL	1/1	240 kHz	WDTA0
					1/128		
C_AWO_TAUJ	CKSCLK_ATAUJ	CKSC_ATAUJS_CTL	MainOSC	CKSC_ATAUJD_CTL	**1/1**	40 MHz	TAUJ0 TAUJ2
			HS IntOSC		1/2		
			LS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable		—		
C_AWO_RTCA	CKSCLK_ARTCA	CKSC_ARTCAS_CTL	MainOSC	CKSC_ARTCAD_CTL	1/1	4 MHz	RTCA0
			LS IntOSC		1/2		
			SubOSC*1		1/4		
			Disable		1/8		
			—		**Disable**		
C_AWO_ADCA	CKSCLK_AADCA	CKSC_AADCAS_CTL	MainOSC	CKSC_AADCAD_CTL	**1/1**	40 MHz	ADCA0
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_AWO_FOUT	CKSCLK_AFOUT	CKSC_AFOUTS_CTL	MainOSC	—	**1/1**	24 MHz	FOUT
			HS IntOSC				
			LS IntOSC				
			SubOSC*1				
			PPLLCLK4				
			Disable				
C_ISO_CPUCLK	CPUCLK	CKSC_CPUCLKS_CTL*4	MainOSC	CKSC_CPUCLKD_CTL	**1/1**	240 / 160 MHz*2	CPU subsystem
			CPLL0OUT (VCO0OUT × 1/2)		1/2		
			CPLL0OUT (VCO0OUT × 1/3)		1/4		
			CPLL0/1OUT (VCO0/1OUT × 1/4)		1/8		
			CPLL0/1OUT (VCO0/1OUT × 1/6)				
			EMCLK		—		
C_ISO_PERI1	CKSCLK_IPERI1	CKSC_IPERI1S_CTL	**PPLLCLK**	—	**1/1**	80 MHz	TAUD0 TAUJ1 TAUJ3 ENCA0 TAPA0 PIC0 SFMAn
			Disable				
C_ISO_PERI2	CKSCLK_IPERI2	CKSC_IPERI2S_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	TAUBn PWBAn PWGAn PWSAn RCFDCn (clkc) RSENTn
			Disable				
C_ISO_LIN	CKSCLK_ILIN	CKSC_ILINS_CTL	MainOSC	CKSC_ILIND_CTL*3	**1/1**	40 MHz	RLIN24n RLIN3n
			HS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable				

Table 12AB.80 List of Selectable Clocks (RH850/F1KM-S4)

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_ISO_ADCA	CKSCLK_IADCA	CKSC_IADCAS_CTL	MainOSC	CKSC_IADCAD_CTL	**1/1**	40MHz	ADCA1
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	**1/1**	80 MHz	RCFDCn (PCLK)
			PPLLCLK				
			Disable				
C_ISO_CANOSC	CKSCLK_ICANOSC	—	**MainOSC**	CKSC_ICANOSCD_CTL	1/1	24 MHz	RCFDCn (clk_xincan)
					1/2		
					Disable		
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSIS_CTL	**PPLLCLK**	—	**1/1**	80 MHz	CSIGn CSIHn
			MainOSC				
			HS IntOSC				
			Disable				
C_ISO_IIC	CKSCLK_IIIC	CKSC_IIICS_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	RIICn
			Disable				

Note: The items written in bold are the initial setting clocks for each register.

Note 1. It isn’t supported with 100-pin product.

Note 2. For the supported settings, refer to the **Table 12AB.18, PLL0 Output Table** and **Table 12AB.23, PLL1 Output Table**.

Note 3. The setting of this register only applies to RLIN30. The settings 1/4 and 1/8 are only available in UART mode.

Note 4. CKSC_CPUCLKS_CTL selects selection of Main OSC, CPLL0OUT, CPLL1OUT and EMCLK. CKSC_CPUCLKD_CTL.CPUCLKDPLL[2:0] selects clock frequency of CPLL0OUT or CPLL1OUT which is selected as a source clock of C_ISO_CPUCLK with CKSC_CPUCLKS_CTL.

CAUTION

To stop the clock source selected for the clock domain before transitioning to STOP/DeepSTOP mode, select “Disable” for that clock domain in advance. Do not stop the source clock of a clock domain for which “Disable” cannot be selected while functions are operating on that clock domain. To stop the clock source selected for the domain by transitioning to STOP/DeepSTOP mode, “Disable” does not need to be selected.
Instead of setting “Disable”, select “Stop” for the clock domain in stand-by mode by using the stop mask register.

12AB.5.4 CPUCLK Source and Divided Clock Selection Method

In case of changing CPUCLK frequency, follow the procedure below.

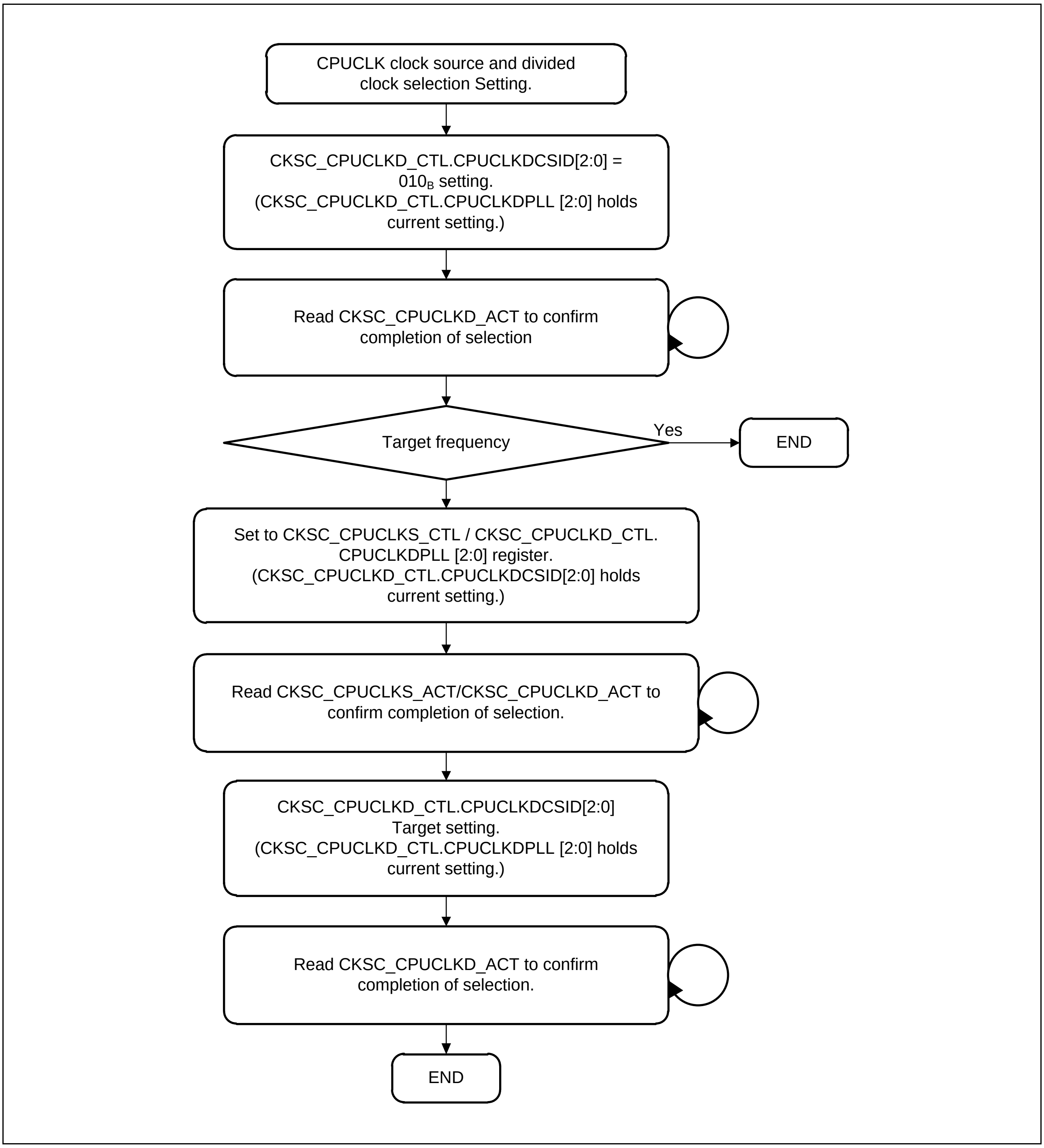


Figure 12AB.10 CPUCLK Source and Divided Clock Selection Method

CAUTION

In RH850/F1KH-D8, operation cannot be guaranteed unless the above procedure is conformed.

In RH850/F1KM-S4, the above procedure is optional.

12AB.5.5 CPUCLK Setting in STOP Mode Method

Transition before STOP mode, follow the procedure below.

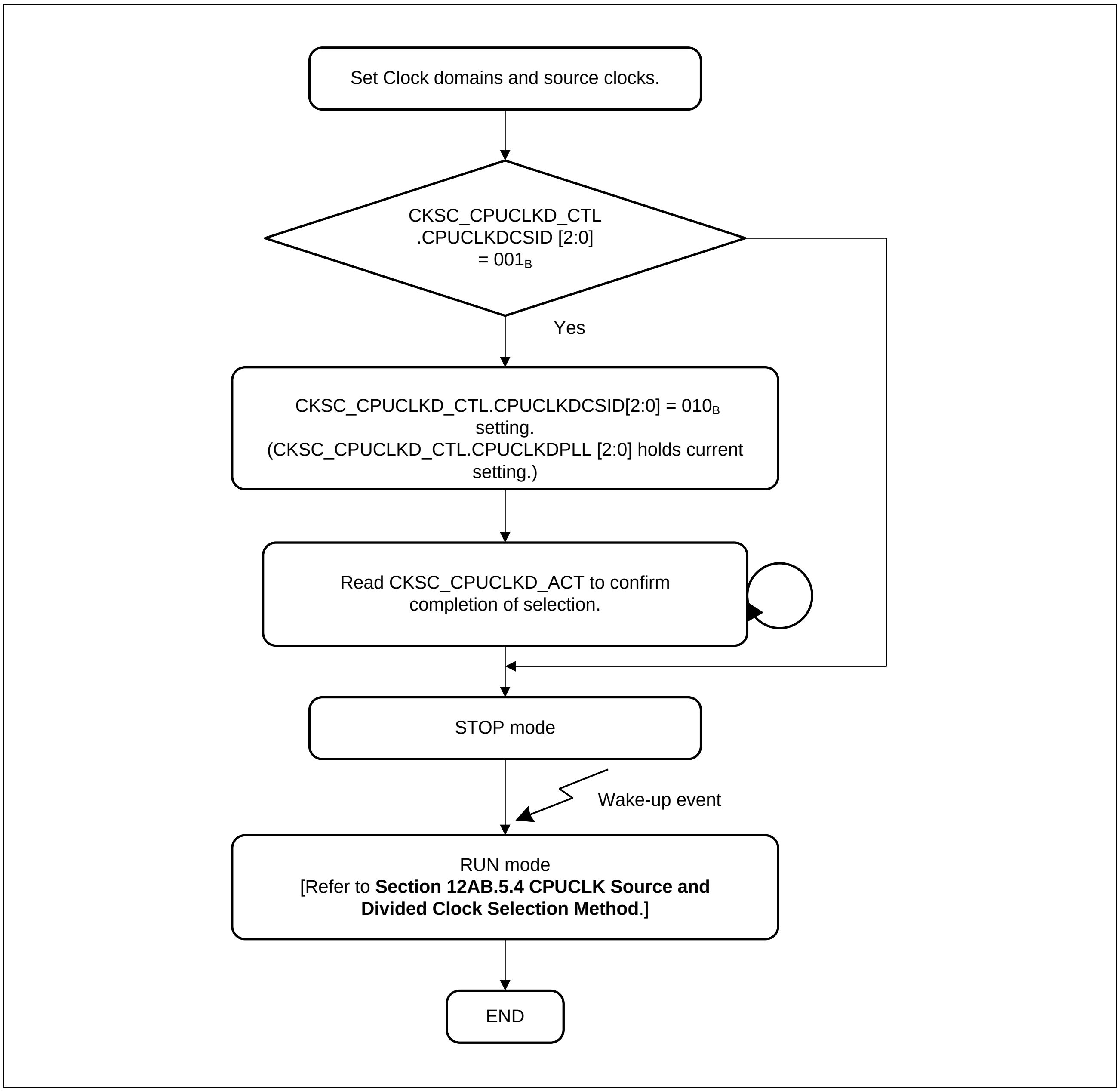


Figure 12AB.11 CPUCLK Setting in STOP Mode Method

CAUTION

In RH850/F1KH-D8, operation cannot be guaranteed unless the above procedure is conformed.

In RH850/F1KM-S4, the above procedure is optional.

12AB.6 Frequency Output Function (FOUT)

The frequency output function (FOUT) allows the clock to be output the clock as the external signal. Furthermore, the frequency can be divided by the clock divider before it is output.

12AB.6.1 Functional Overview

Figure 12AB.12, Frequency Output Function shows the configuration of the frequency output function.

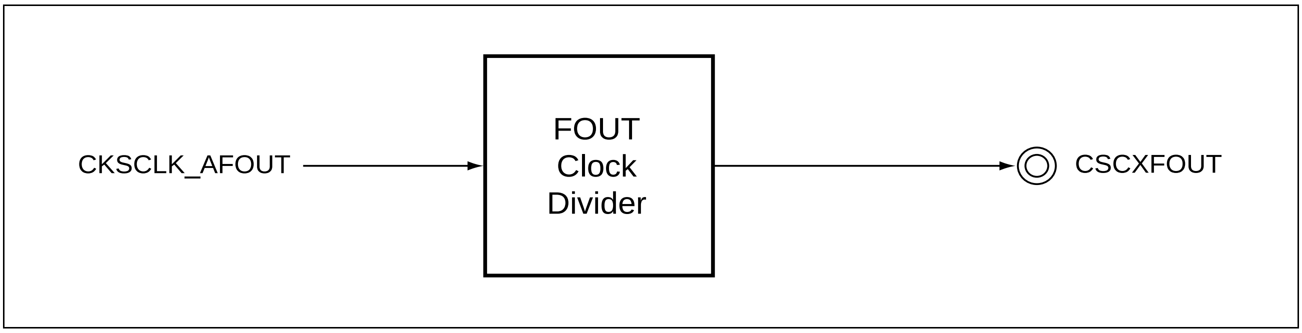


Figure 12AB.12 Frequency Output Function

The clock output function outputs the CKSCLK_AFOUT clock divided by 1 to 63 through the clock divider from CSCXFOUT. Division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register. Clock output frequency f_{CSCXFOUT} is expressed by the following equation.

$$f_{\mathrm{CSCXFOUT}} = (\text{CKSCLK\_AFOUT clock frequency}) / N$$

Clock output starts when, after CKSCLK_AFOUT is set and the clock output for the pin function is selected, division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register.

When a new division ratio is written to the FOUTDIV.FOUTDIV[5:0] bits, it becomes effective in synchronization with the CSCXFOUT output clock. Accordingly, the division ratio can be changed even while the CSCXFOUT clock is operating. The clock output is stopped by writing 000_H to the FOUTDIV[5:0] bits.

12AB.6.2 Clock Supply

The clock supply to the CSCXFOUT is shown in the following table.

Table 12AB.81 Clock Supply

Module	Unit Clock Name	Supply Clock Name	Description
CSCXFOUT	PCLK	CPUCLK_UL	Bus clock (Register access)
	CKSCLK_AFOUT	CKSCLK_AFOUT	Clock source of FOUT clock divider

12AB.6.3 Registers

12AB.6.3.1 List of Registers

The FOUT registers are listed in the following table.

Table 12AB.82 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	Clock division ratio register	FOUTDIV	FFF8 2800_H
	Clock divider status register	FOUTSTAT	FFF8 2804_H

12AB.6.3.2 FOUTDIV — Clock Division Ratio Register

This register defines the clock divisor.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2800$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FOUTDIV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12AB.83 FOUTDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FOUTDIV[5:0]	Clock Divider N 00$_H$: Clock output is stopped 01$_H$: N = 1 02$_H$: N = 2 : 3E$_H$: N = 62 3F$_H$: N = 63

12AB.6.3.3 FOUTSTAT — Clock Divider Status Register

This register indicates the clock output status.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2804_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTCLKACT	FOUTSYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12AB.84 FOUTSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUTCLKACT	Clock Divider Active 0: Frequency output is stopped. 1: Frequency output is ongoing.
0	FOUTSYNC	Clock Divider Synchronized 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

Section 12C Clock Controller of RH850/F1KM-S1

This section explains in general about the clock controller.

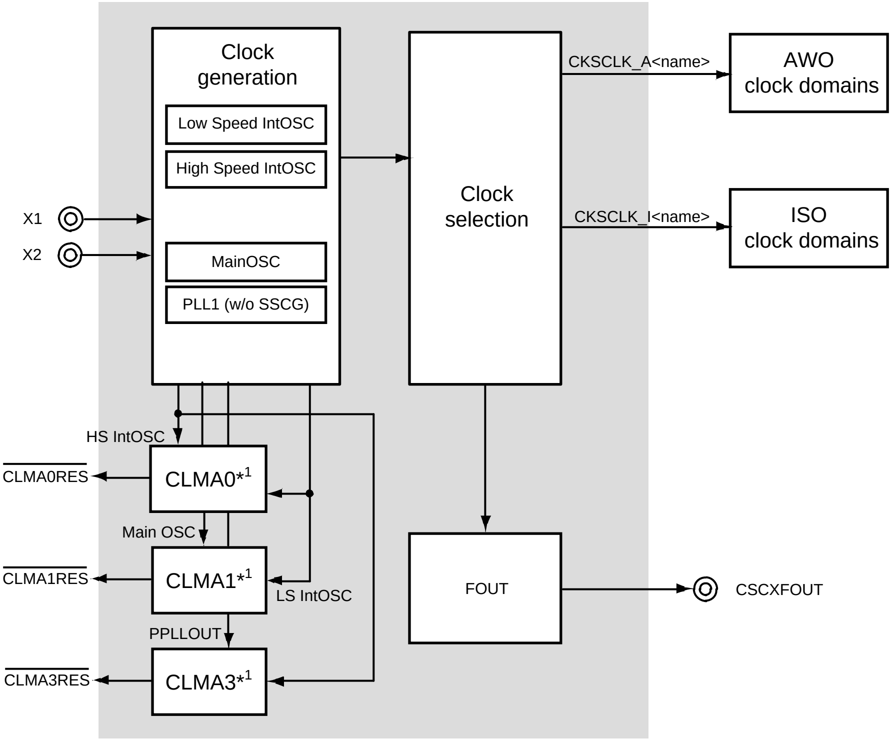
The first part in this section describes the specific features of the clock controller of the RH850/F1KM microcontrollers. The ensuing sections describe the clock oscillation circuit, clock selectors, and clock output function that make up the clock controller.

12C.1 Features of RH850/F1KM Clock Controller

The clock controller of the RH850/F1KM microcontrollers has the following features.

- Four on-chip clock oscillators
  - Main Oscillator (MainOSC) with an oscillation frequency of 8, 16, 20, and 24 MHz
  - High Speed Internal Oscillator (HS IntOSC) with a nominal frequency of 8 MHz (Typ.)
  - Low Speed Internal Oscillator (LS IntOSC) with a nominal frequency of 240 kHz (Typ.)
  - PLL without SSCG (PLL1)
- Fine management of clock supply to peripheral modules through clock domains
- On-chip clock monitor that detects clock anomalies when the Main Oscillator, High Speed Internal Oscillator, or PLL are in use. See **Section 13, Clock Monitor (CLMA)**.
- Clock output (FOUT)

Figure 12C.1, Clock Controller Overview shows the schematic diagram of the clock controller.



Note 1. See **Section 13, Clock Monitor (CLMA)** for details of clock monitors.

Figure 12C.1 Clock Controller Overview

12C.2 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is composed of clock oscillators and clock generation circuits that generate the clocks for the CPU and the peripheral modules, a clock selector for selecting the optimum clock, and clock domains for the CPU and the peripheral modules.

Figure 12C.2, Clock Controller Configuration shows the configuration of the clock controller.

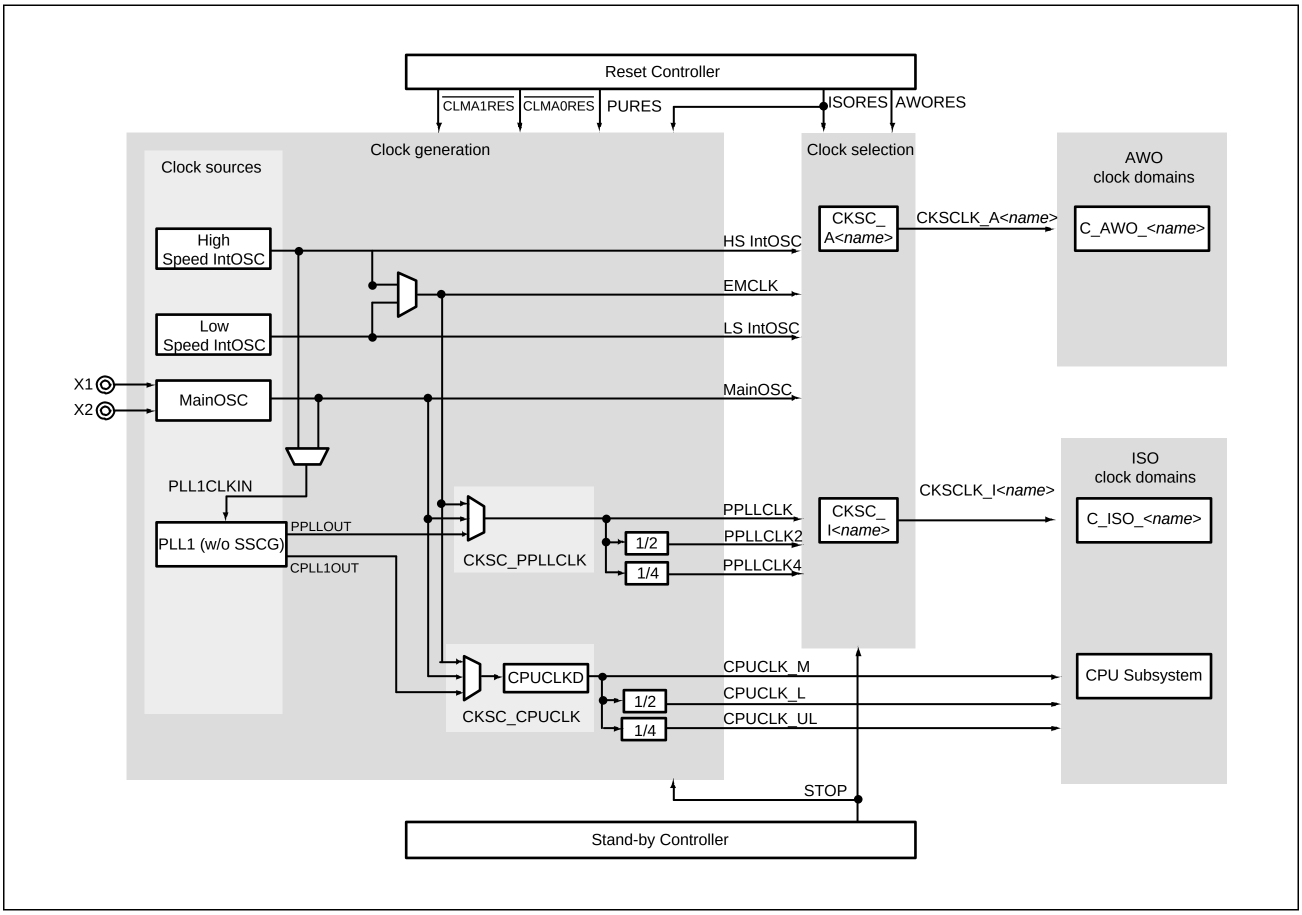


Figure 12C.2 Clock Controller Configuration

NOTE

Clock domain and clock control register naming conventions

The clock signals and their control registers, etc., described in this section are named according to the following naming conventions to reflect the power domain or clock domain to which they belong. The placeholder “<*name*>” is used to identify the target module in the clock domain:

- Clock domain names:
  - C_AWO_<*name*>: Always-On area (AWO area)*[1] clock domain
  - C_ISO_<*name*>: Isolated area (ISO area)*[1] clock domain
- Domain clock names:
  - CKSCLK_A<*name*>: Always-On area (AWO area) domain clock
  - CKSCLK_I<*name*>: Isolated area (ISO area) domain clock
- Clock selector names:
  - CKSC_A<*name*>: Always-On area (AWO area) clock selector
  - CKSC_I<*name*>: Isolated area (ISO area) clock selector
- Clock selector registers:
  - CKSC_A<*name*>S_CTL: Always-On area (AWO area) source clock selector register
  - CKSC_A<*name*>D_CTL: Always-On area (AWO area) source clock divider register
  - CKSC_I<*name*>S_CTL: Isolated area (ISO area) source clock selector register
  - CKSC_I<*name*>D_CTL: Isolated area (ISO area) source clock divider register

Example

The clock signal CKSCLK_AADCA (placeholder <*name*> = ADCA) is the clock supplied to the clock domain C_AWO_ADCA in the Always-On area (AWO area). This clock is selected by the clock selector register CKSC_AADCAS_CTL.

Note 1. Always-On area (AWO area) and Isolated area (ISO area) refer to the power supply domains. Always-On area (AWO area) is an always-on power supply, and Isolated area (ISO area) is an isolated power supply that is switched on or off by the operation mode.
For details, see **Section 10C, Power Supply Circuit of RH850/F1KM-S1**.

12C.2.1 Clock Generation Circuits

Four clock oscillators are provided:

Three clock oscillators are located on the Always-On area (AWO area) and PLL (PLL1) is located on the Isolated area (ISO area).

Main Oscillator (MainOSC)

The MainOSC generates the main clock X.
Generation of the clock X requires the connection of an external resonator to X1 and X2.
The clock X is used as the reference clock for the PLL.

High Speed Internal Oscillator (HS IntOSC)

The HS IntOSC generates the clock RH, which runs at a frequency of 8 MHz (Typ.).

Low Speed Internal Oscillator (LS IntOSC)

The LS IntOSC generates the clock RL, which runs at a frequency of 240 kHz (Typ.). It starts operation at power up and cannot be stopped, hence it is always operating.

PLL

The PLL circuits generate high speed operation clocks CPLL1OUT and PPLLOUT for normal operation of the microcontroller.

The clocks supplied by the clock oscillators (X, XT, RH, RL, CPLL1OUT, PPLLOUT) and their divided clocks (CPUCLK_M, CPUCLK_L, CPUCLK_UL, PPLLCLK, PPLLCLK2, and PPLLCLK4) are all generated in the clock generation circuit.

12C.2.2 Clock Selection

The clocks generated by the clock oscillators are input to the clock selectors CKSC_A<*name*>/CKSC_I<*name*>.

Domain clocks CKSCLK_A<*name*>/CKSCLK_I<*name*> are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

- CKSC_I<*name*>S_CTL/CKSC_I<*name*>D_CTL registers: determine the clock for the Isolated area (ISO area) clock domains.
- CKSC_A<*name*>S_CTL/CKSC_A<*name*>D_CTL registers: determine the clock for the Always-On area (AWO area) clock domains.

Note that not all available clocks generated by the clock oscillators are input to each clock selector.

The following clocks are supplied to the CPU and related modules from the clock generation circuit.

Emergency Clock (EMCLK)

The emergency clock EMCLK is supplied by the

- HS IntOSC, if it is active
- LS IntOSC, if the HS IntOSC is inactive

The selection is done automatically after CLMA0 reset is occurred, so if the HS IntOSC becomes lower than the limit for any reason, vital modules of the microcontroller are still in operation, since the LS IntOSC does not stop.

CPU Subsystem Clock (CPUCLK)

The CPU Subsystem clock CPUCLK is derived from PLL1 clock CPLL1OUT, MainOSC, and EMCLK. The CPU clock selector CKSC_CPUCLK incorporates the selector CPUCLKS, followed by the clock divider CPUCLKD.

The CPUCLK clock divider provides the frequency-divided CPUCLK_L clock signal and CPUCLK_UL clock signal derived from CPUCLK.

12C.2.3 Clock Domains

The clock controller allows selection of the respective clocks for the CPU and peripheral modules. The clock control scope is called the clock domain. For the correspondence between the CPU and peripheral modules and clock domains, see **Section 12C.5.3, Clock Domain Settings**.

12C.2.4 Resetting Clock Oscillators

The clock oscillators on the Always-On area (AWO area) are reset by the PURES signal.

The HS IntOSC is reset when $\overline{\text{CLMA0RES}}$ is generated and the MainOSC is reset when $\overline{\text{CLMA1RES}}$ is generated.

The clock oscillator on the Isolated area (ISO area) is reset by the ISORES signal.

For further details on the clock oscillators, see **Section 12C.3, Clock Oscillators**.

CAUTION

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, see **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.

12C.3 Clock Oscillators

12C.3.1 Main Oscillator (MainOSC)

The Main Oscillator generates the clock X. X is also used as the PLL1 input clock PLL1CLKIN.

Figure 12C.3, Main Oscillator (MainOSC) shows the basic configuration and signals of the MainOSC.

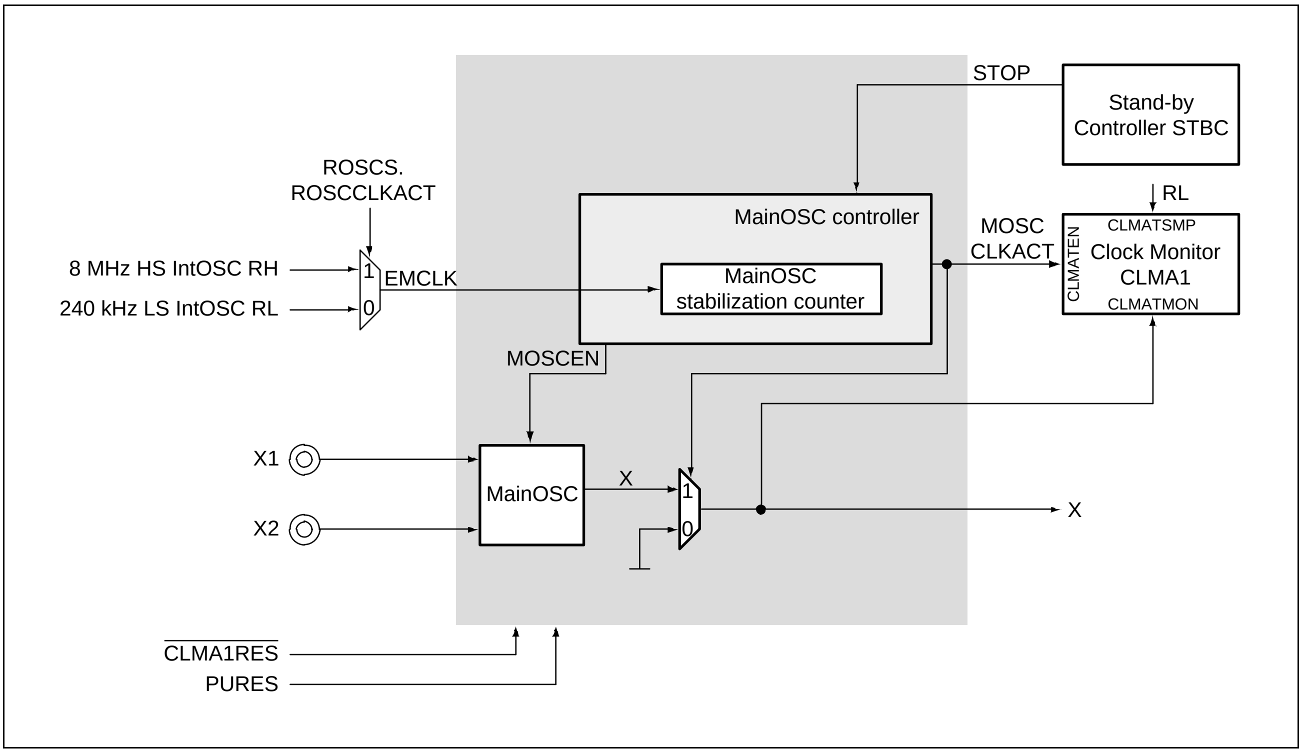


Figure 12C.3 Main Oscillator (MainOSC)

MainOSC

The MainOSC stops operating after reset is released. To use the MainOSC, set the MainOSC enable trigger bit (MOSCE.MOSCENTRG) to 1 to start the MainOSC.

MainOSC Stabilization

The MOSCST.MOSCCLKST[16:0] bits set the MainOSC oscillation stabilization time.
The MainOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. The oscillation stabilization time can be set to up to $2^{17} – 1$ EMCLK cycles.
As long as the MainOSC is not stable, the MOSCCLKACT signal disables the X output.
When the MainOSC stabilization counter reaches the value specified in MOSCST.MOSCCLKST[16:0], X is assumed to be stable and MOSCCLKACT switches from 0 to 1 to enable output of X when a waveform is output from MainOSC.

Stable and active X clock is indicated by MOSCS.MOSCCLKACT = 1.

MainOSC Amplification Gain

By using MOSCC.MOSCAMPSEL[1:0], the MainOSC’s input frequency, determined by the external resonator, can be selected from 8 MHz, 16 MHz, 20 MHz, and 24 MHz.

MainOSC STOP Requests in Stand-by Mode

The STOP signal from the Stand-by Controller requests the MainOSC Controller to switch off the X clock in stand-by modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).

The stop request mask bit MOSCSTPM.MOSCSTPMSK controls whether the MainOSC is stopped during stand-by or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0:
  The STOP request signal is not masked, so the MainOSC is stopped in stand-by.
  If the MainOSC is in operation before stand-by, it is automatically re-started after wake-up from stand-by, and the MainOSC stabilization counter counts the oscillation stabilization time.
  However, the STOP request is masked under the following conditions, even if MOSCSTPM.MOSCSTPMSK = 0. Therefore, the MainOSC will continue to operate even in stand-by mode.
  - If the stop mask is set (CKSC_xxxx_STPM = 0000 0003_H) for a clock domain for which the MainOSC is selected.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is masked, so the MainOSC continues to operate in stand-by.

Clock Monitor Control

The MainOSC activity signal MOSCCLKACT enables or disables supervision by the Clock Monitor CLMA1. In case the MainOSC is inactive (MOSCCLKACT = 0), supervision of its output clock X by CLMA1 is also disabled.

MainOSC Enable/Disable Trigger

The MainOSC can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger MOSCE.MOSCENTRG = 1 starts the MainOSC.
  Note that setting the enable trigger is only effective if the MainOSC is inactive, i.e. if MOSCS.MOSCCLKACT = 0.
- Disable trigger MOSCE.MOSCDISTRG = 1 stops the MainOSC.
  Note that setting the disable trigger is only effective if the MainOSC is active (MOSCCLKACT = 1) and the MainOSC stop requests are not masked (MOSCSTPM.MOSCSTPMSK = 0).

Direct Clock Input to X1 (EXCLK mode)

A clock waveform from an external clock source can be supplied to X1 pin. In this case, set the MOSCM bit of MOSCM register to 1 before clock input to X1 pin is supplied.

12C.3.2 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock RH. RH has a nominal frequency of 8 MHz.

Figure 12C.4, High Speed Internal Oscillator (HS IntOSC) shows the basic configuration and signals of the HS IntOSC.

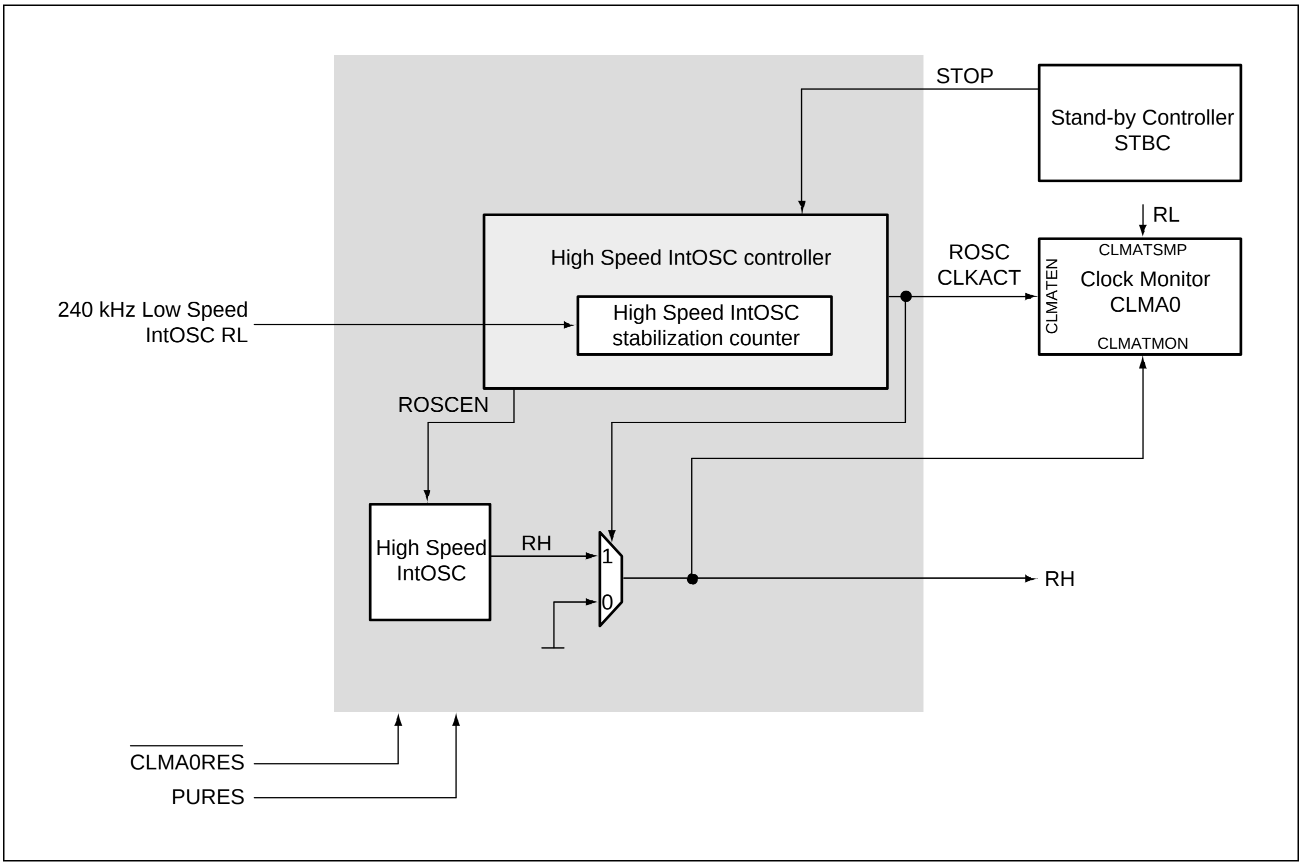


Figure 12C.4 High Speed Internal Oscillator (HS IntOSC)

After reset release the HS IntOSC starts operation.

NOTE

The HS IntOSC can neither be stopped nor started by software. It can only be stopped in stand-by mode. On the other hand, when CLMA0 is reset, the HS IntOSC can be enabled to stop by software.

HS IntOSC Stabilization

HS IntOSC outputs RH when it is stabilized.
Stable and active RH clock is indicated by ROSCS.ROSCCLKACT = 1.

HS IntOSC STOP Requests in Stand-by Mode

The STOP signal from the Stand-By Controller requests the HS IntOSC Controller to switch off the RH clock in stand-by modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).
The stop request mask bit ROSCSTPM.ROSCSTPMSK controls whether the HS IntOSC is stopped during stand-by or continues operation:

- ROSCSTPM.ROSCSTPMSK = 0:
  The STOP request signal is not masked, so the HS IntOSC is stopped during stand-by and automatically restarted after wake-up from stand-by.
  However, the STOP request is masked under the following conditions, even if ROSCSTPM.ROSCSTPMSK = 0. Therefore, the HS IntOSC will continue to operate even in stand-by mode.
  - If the stop mask is set (CKSC_xxxx_STPM = 0000 0003_H) for a clock domain for which the HS IntOSC is selected.
  - If the low power sampler (LPS) is operating
- ROSCSTPM.ROSCSTPMSK = 1:
  The STOP request signal is masked, so the HS IntOSC continues to operate during stand-by.

Clock Monitor Control

The HS IntOSC activity signal ROSCCLKACT enables or disables supervision by the Clock Monitor CLMA0. In case the HS IntOSC is inactive (ROSCCLKACT = 0), supervision of its output clock by CLMA0 is also deactivated.

The HS IntOSC clock RH is used as the sampling clock for Clock Monitor CLMA3.

HS IntOSC Disable Trigger

The disable trigger, ROSCE.ROSCDISTRG = 1 stops the HS IntOSC.

The setting of the disable trigger is enabled when HS IntOSC is active (ROSCS.ROSCCLKACT = 1) and HS IntOSC stop requests are not masked (ROSCSTPM.ROSCSTPMSK = 0).

HS IntOSC User Calibration Function

The HS IntOSC User trimming register (ROSCUT) enables adjustment of HS IntOSC frequency. The initial value of ROSCUT is preset value of the HS IntOSC trimming data. Overwrite the value with “read value + 1” or “read value – 1” until the HS IntOSC frequency reaches the target frequency range.

12C.3.3 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock RL. RL has a nominal frequency of 240 kHz.

Figure 12C.5, Low Speed Internal Oscillator (LS IntOSC) shows the basic configuration and signals of the LS IntOSC.

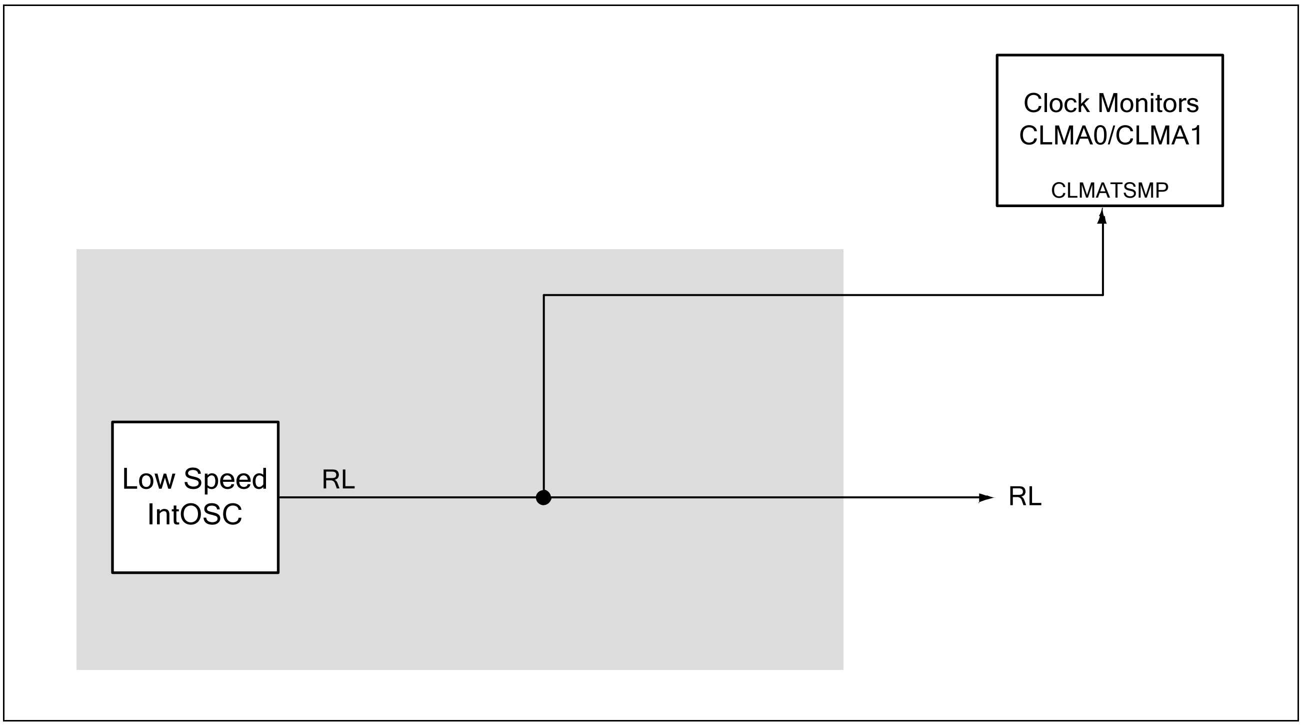


Figure 12C.5 Low Speed Internal Oscillator (LS IntOSC)

After reset release the LS IntOSC starts operation. It cannot be stopped.

The LS IntOSC clock RL is used as the sampling clock for the Clock Monitors CLMA0 and CLMA1.

12C.3.4 PLL

MainOSC or HS IntOSC is input to the phase-locked loops (PLL1) clock oscillator as PLL1CLKIN. The PLL1 output clocks CPLL1OUT and PPLLOUT serve as the main operation clocks for the microcontroller.

Figure 12C.6, PLL shows the basic configuration and signals of the PLL.

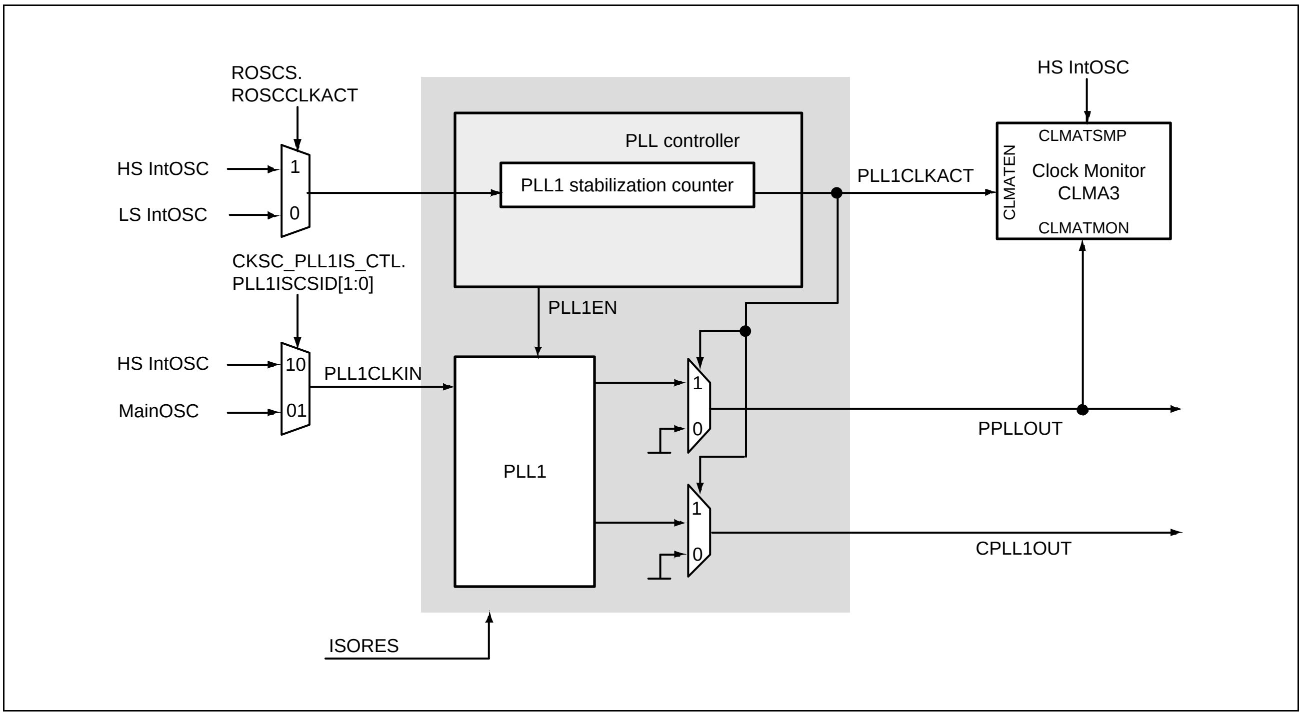


Figure 12C.6 PLL

PLL Enable

The PLL1 stops operating after reset is released. To use the PLL1, set the PLL1 enable trigger bit (PLL1E.PLL1ENTRG) to 1 to start the PLL1.

PLL Stabilization

The PLL1 stabilization counter starts counting the stabilization time, after PLL1 enabled.
As long as the PLL1 is not stable, the PLL1CLKACT signal disables the PPLLOUT and CPLL1OUT outputs.
When the PLL1 stabilization counter reaches the predefined value, PPLLOUT and CPLL1OUT are assumed to be stable and PLL1CLKACT switches from 0 to 1 to enable output of PPLLOUT and CPLL1OUT.

The stable and active state of the PPLLOUT and CPLL1OUT clocks is indicated by PLL1S.PLL1CLKACT = 1.

PLL in Stand-by Modes

In STOP mode, the PLL1 is automatically disabled and resumes operation after wake-up from STOP mode, if it was operating before entering STOP mode.

The PLL1 is also automatically disabled when transitioning to DeepSTOP mode. However, after restoring from DeepSTOP mode, the PLL1 needs to be reconfigured.

In Cyclic RUN and Cyclic STOP mode, the PLL1 is not available. Do not enable the PLL1 in Cyclic RUN mode.

Clock Monitor Control

The PLL1 activity signal PLL1CLKACT enables or disables supervision by the Clock Monitor CLMA3. In case the PLL1 is inactive (PLL1CLKACT = 0), supervision of the output clock PPLLOUT by CLMA3 is also deactivated.

PLL Enable/Disable Trigger

The PLL1 can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLL1E.PLL1ENTRG = 1 starts the PLL1
  Note that setting the enable trigger is only effective if the PLL1 is inactive, i.e. if PLL1S.PLL1CLKACT = 0.
- Disable trigger PLL1E.PLL1DISTRG = 1 stops the PLL1
  Note that setting the disable trigger is only effective if the PLL1 is active, i.e. if PLL1S.PLL1CLKACT = 1.

PLL Input Clock Selection

The PLL1 input clock (PLL1CLKIN) can be selected from MainOSC and HS IntOSC by using the CKSC_PLL1IS_CTL register.

The maximum frequency of CPLL1OUT and PPLLOUT is limited when the HS IntOSC is selected as PLL1 input clock.

12C.3.4.1 PLL1 Parameters

The PLL1 is configured by a set of parameters, loaded from the control register PLL1C and CKSC_CPUCLKD_CTL.

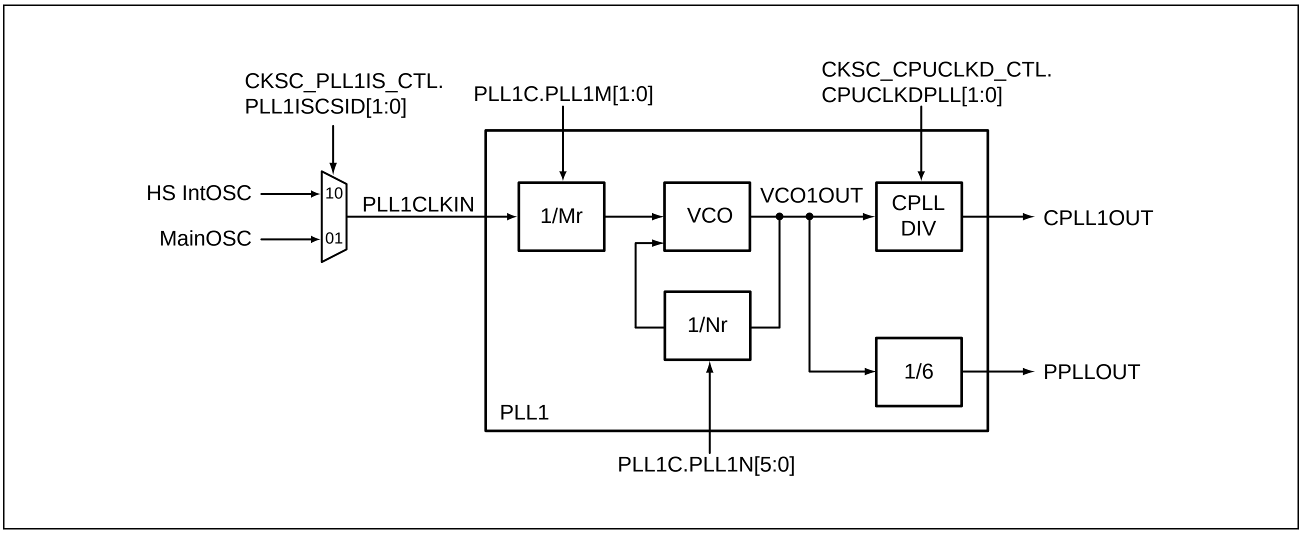


Figure 12C.7 PLL1 Circuit

CPLL1OUT and PPLLOUT

The PLL1 has two clock outputs “CPLL1OUT” and “PPLLOUT”. The CPLL1OUT is one of the clock sources of the CPU subsystem, and the PPLLOUT is one of the clock sources of the peripheral modules. CPLL1OUT and PPLLOUT shares the same clock source “VCO1OUT”, which is the output of the voltage controlled oscillator (VCO). The clock frequency of VCO1OUT is calculated by the following formula:

$f_{VCO1OUT} = f_{PLL1CLKIN} \times (Nr / Mr)$

The clock frequency of CPLL1OUT “$f_{CPLL1OUT}$” and that of PPLLOUT “$f_{PPLLOUT}$” are integer fractions of the VCO output frequency $f_{VCO1OUT}$. $f_{CPLL1OUT}$ and $f_{PPLLOUT}$ are calculated by the following formulas:

$f_{CPLL1OUT} = f_{VCO1OUT} \times 1/Pr = f_{PLL1CLKIN} \times (Nr / Mr) \times 1/Pr$

$f_{PPLLOUT} = f_{VCO1OUT} \times 1/6 = f_{PLL1CLKIN} \times (Nr / Mr) \times 1/6$

The values Nr and Mr are derived from PLL1C register bits:

Nr = PLL1C.PLL1N[5:0] + 1

Mr = PLL1C.PLL1M[1:0] + 1

The setting range of Mr: $1 \leq Mr \leq 3$

The value Pr is derived from CKSC_CPUCLKD_CTL. CPUCLKDPLL[1:0], and the value is 4, 5, or 6.

12C.4 Registers

12C.4.1 List of Registers

The registers of the clock controller are listed below.

Table 12C.1 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	Clock oscillator registers:		
	MainOSC enable register	MOSCE	FFF8 1100_H
	MainOSC status register	MOSCS	FFF8 1104_H
	MainOSC control register	MOSCC	FFF8 1108_H
	MainOSC stabilization time register	MOSCST	FFF8 $110C_H$
	MainOSC stop mask register	MOSCSTPM	FFF8 1118_H
	MainOSC mode control register	MOSCM	FFF8 $111C_H$
	HS IntOSC enable register	ROSCE	FFF8 1000_H
	HS IntOSC status register	ROSCS	FFF8 1004_H
	HS IntOSC stop mask register	ROSCSTPM	FFF8 1018_H
	HS IntOSC user trimming register	ROSCUT	FFF8 $101C_H$
	PLL1 enable register	PLL1E	FFF8 9000_H
	PLL1 status register	PLL1S	FFF8 9004_H
	PLL1 control register	PLL1C	FFF8 9008_H
	PLL1 input clock selection register	CKSC_PLL1IS_CTL	FFF8 $A700_H$
	PLL1 input clock active register	CKSC_PLL1IS_ACT	FFF8 $A708_H$
	PPLLCLK source clock selection register	CKSC_PPLLCLKS_CTL	FFF8 $A010_H$
	PPLLCLK source clock active register	CKSC_PPLLCLKS_ACT	FFF8 $A018_H$
	Clock selector control register:		
	C_AWO_WDTA clock divider selection register	CKSC_AWDTAD_CTL	FFF8 2000_H
	C_AWO_WDTA clock divider active register	CKSC_AWDTAD_ACT	FFF8 2008_H
	C_AWO_WDTA stop mask register	CKSC_AWDTAD_STPM	FFF8 2018_H
	C_AWO_TAUJ source clock selection register	CKSC_ATAUJS_CTL	FFF8 2100_H
	C_AWO_TAUJ source clock active register	CKSC_ATAUJS_ACT	FFF8 2108_H
	C_AWO_TAUJ clock divider selection register	CKSC_ATAUJD_CTL	FFF8 2200_H
	C_AWO_TAUJ clock divider active register	CKSC_ATAUJD_ACT	FFF8 2208_H
	C_AWO_TAUJ stop mask register	CKSC_ATAUJD_STPM	FFF8 2218_H
	C_AWO_RTCA source clock selection register	CKSC_ARTCAS_CTL	FFF8 2300_H
	C_AWO_RTCA source clock active register	CKSC_ARTCAS_ACT	FFF8 2308_H
	C_AWO_RTCA clock divider selection register	CKSC_ARTCAD_CTL	FFF8 2400_H
	C_AWO_RTCA clock divider active register	CKSC_ARTCAD_ACT	FFF8 2408_H
	C_AWO_RTCA stop mask register	CKSC_ARTCAD_STPM	FFF8 2418_H
	C_AWO_ADCA source clock selection register	CKSC_AADCAS_CTL	FFF8 2500_H
	C_AWO_ADCA source clock active register	CKSC_AADCAS_ACT	FFF8 2508_H
	C_AWO_ADCA clock divider selection register	CKSC_AADCAD_CTL	FFF8 2600_H
	C_AWO_ADCA clock divider active register	CKSC_AADCAD_ACT	FFF8 2608_H
	C_AWO_ADCA stop mask register	CKSC_AADCAD_STPM	FFF8 2618_H
	C_AWO_FOUT source clock selection register	CKSC_AFOUTS_CTL	FFF8 2700_H
	C_AWO_FOUT source clock active register	CKSC_AFOUTS_ACT	FFF8 2708_H
	C_AWO_FOUT stop mask register	CKSC_AFOUTS_STPM	FFF8 2718_H

Table 12C.1 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	C_ISO_CPUCLK source clock selection register	CKSC_CPUCLKS_CTL	FFF8 A000$_H$
	C_ISO_CPUCLK source clock active register	CKSC_CPUCLKS_ACT	FFF8 A008$_H$
	C_ISO_CPUCLK clock divider selection register	CKSC_CPUCLKD_CTL	FFF8 A100$_H$
	C_ISO_CPUCLK clock divider active register	CKSC_CPUCLKD_ACT	FFF8 A108$_H$
	C_ISO_PERI1 source clock selection register	CKSC_IPERI1S_CTL	FFF8 A200$_H$
	C_ISO_PERI1 source clock active register	CKSC_IPERI1S_ACT	FFF8 A208$_H$
	C_ISO_PERI2 source clock selection register	CKSC_IPERI2S_CTL	FFF8 A300$_H$
	C_ISO_PERI2 source clock active register	CKSC_IPERI2S_ACT	FFF8 A308$_H$
	C_ISO_LIN source clock selection register	CKSC_ILINS_CTL	FFF8 A400$_H$
	C_ISO_LIN source clock active register	CKSC_ILINS_ACT	FFF8 A408$_H$
	C_ISO_LIN clock divider selection register	CKSC_ILIND_CTL	FFF8 A800$_H$
	C_ISO_LIN clock divider active register	CKSC_ILIND_ACT	FFF8 A808$_H$
	C_ISO_LIN stop mask register	CKSC_ILIND_STPM	FFF8 A818$_H$
	C_ISO_CAN source clock selection register	CKSC_ICANS_CTL	FFF8 A900$_H$
	C_ISO_CAN source clock active register	CKSC_ICANS_ACT	FFF8 A908$_H$
	C_ISO_CAN stop mask register	CKSC_ICANS_STPM	FFF8 A918$_H$
	C_ISO_CANOSC clock divider selection register	CKSC_ICANOSCD_CTL	FFF8 AA00$_H$
	C_ISO_CANOSC clock divider active register	CKSC_ICANOSCD_ACT	FFF8 AA08$_H$
	C_ISO_CANOSC stop mask register	CKSC_ICANOSCD_STPM	FFF8 AA18$_H$
	C_ISO_CSI source clock selection register	CKSC_ICSIS_CTL	FFF8 AB00$_H$
	C_ISO_CSI source clock active register	CKSC_ICSIS_ACT	FFF8 AB08$_H$
	C_ISO_IIC source clock selection register	CKSC_IIICS_CTL	FFF8 AC00$_H$
	C_ISO_IIC source clock active register	CKSC_IIICS_ACT	FFF8 AC08$_H$

12C.4.2 Clock Oscillator Registers

12C.4.2.1 MOSCE — MainOSC Enable Register

This register is used to start and stop the MainOSC.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1100$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCDISTRG	MOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.2 MOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MOSCDISTRG	MainOSC Disable Trigger*1,*3 MOSCSTPM.MOSCSTPMSK = 0 0: No function 1: Stops MainOSC MOSCSTPM.MOSCSTPMSK = 1 When writing this bit while MOSCSTPM.MOSCSTPMSK = 1, the write value should be 0. 0: No function 1: Setting Prohibited This bit is automatically cleared to 0 by hardware after MainOSC is disabled.
0	MOSCENTRG	MainOSC Enable Trigger*2,*3 0: No function 1: Starts MainOSC This bit is automatically cleared to 0 by hardware after MainOSC is enabled.

Note 1. Follow the procedure given below for stopping the MainOSC by using MOSCDISTRG.

1. Confirm that the MainOSC is neither waiting for being enabled nor stopped (both MOSCE.MOSCDISTRG = 0 and MOSCE.MOSCENTRG = 0).
2. Confirm that the MainOSC is active (MOSCS.MOSCCLKACT = 1).
3. Check that there is no clock domain for which the MainOSC is selected. If the MainOSC is selected for a clock domain, disable the setting or select a clock source other than MainOSC.
4. Confirm that the MainOSC stop mask register (MOSCSTPM) is NOT set to “MainOSC continues operation in standby mode” (MOSCSTPM.MOSCSTPMSK = 1). Otherwise, set the register to “MainOSC stops operation in stand-by mode” (MOSCSTPM.MOSCSTPMSK = 0).
5. Stop the MainOSC (MOSCE.MOSCDISTRG = 1).
6. Confirm that the MainOSC has been stopped (MOSCS.MOSCCLKACT = 0).

Note 2. Follow the procedure given below for starting the MainOSC by using MOSCENTRG.

1. Confirm that the MainOSC is neither waiting for being enabled nor stopped (both MOSCE.MOSCDISTRG = 0 and MOSCE.MOSCENTRG = 0).
2. Confirm that the MainOSC is inactive (MOSCS.MOSCCLKACT = 0).
3. Start the MainOSC (MOSCE.MOSCENTRG = 1).
4. Confirm that the MainOSC has been started (MOSCS.MOSCCLKACT = 1).

Note 3. Starting and stopping the MainOSC at the same time by using the start and stop bits, i.e., by setting MOSCE.MOSCENTRG = 1 and MOSCE.MOSCDISTRG = 1 at the same time is not allowed.

12C.4.2.2 MOSCS — MainOSC Status Register

This register provides active status information about the MainOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 1104$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1^{*1}	0^{*1}
	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12C.3 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	MOSCCLKACT	MainOSC Active Status 0: MainOSC is inactive 1: MainOSC is active
1, 0	Reserved	When read, an undefined value is returned.

12C.4.2.3 MOSCC — MainOSC Control Register

This register is used to specify amplification gain of the MainOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1108_H

Value after reset: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCAMPSEL [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.4 MOSCC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MOSCAMPSEL[1:0]	MainOSC Amplification Gain Selection 00: Amplification gain for fx = 24 MHz 01: Amplification gain for fx = 20 MHz 10: Amplification gain for fx = 16 MHz 11: Amplification gain for fx = 8 MHz

CAUTION

Set this register when MainOSC is stopped.

12C.4.2.4 MOSCST — MainOSC Stabilization Time Register

This register determines the MainOSC stabilization time.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 110C$_H$

Value after reset: 0000 44C0$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKST16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSCCLKST[15:0]															
Value after reset	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12C.5 MOSCST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16 to 0	MOSCCLKST[16:0]	The MOSCCLKST[16:0] bits specify the count value for the MainOSC stabilization counter. ● If HS IntOSC active (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f_{RH} ● If HS IntOSC inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = MOSCCLKST[16:0] / f_{RL}

NOTE

See **Section 47C, Electrical Characteristics of RH850/F1KM-S1** for information about the MainOSC stabilization time.

CAUTION

Set this register when MainOSC is stopped.

12C.4.2.5 MOSCSTPM — MainOSC Stop Mask Register

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1118_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12C.6 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCSTPMSK	MainOSC Stop Request Mask 0: MainOSC stops operation in stand-by mode. MainOSC stops operation in the case the MainOSC disable trigger MOSCE.MOSCDISTRG is set to 1. 1: MainOSC continues operation in stand-by mode.

12C.4.2.6 MOSCM — MainOSC Mode Control Register

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA1RES}}$.

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 111C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12C.7 MOSCM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCM	MainOSC Mode control 0: OSC mode. (default) 1: EXCLK mode. MainOSC amplifier is disabled.

CAUTION

Set this register when MainOSC is stopped.

NOTE

EXCLK mode is a mode to directly input clock to X1. For details, see **Section 12C.3.1, Main Oscillator (MainOSC)**.

12C.4.2.7 ROSCE — HS IntOSC Enable Register

This register is used to stop the HS IntOSC operation.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

CAUTION

Set the ROSCE.ROSCDISTRG bit only when the $\overline{\text{CLMA0RES}}$ has occurred. In other cases, setting this bit is prohibited.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCDISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 12C.8 ROSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ROSCDISTRG	HS IntOSC Disable Trigger ROSCSTPM.ROSCSTPMSK = 0 0: No function 1: Stops HS IntOSC ROSCSTPM.ROSCSTPMSK = 1 Setting prohibited
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

12C.4.2.8 ROSCS — HS IntOSC Status Register

This register provides active status information about the HS IntOSC.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 1004$_H$
Value after reset:	0000 0007$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1[*1]	0[*1]
	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12C.9 ROSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	ROSCCLKACT	HS IntOSC Active Status 0: HS IntOSC is inactive 1: HS IntOSC is active
1, 0	Reserved	When read, an undefined value is returned.

12C.4.2.9 ROSCSTPM — HS IntOSC Stop Mask Register

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1018$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12C.10 ROSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ROSCSTPMSK	HS IntOSC Stop Request Mask 0: HS IntOSC stops operation in stand-by mode 1: HS IntOSC continues operation in stand-by mode Do not set the HS IntOSC disable trigger ROSCE.ROSCDISTRG to 1 while ROSCSTPMSK bit is set to 1.

12C.4.2.10 **PLL1E — PLL1 Enable Register**

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 9000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1DISTRG	PLL1ENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.11 PLL1E Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLL1DISTRG	PLL1 Disable Trigger[*1,*4] 0: No function 1: Stops PLL1 This bit is automatically cleared to 0 by hardware after PLL1 is disabled.
0	PLL1ENTRG	PLL1 Enable Trigger[*2,*3,*4] 0: No function 1: Starts PLL1 This bit is automatically cleared to 0 by hardware after PLL1 is activated.

Note 1. Follow the procedure given below for stopping the PLL1 by using PLL1DISTRG.

1. Confirm that the PLL1 is neither waiting for being enabled nor stopped (both PLL1E.PLL1DISTRG = 0 and PLL1E.PLL1ENTRG = 0).
2. Confirm that the PLL1 is active (PLL1S.PLL1CLKACT = 1).
3. Check that there is no clock domain for which the PLL1 is selected. If the PLL1 is selected for a clock domain, disable the setting or select a clock source other than the PLL1.
4. Stop the PLL1 (PLL1E.PLL1DISTRG = 1).
5. Confirm that the PLL1 has been stopped (PLL1S.PLL1CLKACT = 0).

Note 2. Before starting PLL1 using PLL1ENTRG, confirm that the PLL1 input clock (MainOSC or HS IntOSC, selected by the CKSC_PLL1IS_CTL) is operating.

Note 3. Follow the procedure given below for starting the PLL1 by using PLL1ENTRG.

1. Confirm that the PLL1 is neither waiting for being enabled nor stopped (both PLL1E.PLL1DISTRG = 0 and PLL1E.PLL1ENTRG = 0).
2. Confirm that the PLL1 is inactive (PLL1S.PLL1CLKACT = 0).
3. Start the PLL1 (PLL1E.PLL1ENTRG = 1).
4. Confirm that the PLL1 has been started (PLL1S.PLL1CLKACT = 1).

Note 4. Starting and stopping the PLL1 at the same time by using the start and stop bits, i.e., by setting PLL1E.PLL1ENTRG = 1 and PLL1E.PLL1DISTRG = 1 at the same time is not allowed.

12C.4.2.11 PLL1S — PLL1 Status Register

This register provides active status information about the PLL1.

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 9004$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1[*1]	0[*1]
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1CLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12C.12 PLL1S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLL1CLKACT	PLL1 Active Status 0: PLL1 is inactive 1: PLL1 is active
1, 0	Reserved	When read, an undefined value is returned.

12C.4.2.12 PLL1C — PLL1 Control Register

This register is used to set the PLL1 VCO output clock frequency $f_{VCO1OUT}$, shown in **Section 12C.3.4.1, PLL1 Parameters**.

This register can only be written, if the PLL1 is disabled.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 9008$_{H}$
Value after reset:	0001 133B$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PLL1M[1:0]		—	—	—	—	—	PLL1N[5:0]					
Value after reset	0	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12C.13 PLL1C Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12, 11	PLL1M[1:0]	Division ratio Mr is set. For PLL1M[1:0] settings, see **Table 12C.14, PLL1 Output Table**.
10 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	PLL1N[5:0]	Division ratio Nr is set. For PLL1N[5:0] settings, see **Table 12C.14, PLL1 Output Table**.

CAUTION

Set this register when PLL1 is stopped.

Table 12C.14 PLL1 Output Table

PLL1CLKIN frequency $f_{PLL1CLKIN}$ (MHz)	PLL1C. PLL1M[1:0] (Mr)*5	PLL1C. PLL1N[5:0] (Nr)*5	VCO1OUT frequency $f_{VCO1OUT}$ (MHz)	CPLL1OUT frequency $f_{CPLL1OUT}$ (MHz)*1			PPLLOUT frequency $f_{PPLLOUT}$ (MHz)
				VCO1OUT × 1/4	VCO1OUT × 1/5	VCO1OUT × 1/6	
8 (MainOSC)	00_B (Mr = 1)	$3B_H$ (Nr = 60)	480.0	120.0	96.0	80.0	80.0
16 (MainOSC)	01_B (Mr = 2)	$3B_H$ (Nr = 60)	480.0	120.0	96.0	80.0	80.0
20 (MainOSC)	01_B (Mr = 2)	$2F_H$ (Nr = 48)	480.0	120.0	96.0	80.0	80.0
24 (MainOSC)	01_B (Mr = 2)	27_H (Nr = 40)	480.0	120.0	96.0	80.0	80.0
	10_B (Mr = 3)	$3B_H$ (Nr = 60)	480.0	120.0	96.0	80.0	80.0
8 (HS IntOSC)*2, *4	00_B (Mr = 1)	$3B_H$ (Nr = 60)	480.0	N/A	N/A	80.0*2, *3	80.0*2

Note 1. The CPLL1OUT frequency is defined by CKSC_CPUCLKD_CTL.CPUCLKDPLL[1:0]. Refer to the CKSC_CPUCLKD_CTL register description.

Note 2. Typical frequencies. User calibration of HS IntOSC is required before setting HS IntOSC as PLL1CLKIN.

Note 3. The limit of CPLL1OUT frequency is 80 MHz (typ.) when HS IntOSC is selected as clock source of PLL1.

Note 4. See **Section 44.10, Usage Notes**.

Note 5. Settings other than those shown in this table are prohibited.

12C.4.2.13 PLL1 Input Clock Selection

(1) CKSC_PLL1IS_CTL — PLL1 Input Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A700$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1ISCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.15 CKSC_PLL1IS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PLL1ISCSID[1:0]	Source Clock Setting for PLL1 input clock 01$_B$: MainOSC (Default) 10$_B$: HS IntOSC^{*1} Other than above: Setting prohibited

Note 1. The maximum frequency of CPLL1OUT and PPLLOUT is limited when the HS IntOSC is selected as the PLL1 input clock.

CAUTION

Set this register when PLL1 is stopped.

(2) CKSC_PLL1IS_ACT — PLL1 Input Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A708$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1ISACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.16 CKSC_PLL1IS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	PLL1ISACT[1:0]	Source clock for currently active PLL1 input clock

12C.4.2.14 PPLLCLK Source Clock Selection

(1) CKSC_PPLLCLKS_CTL — PPLLCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A010_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPLLCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.17 CKSC_PPLLCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PPLLCLKSCSID[1:0]	Source Clock Setting for PPLLCLK 00_B: Setting prohibited 01_B: EMCLK (default) 10_B: MainOSC 11_B: PPLLOUT

(2) CKSC_PPLLCLKS_ACT — PPLLCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A018$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPLLCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.18 CKSC_PPLLCLKS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	PPLLCLKSACT[1:0]	Source clock for currently active PPLLCLK*[1]

Note 1. The data read from this register is 00$_B$ if the selected source clock for all of the following clock domains is other than PPLLCLK (or PPLLCLK2):
C_ISO_PERI1, C_ISO_PERI2, C_ISO_LIN, C_ISO_CAN, C_ISO_CSI, C_ISO_IIC

12C.4.2.15 ROSCUT — HS IntOSC User Trimming Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES and $\overline{\text{CLMA0RES}}$.

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 $101C_H$
Value after reset:	0000 $0XXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FADJUST[8:0]								
Value after reset	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12C.19 ROSCUT Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	FADJUST[8:0]	Frequency Adjustment parameters of HS IntOSC. The value after reset of this register is pre-set value of HS IntOSC trimming data. Overwrite this register with “read value + 1” or “read value – 1” until the HS IntOSC frequency becomes target frequency range.

12C.4.3 Clock Selector Control Register

12C.4.3.1 WDTA0 Clock Domain C_AWO_WDTA

(1) CKSC_AWDTAD_CTL — C_AWO_WDTA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2000_H
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.20 CKSC_AWDTAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AWDTADCSID[1:0]	Clock Divider Setting for C_AWO_WDTA 00_B: Setting prohibited 01_B: LS IntOSC / 128 (default) 10_B: LS IntOSC / 1 11_B: Setting prohibited

(2) CKSC_AWDTAD_ACT — C_AWO_WDTA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2008_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.21 CKSC_AWDTAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AWDTADACT[1:0]	Clock divider for currently active C_AWO_WDTA

(3) CKSC_AWDTAD_STPM — C_AWO_WDTA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2018_H
Value after reset:	0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.22 CKSC_AWDTAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AWDTADSTPMSK*1	0: Clock domain C_AWO_WDTA is stopped in stand-by mode. 1: Clock domain C_AWO_WDTA is not stopped in stand-by mode.

Note 1. The return time from stand-by mode can be shortened by setting this bit to 1.

12C.4.3.2 TAUJ Clock Domain C_AWO_TAUJ

(1) CKSC_ATAUJS_CTL — C_AWO_TAUJ Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2100_H
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.23 CKSC_ATAUJS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJSCSID[2:0]	Source Clock Setting for C_AWO_TAUJ*1 000_B: Disabled 001_B: HS IntOSC (default) 010_B: MainOSC 011_B: LS IntOSC 100_B: PPLLCLK2 Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ATAUJS_ACT — C_AWO_TAUJ Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2108_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.24 CKSC_ATAUJS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJSACT[2:0]	Source clock for currently active C_AWO_TAUJ

(3) CKSC_ATAUJD_CTL — C_AWO_TAUJ Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2200_H
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.25 CKSC_ATAUJD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJDCSID[2:0]	Clock Divider Setting for C_AWO_TAUJ 000_B: Setting prohibited 001_B: CKSC_ATAUJS_CTL selection /1 (default) 010_B: CKSC_ATAUJS_CTL selection /2 011_B: CKSC_ATAUJS_CTL selection /4 100_B: CKSC_ATAUJS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ATAUJD_ACT — C_AWO_TAUJ Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2208_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.26 CKSC_ATAUJD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJDACT[2:0]	Clock divider for currently active C_AWO_TAUJ

(5) CKSC_ATAUJD_STPM — C_AWO_TAUJ Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2218$_H$
Value after reset:	0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJ DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.27 CKSC_ATAUJD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ATAUJDSTPMSK	0: Clock domain C_AWO_TAUJ is stopped in stand-by mode. 1: Clock domain C_AWO_TAUJ is not stopped in stand-by mode.

12C.4.3.3 RTCA Clock Domain C_AWO_RTCA

(1) CKSC_ARTCAS_CTL — C_AWO_RTCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2300$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.28 CKSC_ARTCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ARTCASCSID[1:0]	Source Clock Setting for C_AWO_RTCA 00$_B$: Disabled (default) 01$_B$: Setting Prohibited 10$_B$: MainOSC*1 11$_B$: LS IntOSC

Note 1. To avoid supplying a clock signal equal to or higher than 4 MHz to the C_AWO_RTCA clock domain, check that CKSC_ARTCAD_ACT = 0000 0000$_H$ (disabled) before setting the CKSC_ARTCAS_CTL to 10$_B$ (MainOSC).

(2) CKSC_ARTCAS_ACT — C_AWO_RTCA Source Clock Active Register

This register is initialized by a power-up reset PURES.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 2308_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.29 CKSC_ARTCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ARTCASACT[1:0]	Source clock for currently active C_AWO_RTCA

(3) CKSC_ARTCAD_CTL — C_AWO_RTCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2400_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.30 CKSC_ARTCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ARTCADCSID[2:0]	Clock Divider Setting for C_AWO_RTCA 000_B: Disabled (default) 001_B: CKSC_ARTCAS_CTL selection /1 010_B: CKSC_ARTCAS_CTL selection /2 011_B: CKSC_ARTCAS_CTL selection /4 100_B: CKSC_ARTCAS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ARTCAD_ACT — C_AWO_RTCA Clock Divider Active Register

This register is initialized by a power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2408$_\text{H}$

Value after reset: 0000 0000$_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.31 CKSC_ARTCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ARTCADACT[2:0]	Clock divider for currently active C_AWO_RTCA

(5) CKSC_ARTCAD_STPM — C_AWO_RTCA Stop Mask Register

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2418_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCA DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.32 CKSC_ARTCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARTCADSTPMSK	0: Clock domain C_AWO_RTCA is stopped in stand-by mode. 1: Clock domain C_AWO_RTCA is not stopped in stand-by mode.

12C.4.3.4 ADCA0 Clock Domain C_AWO_ADCA

(1) CKSC_AADCAS_CTL — C_AWO_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2500$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.33 CKSC_AADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCASCSID[1:0]	Source Clock Setting for C_AWO_ADCA*1 00$_B$: Disabled 01$_B$: HS IntOSC (default) 10$_B$: MainOSC 11$_B$: PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_AADCAS_ACT — C_AWO_ADCA Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 2508$_{H}$
Value after reset:	0000 0001$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.34 CKSC_AADCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCASACT[1:0]	Source clock for currently active C_AWO_ADCA

(3) CKSC_AADCAD_CTL — C_AWO_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2600$_{H}$

Value after reset: 0000 0001$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.35 CKSC_AADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCADCSID[1:0]	Clock Divider Setting for C_AWO_ADCA 00_{B}: Setting prohibited 01_{B}: CKSC_AADCAS_CTL selection /1 (default) 10_{B}: CKSC_AADCAS_CTL selection /2*1 11_{B}: Setting prohibited

Note 1. Make sure that the frequency of CKSC_AADCA is no less than 8 MHz after division by 2.

(4) CKSC_AADCAD_ACT — C_AWO_ADCA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2608_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.36 CKSC_AADCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCADACT[1:0]	Clock divider for currently active for C_AWO_ADCA

(5) CKSC_AADCAD_STPM — C_AWO_ADCA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 2618$_H$
Value after reset:	0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCA DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.37 CKSC_AADCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AADCADSTPMSK	0: Clock domain C_AWO_ADCA is stopped in stand-by mode. 1: Clock domain C_AWO_ADCA is not stopped in stand-by mode.

12C.4.3.5 FOUT Clock Domain C_AWO_FOUT

(1) CKSC_AFOUTS_CTL — C_AWO_FOUT Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2700$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.38 CKSC_AFOUTS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	AFOUTSCSID[2:0]	Source Clock Setting for C_AWO_FOUT*1 000$_B$: Disabled (default) 001$_B$: MainOSC 010$_B$: HS IntOSC 011$_B$: LS IntOSC 100$_B$: Setting prohibited 101$_B$: PPLLCLK4 110$_B$: PPLLCLK4 111$_B$: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK4.

(2) CKSC_AFOUTS_ACT — C_AWO_FOUT Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2708$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.39 CKSC_AFOUTS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	AFOUTSACT[2:0]	Source clock for currently active C_AWO_FOUT

(3) CKSC_AFOUTS_STPM — C_AWO_FOUT Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2718_H

Value after reset: 0000 0002_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.40 CKSC_AFOUTS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AFOUTSSTPMSK	0: Clock domain C_AWO_FOUT is stopped in stand-by mode. 1: Clock domain C_AWO_FOUT is not stopped in stand-by mode.

12C.4.3.6 CPU Clock Domain C_ISO_CPUCLK

(1) CKSC_CPUCLKS_CTL — C_ISO_CPUCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A000$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.41 CKSC_CPUCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CPUCLKSCSID[1:0]	Source Clock Setting for C_ISO_CPUCLK 00$_B$: Setting prohibited 01$_B$: EMCLK (default) 10$_B$: MainOSC 11$_B$: CPLL1OUT

CAUTION

The clock source selected for the C_ISO_CPUCLK clock domain should not be stopped by software.

(2) CKSC_CPUCLKS_ACT — C_ISO_CPUCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A008$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.42 CKSC_CPUCLKS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	CPUCLKSACT[1:0]	Source clock for currently active C_ISO_CPUCLK

(3) CKSC_CPUCLKD_CTL — C_ISO_CPUCLK Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A100$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDPLL [1:0]		CPUCLKDCSID [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 12C.43 CKSC_CPUCLKD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4, 3	CPUCLKDPLL[1:0]	Clock Divider CPLLDIV Setting Specifies the CPLLDIV divisor, which determines maximum clock frequency of C_ISO_CPUCLK. 00_B: CPLL1OUT = VCO1OUT × 1/6 (80 MHz) 01_B: CPLL1OUT = VCO1OUT × 1/5 (96 MHz) 10_B: CPLL1OUT = VCO1OUT × 1/4 (120 MHz) 11_B: Setting prohibited
2 to 0	CPUCLKDCSID[2:0]	Clock Divider Setting for C_ISO_CPUCLK 000_B: Setting prohibited 001_B: CKSC_CPUCLKS_CTL selection /1 (Default) 010_B: CKSC_CPUCLKS_CTL selection /2 011_B: CKSC_CPUCLKS_CTL selection /4 100_B: CKSC_CPUCLKS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_CPUCLKD_ACT — C_ISO_CPUCLK Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A108$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDPLLACT[1:0]		CPUCLKDACT [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.44 CKSC_CPUCLKD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4, 3	CPUCLKDPLLACT [1:0]	Clock divider for currently active CPLLDIV.
2 to 0	CPUCLKDACT [2:0]	Clock divider for currently active C_ISO_CPUCLK

12C.4.3.7 Peripheral Clock Domains C_ISO_PERI1 and C_ISO_PERI2

(1) CKSC_IPERI1S_CTL — C_ISO_PERI1 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A200$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.45 CKSC_IPERI1S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI1SCSID[1:0]	Source Clock Setting for C_ISO_PERI1 00$_B$: Disabled 01$_B$: PPLLCLK (default) 10$_B$: PPLLCLK 11$_B$: Setting prohibited

(2) CKSC_IPERI1S_ACT — C_ISO_PERI1 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 $A208_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.46 CKSC_IPERI1S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI1SACT[1:0]	Source clock for currently active C_ISO_PERI1

(3) CKSC_IPERI2S_CTL — C_ISO_PERI2 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A300$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.47 CKSC_IPERI2S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI2SCSID[1:0]	Source Clock Setting for C_ISO_PERI2 00$_B$: Disabled 01$_B$: PPLLCLK2 (default) 10$_B$: PPLLCLK2 11$_B$: Setting prohibited

(4) CKSC_IPERI2S_ACT — C_ISO_PERI2 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A308$_{\text{H}}$

Value after reset: 0000 0001$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.48 CKSC_IPERI2S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI2SACT[1:0]	Source clock for currently active C_ISO_PERI2

12C.4.3.8 RLIN Clock Domains C_ISO_LIN

(1) CKSC_ILINS_CTL — C_ISO_LIN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A400$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.49 CKSC_ILINS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ILINSCSID[2:0]	Source Clock Setting for C_ISO_LIN*[1] 000$_B$: Disabled 001$_B$: PPLLCLK2 (default) 010$_B$: MainOSC 011$_B$: PPLLCLK2 100$_B$: HS IntOSC Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ILINS_ACT — C_ISO_LIN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 $A408_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.50 CKSC_ILINS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ILINSACT[2:0]	Source clock for currently active C_ISO_LIN

(3) CKSC_ILIND_CTL — C_ISO_LIN Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 A800$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.51 CKSC_ILIND_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINDCSID[1:0]	Clock Divider Setting for C_ISO_LIN 00$_B$: Setting prohibited 01$_B$: CKSC_ILINS_CTL selection /1 (default) 10$_B$: CKSC_ILINS_CTL selection /4 11$_B$: CKSC_ILINS_CTL selection /8

NOTE

The setting of this register is only applicable to RLIN30. The settings 10$_B$ (CKSC_ILINS_CTL selection /4) and 11$_B$ (CKSC_ILINS_CTL selection /8) are only available in UART mode.

(4) CKSC_ILIND_ACT — C_ISO_LIN Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A808$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.52 CKSC_ILIND_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINDACT[1:0]	Clock divider for currently active C_ISO_LIN

(5) CKSC_ILIND_STPM — C_ISO_LIN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A818$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILIND STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.53 CKSC_ILIND_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ILINDSTPMSK	0: Clock domain C_ISO_LIN is stopped in stand-by mode. 1: Clock domain C_ISO_LIN is not stopped in stand-by mode.

12C.4.3.9 RS-CANFD Clock Domains C_ISO_CAN and C_ISO_CANOSC

(1) CKSC_ICANS_CTL — C_ISO_CAN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A900$_{\text{H}}$

Value after reset: 0000 0003$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.54 CKSC_ICANS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANSCSID[1:0]	Source Clock Setting for C_ISO_CAN*[1] 00$_{\text{B}}$: Disabled 01$_{\text{B}}$: MainOSC 10$_{\text{B}}$: PPLLCLK 11$_{\text{B}}$: PPLLCLK (default)

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK.

(2) CKSC_ICANS_ACT — C_ISO_CAN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A908$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.55 CKSC_ICANS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANSACT[1:0]	Source clock for currently active C_ISO_CAN

(3) CKSC_ICANS_STPM — C_ISO_CAN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A918$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANS STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.56 CKSC_ICANS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANSSTPMSK	0: Clock domain C_ISO_CAN is stopped in stand-by mode. 1: Clock domain C_ISO_CAN is not stopped in stand-by mode.

(4) CKSC_ICANOSCD_CTL — C_ISO_CANOSC Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCD CSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.57 CKSC_ICANOSCD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANOSCDCSID[1:0]	Clock Divider Setting for C_ISO_CANOSC*[1] 00$_B$: Disabled (default) 01$_B$: MainOSC/1 10$_B$: MainOSC/2 11$_B$: Setting prohibited

Note 1. Select MainOSC/2 when the source clock setting of C_ISO_CAN is MainOSC.

(5) CKSC_ICANOSCD_ACT — C_ISO_CANOSC Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 AA08$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.58 CKSC_ICANOSCD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANOSCDACT[1:0]	Clock divider for currently active C_ISO_CANOSC

(6) CKSC_ICANOSCD_STPM — C_ISO_CANOSC Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA18$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the “1” value after reset of bit 1.

Table 12C.59 CKSC_ICANOSCD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANOSCDSTPMSK	0: Clock domain C_ISO_CANOSC is stopped in stand-by mode. 1: Clock domain C_ISO_CANOSC is not stopped in stand-by mode.

12C.4.3.10 CSI Clock Domain C_ISO_CSI

(1) CKSC_ICSIS_CTL — C_ISO_CSI Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 AB00$_H$
Value after reset:	0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12C.60 CKSC_ICSIS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ICSISCSID[2:0]	Source Clock Setting for C_ISO_CSI 000$_B$: Disabled 001$_B$: PPLLCLK (default) 010$_B$: PPLLCLK 011$_B$: MainOSC 100$_B$: HS IntOSC Other than above: Setting prohibited

(2) CKSC_ICSIS_ACT — C_ISO_CSI Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AB08$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.61 CKSC_ICSIS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ICSISACT[2:0]	Source clock for currently active C_ISO_CSI

12C.4.3.11 RIIC Clock Domain C_ISO_IIC

(1) CKSC_IIICS_CTL — C_ISO_IIC Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AC00$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12C.62 CKSC_IIICS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IIICSCSID[1:0]	Source Clock Setting for C_ISO_IIC 00$_B$: Disabled 01$_B$: PPLLCLK2 (default) 10$_B$: PPLLCLK2 11$_B$: Setting prohibited

(2) CKSC_IIICS_ACT — C_ISO_IIC Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AC08$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.63 CKSC_IIICS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IIICSACT[1:0]	Source clock for currently active C_ISO_IIC

12C.5 Clock Domain Setting Method

12C.5.1 Clock Domain Setting

12C.5.1.1 Overview of Clock Selector Register

The clock selector for a clock domain C_AWO_<*name*>/C_ISO_<*name*> can be controlled by the following registers:

- Source clock selection registers
  These registers select the clock to be used as the domain clock from the available source clocks.
  - AWO source clock selection: CKSC_A<*name*>S_CTL
  - ISO source clock selection: CKSC_I<*name*>S_CTL
- Clock divider selection registers
  These registers specify the clock division ratio for the selected source clock.
  - AWO clock divider: CKSC_A<*name*>D_CTL
  - ISO clock divider: CKSC_I<*name*>D_CTL
- Source clock active registers and clock divider active registers
  These registers return the currently active source clock selection and division ratio, respectively.
  - AWO source clock active register/clock divider active register: CKSC_A<*name*>S_ACT/CKSC_A<*name*>D_ACT
  - ISO source clock active register/clock divider active register: CKSC_I<*name*>S_ACT/CKSC_I<*name*>D_ACT

NOTE

- Not all clock selectors provide all of the control functions described above.
- The symbol “I”, which indicates the power domain, is not added to the names of registers within clock domain C_ISO_CPUCLK.

12C.5.1.2 Setting Procedure for Clock Domain

Procedure of setting up clock domain is described as below:

1. Set up a source clock
   - Select a source clock. (CKSC_A<*name*>S_CTL, CKSC_I<*name*>S_CTL)
   - Confirm completion of selection. (CKSC_A<*name*>S_ACT, CKSC_I<*name*>S_ACT)*[1]
2. Setting a clock divider
   - Select a clock divider. (CKSC_A<*name*>D_CTL, CKSC_I<*name*>D_CTL)
   - Confirm completion of selection. (CKSC_A<*name*>D_ACT, CKSC_I<*name*>D_ACT)*[2]

Note 1. Continue processing after CKSC_A<*name*>S_ACT and CKSC_I<*name*>S_ACT are updated with the new values written to CKSC_A<*name*>S_CTL and CKSC_I<*name*>S_CTL.

Note 2. Continue processing after CKSC_A<*name*>D_ACT and CKSC_I<*name*>D_ACT are updated with the new values written to CKSC_A<*name*>D_CTL and CKSC_I<*name*>D_CTL.

CAUTION

The source clock to be selected must be operating before performing these settings.
The behavior and performance are not guaranteed if setup is performed while the source clock is stopped.
Access to a peripheral module is prohibited while the clock is not supplied to the module.

12C.5.2 Stopping the Clock in Stand-by Mode

In stand-by mode (STOP mode, DeepSTOP mode, and Cyclic STOP mode), clock domain C_AWO_<*name*>/C_ISO_<*name*> can be configured to stop or continue its clock CKSCLK_A<*name*>/CKSCLK_I<*name*> in response to clock stop requests from the stand-by controller.

The clock stop mask registers are used to determine the operation status of the clock in stand-by mode:

- CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK = 0:
  The STOP request signal is not masked, so the domain clock CKSCLK_A<*name*>/CKSCLK_I<*name*> is stopped during stand-by mode.
  If the domain clock was in operation before transition to stand-by mode, it is automatically re-started after wake-up from stand-by mode.
  If there is another clock domain which the same source clock is selected and its stop mask setting is set to 1 (CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK = 1), the source clock will continue operation in stand-by mode.
  The CPU clock domain C_ISO_CPUCLK is always stopped in stand-by mode.
- CKSC_A<*name*>_STPM.A<*name*>STPMSK/CKSC_I<*name*>_STPM.I<*name*>STPMSK = 1:
  The STOP request signal is masked, so CKSCLK_A<*name*>/CKSCLK_I<*name*> continues to operate during stand-by.
  The source clock selected for the target clock domain will also continue to operate in stand-by mode.
  Supply of a clock signal to the clock domains in the Isolated area (ISO area) will be stopped in DeepSTOP mode.

12C.5.3 Clock Domain Settings

The following table shows a selectable source clock, a frequency division ratio, and a register to be used for each clock domain.

Table 12C.64 List of Selectable Clocks

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_AWO_WDTA	CKSCLK_AWDTA	—	**LS IntOSC**	CKSC_AWDTAD_CTL	1/1	240 kHz	WDTA0
					1/128		
C_AWO_TAUJ	CKSCLK_ATAUJ	CKSC_ATAUJS_CTL	MainOSC	CKSC_ATAUJD_CTL	**1/1**	40 MHz	TAUJ0 TAUJ2
			HS IntOSC		1/2		
			LS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable		—		
C_AWO_RTCA	CKSCLK_ARTCA	CKSC_ARTCAS_CTL	MainOSC	CKSC_ARTCAD_CTL	1/1	4 MHz	RTCA0
			LS IntOSC		1/2		
			Disable		1/4		
			—		1/8		
					Disable		
C_AWO_ADCA	CKSCLK_AADCA	CKSC_AADCAS_CTL	MainOSC	CKSC_AADCAD_CTL	**1/1**	40 MHz	ADCA0
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_AWO_FOUT	CKSCLK_AFOUT	CKSC_AFOUTS_CTL	MainOSC	—	**1/1**	24 MHz	FOUT
			HS IntOSC				
			LS IntOSC				
			PPLLCLK4				
			Disable				
C_ISO_CPUCLK	CPUCLK	CKSC_CPUCLKS_CTL*2 CKSC_CPUCLKD_CTL*2	MainOSC	CKSC_CPUCLKD_CTL	**1/1**	120 MHz*3	CPU subsystem
			CPLL1OUT (VCO1OUT × 1/4)		1/2		
			CPLL1OUT (VCO1OUT × 1/5)		1/4		
			CPLL1OUT (VCO1OUT × 1/6)		1/8		
			EMCLK		—		
C_ISO_PERI1	CKSCLK_IPERI1	CKSC_IPERI1S_CTL	**PPLLCLK**	—	**1/1**	80 MHz	TAUD0 TAUJ1 TAUJ3 ENCA0 TAPA0 PIC0
			Disable				
C_ISO_PERI2	CKSCLK_IPERI2	CKSC_IPERI2S_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	TAUBn PWBAn PWGAn PWSAn RCFDCn (clkc) RSENTn
			Disable				
C_ISO_LIN	CKSCLK_ILIN	CKSC_ILINS_CTL	MainOSC	CKSC_ILIND_CTL*1	**1/1**	40 MHz	RLIN24n RLIN3n
			HS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable				

Table 12C.64 List of Selectable Clocks

Clock Domain	Clock Name	Selectable Register		Frequency Divided Register		Maximum Frequency	Applicable Unit
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	**1/1**	80 MHz	RCFDCn (PCLK)
			PPLLCLK				
			Disable				
C_ISO_CANOSC	CKSCLK_ICANOSC	—	**MainOSC**	CKSC_ICANOSCD_CTL	1/1	24 MHz	RCFDCn (clk_xincan)
					1/2		
					Disable		
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSIS_CTL	**PPLLCLK**	—	**1/1**	80 MHz	CSIGn CSIHn
			MainOSC				
			HS IntOSC				
			Disable				
C_ISO_IIC	CKSCLK_IIIC	CKSC_IIICS_CTL	**PPLLCLK2**	—	**1/1**	40 MHz	RIICn
			Disable				

Note: The items written in bold are the initial setting clocks for each register.

Note 1. The setting of this register only applies to RLIN30. The settings 1/4 and 1/8 are only available in UART mode.

Note 2. CKSC_CPUCLKS_CTL selects selection of Main OSC, CPLL1OUT and EMCLK. CKSC_CPUCLKD_CTL.CPUCLKDPLL[1:0] selects CPLL1OUT clock frequency.

Note 3. For the supported settings, refer to the **Table 12C.14, PLL1 Output Table**.

CAUTION

To stop the clock source selected for the clock domain before transitioning to STOP/DeepSTOP mode, select “Disable” for that clock domain in advance. Do not stop the source clock of a clock domain for which “Disable” cannot be selected while functions are operating on that clock domain. To stop the clock source selected for the domain by transitioning to STOP/DeepSTOP mode, “Disable” does not need to be selected.
Instead of the setting “Disable”, select “Stop” for the clock domain in stand-by mode by using the stop mask register.

12C.6 Frequency Output Function (FOUT)

The frequency output function (FOUT) allows the clock to be output the clock as the external signal. Furthermore, the frequency can be divided by the clock divider before it is output.

12C.6.1 Functional Overview

Figure 12C.8, Frequency Output Function shows the configuration of the frequency output function.

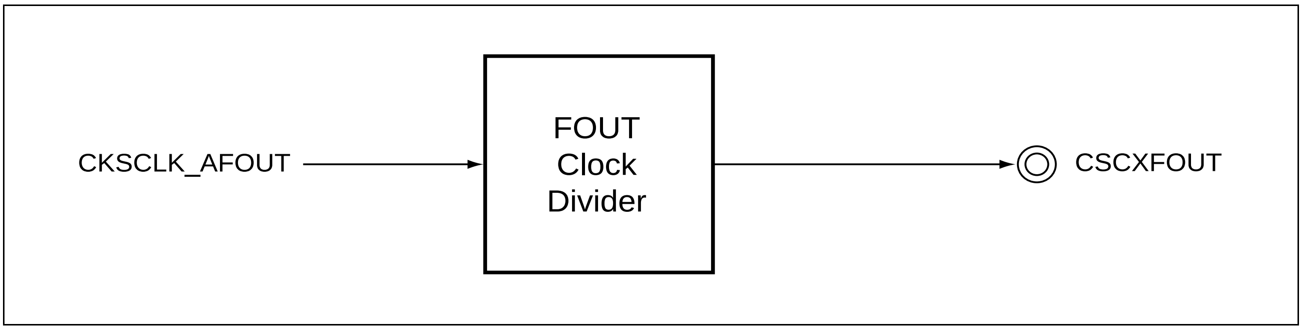


Figure 12C.8 Frequency Output Function

The clock output function outputs the CKSCLK_AFOUT clock divided by 1 to 63 through the clock divider from CSCXFOUT. Division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register. Clock output frequency f_{CSCXFOUT} is expressed by the following equation.

$$f_{\text{CSCXFOUT}} = (\text{CKSCLK\_AFOUT clock frequency}) / N$$

Clock output starts when, after CKSCLK_AFOUT is set and the clock output for the pin function is selected, division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register.

When a new division ratio is written to the FOUTDIV.FOUTDIV[5:0] bits, it becomes effective in synchronization with the CSCXFOUT output clock. Accordingly, the division ratio can be changed even while the CSCXFOUT clock is operating. The clock output is stopped by writing 000_H to the FOUTDIV[5:0] bits.

12C.6.2 Clock Supply

The clock supply to the CSCXFOUT is shown in the following table.

Table 12C.65 Clock Supply

Module	Unit Clock Name	Supply Clock Name	Description
CSCXFOUT	PCLK	CPUCLK_UL	Bus clock (Register access)
	CKSCLK_AFOUT	CKSCLK_AFOUT	Clock source of FOUT clock divider

12C.6.3 Registers

12C.6.3.1 List of Registers

The FOUT registers are listed in the following table.

Table 12C.66 List of Registers

Module Name	Register Name	Symbol	Address
CLKCTL	Clock division ratio register	FOUTDIV	FFF8 2800_H
	Clock divider status register	FOUTSTAT	FFF8 2804_H

12C.6.3.2 FOUTDIV — Clock Division Ratio Register

This register defines the clock divisor.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2800_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FOUTDIV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12C.67 FOUTDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FOUTDIV[5:0]	Clock Divider N 00_H: Clock output is stopped 01_H: N = 1 02_H: N = 2 : $3E_H$: N = 62 $3F_H$: N = 63

12C.6.3.3 FOUTSTAT — Clock Divider Status Register

This register indicates the clock output status.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2804$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTCLKACT	FOUTSYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12C.68 FOUTSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUTCLKACT	Clock Divider Active 0: Frequency output is stopped. 1: Frequency output is ongoing.
0	FOUTSYNC	Clock Divider Synchronized 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

Section 13 Clock Monitor (CLMA)

This section contains a generic description of the clock monitor (CLMA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of CLMA.

13.1 Features of RH850/F1KH, RH850/F1KM CLMA

13.1.1 Number of Units

This microcontroller has the following number of CLMA units.

Each CLMA unit has single channel interface. “channel” is used with the same meaning as “unit” in this section.

Table 13.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	4	4	4
Name	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)

Table 13.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	4	4	4	4	4
Name	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)	CLMAn (n = 0 to 3)

Table 13.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	3	3	3	3
Name	CLMAn (n = 0, 1, 3)	CLMAn (n = 0, 1, 3)	CLMAn (n = 0, 1, 3)	CLMAn (n = 0, 1, 3)

Table 13.4 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual CLMA Channels are identified by the index “n”: for example, CLMAnCTL0 (n = 0 to 3) is the CLMAn control register 0.

Table 13.5 Index (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual CLMA Channels are identified by the index “n”: for example, CLMAnCTL0 (n = 0 to 3) is the CLMAn control register 0.

Table 13.6 Index (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual CLMA Channels are identified by the index “n”: for example, CLMAnCTL0 (n = 0, 1, 3) is the CLMAn control register 0.

13.1.2 Register Base Addresses

The CLMA base addresses are listed in the following table.

The CLMA register addresses are given as offsets from the base addresses.

Table 13.7 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<CLMA0_base>	FFF8 C000$_H$
<CLMA1_base>	FFF8 D000$_H$
<CLMA2_base>	FFF8 E000$_H$
<CLMA3_base>	FFF8 F000$_H$

Table 13.8 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<CLMA0_base>	FFF8 C000$_H$
<CLMA1_base>	FFF8 D000$_H$
<CLMA2_base>	FFF8 E000$_H$
<CLMA3_base>	FFF8 F000$_H$

Table 13.9 Register Base Addresses (RH850/F1KM-S1)

Base Address Name	Base Address
<CLMA0_base>	FFF8 C000$_H$
<CLMA1_base>	FFF8 D000$_H$
<CLMA3_base>	FFF8 E000$_H$

13.1.3 Clock Supply

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.

Table 13.10 Clock Supply (RH850/F1KH-D8)

Channel Name	Unit Clock Name	Supply Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA2	CLMATMON (monitored clock)	PLL0OUT*1
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK_UL
CLMA3	CLMATMON (monitored clock)	PPLLOUT
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK_UL

Note 1. For details, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4, Section 12AB.3.5.1, PLL0 Parameters, Figure 12AB.8, PLL0 Circuit**.

Table 13.11 Clock Supply (RH850/F1KM-S4)

Channel Name	Unit Clock Name	Supply Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA2	CLMATMON (monitored clock)	PLL0OUT*1
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK_UL
CLMA3	CLMATMON (monitored clock)	PPLLOUT
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK_UL

Note 1. For details, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4, Section 12AB.3.5.1, PLL0 Parameters, Figure 12AB.8, PLL0 Circuit**.

Table 13.12 Clock Supply (RH850/F1KM-S1)

Channel Name	Unit Clock Name	Supply Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK_UL
CLMA3	CLMATMON (monitored clock)	PPLLOUT
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK_UL

13.1.4 Reset Sources

The reset sources of the CLMA are listed in the following table. The CLMA are initialized by these reset sources.

Table 13.13 Reset Sources (RH850/F1KH-D8)

Channel Name	Reset Source
CLMA0	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA1	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA2	All reset sources (ISORES)
CLMA3	All reset sources (ISORES)
Common Registers (CLMATEST, CLMATESTS)	Power-up reset PURES (power-on-clear or debugger reset)

Table 13.14 Reset Sources (RH850/F1KM-S4)

Channel Name	Reset Source
CLMA0	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA1	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA2	All reset sources (ISORES)
CLMA3	All reset sources (ISORES)
Common Registers (CLMATEST, CLMATESTS)	Power-up reset PURES (power-on-clear or debugger reset)

Table 13.15 Reset Sources (RH850/F1KM-S1)

Channel Name	Reset Source
CLMA0	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA1	Reset sources other than transition to DeepSTOP mode (AWORES)
CLMA3	All reset sources (ISORES)
Common Registers (CLMATEST, CLMATESTS)	Power-up reset PURES (power-on-clear or debugger reset)

13.1.5 Internal Input/Output Signals

The internal input/output signals of CLMA are listed in the following table.

Table 13.16 Internal Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Connection
$\overline{\text{CLMATRES}}$	CLMA0 error reset output	Reset controller ($\overline{\text{CLMA0RES}}$)
$\overline{\text{CLMATRES}}$	CLMA1 error reset output	Reset controller ($\overline{\text{CLMA1RES}}$)
$\overline{\text{CLMATRES}}$	CLMA2 error reset output	Reset controller ($\overline{\text{CLMA2RES}}$)
$\overline{\text{CLMATRES}}$	CLMA3 error reset output	Reset controller ($\overline{\text{CLMA3RES}}$)

Table 13.17 Internal Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Connection
$\overline{\text{CLMATRES}}$	CLMA0 error reset output	Reset controller ($\overline{\text{CLMA0RES}}$)
$\overline{\text{CLMATRES}}$	CLMA1 error reset output	Reset controller ($\overline{\text{CLMA1RES}}$)
$\overline{\text{CLMATRES}}$	CLMA2 error reset output	Reset controller ($\overline{\text{CLMA2RES}}$)
$\overline{\text{CLMATRES}}$	CLMA3 error reset output	Reset controller ($\overline{\text{CLMA3RES}}$)

Table 13.18 Internal Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Connection
$\overline{\text{CLMATRES}}$	CLMA0 error reset output	Reset controller ($\overline{\text{CLMA0RES}}$)
$\overline{\text{CLMATRES}}$	CLMA1 error reset output	Reset controller ($\overline{\text{CLMA1RES}}$)
$\overline{\text{CLMATRES}}$	CLMA3 error reset output	Reset controller ($\overline{\text{CLMA3RES}}$)

13.2 Overview

13.2.1 Functional Overview

Clock monitor CLMA detects frequency abnormalities in the monitored clock. It uses sampling clock CLMATSMP to monitor whether the frequency of input clock CLMATMON is within a specific range.

Upon detection of an abnormal clock, it outputs a reset request signal.

The main components of the clock monitor are shown in **Figure 13.1, Block Diagram of the Clock Monitor**.

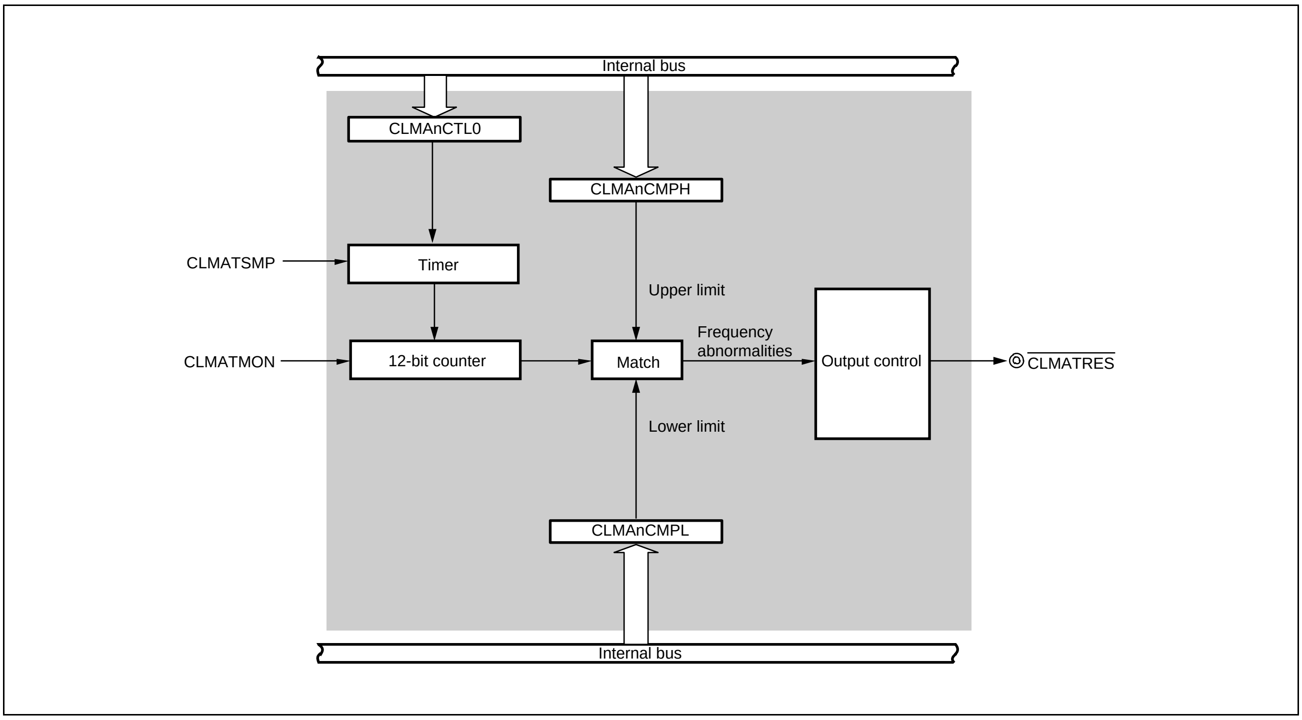


Figure 13.1 Block Diagram of the Clock Monitor

NOTE

Once enabled, only a reset can disable the CLMAn.

13.3 Enabling CLMA

Clock monitoring is started by the clock monitor when CLMAnCTL0.CLMAnCLME = 1.

When the monitored clock is stopped by a register operation or transition to stand-by mode, the corresponding clock monitor is automatically disabled. After the monitored clock starts oscillation again and becomes stable, the clock monitor also starts operation.

Since CLMA2 and CLMA3 is initialized on return from DeepSTOP, the CLMA2 and CLMA3 register must be set again before further operation is started.

13.4 Functions

13.4.1 Detection of Abnormal Clock Frequencies

Detection Method

1. CLMAn counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMP and then compares the counter value with the specified thresholds:
    - CLMAnCMPL.CLMAnCMPL[11:0] defines the lower threshold.
    - CLMAnCMPH.CLMAnCMPH[11:0] defines the upper threshold.
2. When CLMATMON frequency*[1] is lower than the limit, the counter falls below CLMAnCMPL.CLMAnCMPL[11:0].
3. When the frequency of CLMATMON is higher than the limit, the counter exceeds CLMAnCMPH.CLMAnCMPH[11:0].

Note 1. There is a case that the abnormal state is not detected when the monitor clock completely stops.

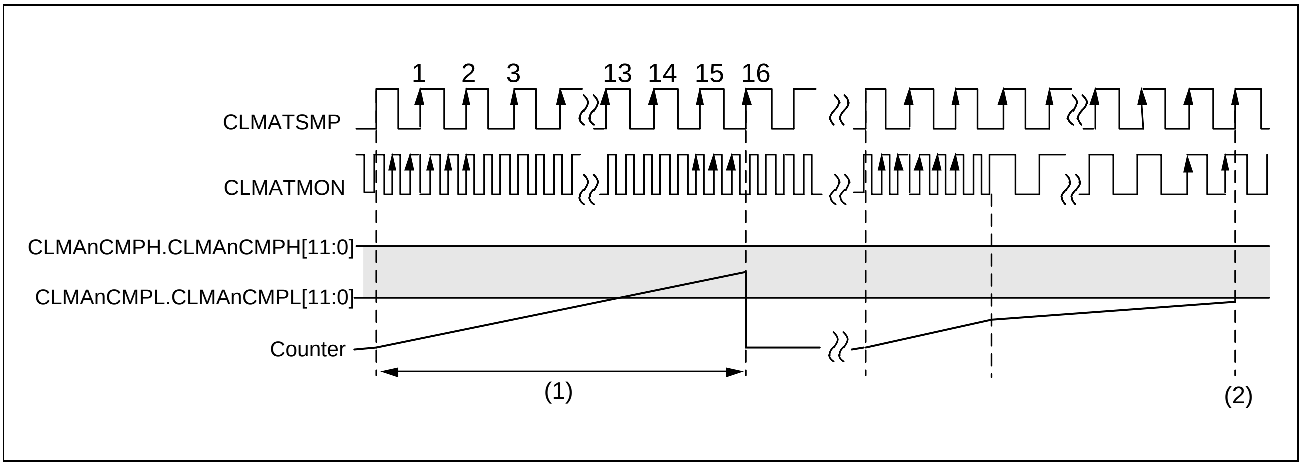


Figure 13.2 Example: $f_{CLMATMON}$ is Lower than the Specified Limit

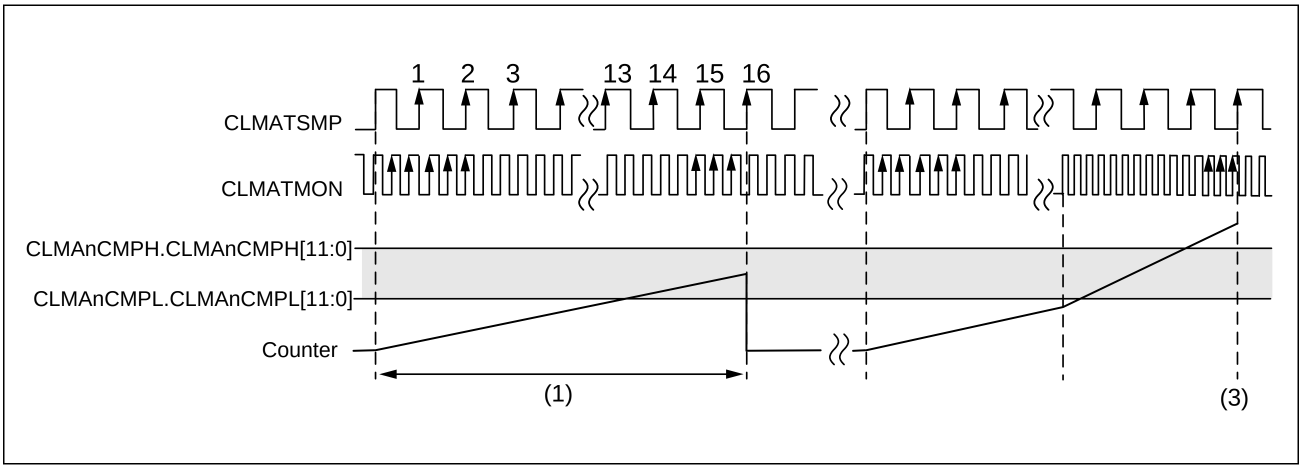


Figure 13.3 Example: $f_{CLMATMON}$ is Higher than the Specified Limit

NOTE

Even if $f_{CLMATMON}$ exceeds or falls below the specified limits during a sampling interval, the counter might be within the valid range.

Abnormal $f_{CLMATMON}$ is detected after one sampling interval.

(1) Calculation Method of the Thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]

The compare registers CLMAnCMPL and CLMAnCMPH are configured with the minimum and maximum number of clock cycles of CLMATMON that are assumed to be valid within 16 cycles of the sampling clock CLMATSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMATSMP}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the threshold values can be calculated by the following formulas:

$$\text{Lower threshold} = N_{min} = \frac{f_{CLMATMON(dmin)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

NOTE

The jitter of the PLL is covered by “+1” and “–1” in the formulas.

Example:

When $f_{CLMATSMP}$ = 240 kHz (±8%) and $f_{CLMATMON}$ = 16 MHz (±5%), the recommended threshold values are as follows:

N_{min} = 15,200 / 259.2 × 16 – 1
= 937.27
CLMAnCMPL = 937 = $03A9_H$

N_{max} = 16,800 / 220.8 × 16 + 1
= 1218.39
CLMAnCMPH = 1219 = $04C3_H$

Minimum Thresholds

The following restrictions must be taken into account:

- CLMAnCMPL ≥ 0001_H
- CLMAnCMPH ≥ CLMAnCMPL + 0003_H

(2) Definition of the Initial Value Input to the Threshold Registers

The reset values of the threshold registers are set as follows:

- CLMAnCMPL[11:0] = 001_H
- CLMAnCMPH[11:0] = $3FF_H$

13.4.2 Notification of Abnormal Clock Frequency

If $f_{CLMATMON}$ exceeds the upper threshold or falls below the lower threshold, this is indicated as follows:

1. The reset request signal $\overline{\text{CLMATRES}}$ is set to low level.
2. The system reset (AWORES or ISORES) is generated and CLMAn is reset.

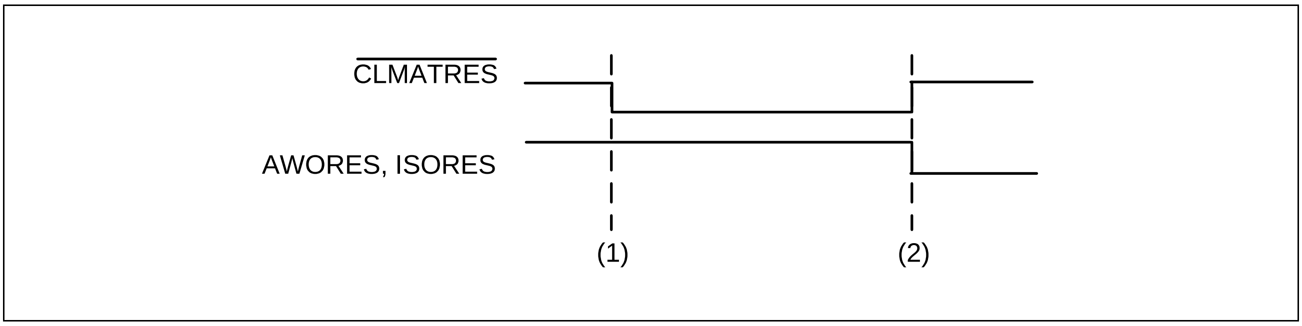


Figure 13.4 Error Request Signal Output if $f_{CLMATMON}$ Exceeds Upper Threshold

CAUTION

For usage notes for CLMAn abnormality detection, see **Section 13.6, Usage Notes for CLMAn**.

13.5 Registers

13.5.1 List of Registers

The following table lists the CLMA registers.

<CLMAn_base> is defined in **Section 13.1.2, Register Base Addresses**.

Table 13.19 List of Registers

Module Name	Register Name	Symbol	Address
CLMAn	CLMAn control register 0	CLMAnCTL0	<CLMAn_base> + 00_H
	CLMAn compare register L	CLMAnCMPL	<CLMAn_base> + 08_H
	CLMAn compare register H	CLMAnCMPH	<CLMAn_base> + $0C_H$
	CLMAn emulation register 0	CLMAnEMU0	<CLMAn_base> + 18_H
CLMA	CLMA test register	CLMATEST	FFF8 $C100_H$
	CLMA test status register	CLMATESTS	FFF8 $C104_H$

13.5.2 CLMAnCTL0 — CLMAn Control Register 0

This register enables the clock monitor CLMAn.

The correct write sequence using the CLMAnPCMD register is required in order to update this register.
For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 8-bit units.

Address: <CLMAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnCLME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 13.20 CLMAnCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CLMAnCLME	Enables or disables the clock monitor. 0: CLMAn is disabled. 1: CLMAn is enabled.

CAUTION

The CLMAnPRERR bit in CLMAnPS register is set to 1 when 0 is written to the CLMAnCTL0.CLMAnCLME bit. The only condition for clearing the bit is a reset (AWORES, ISORES). In addition, the bit is cleared when the CLMATEST.RESCLM bit is set to 1 during self-test of CLMAn.

13.5.3 CLMAnCMPH — CLMAn Compare Register H

This register specifies the upper limit of frequency.

It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

For details, see **Section 13.4.1(1), Calculation Method of the Thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + $0C_H$

Value after reset: $03FF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPH[11:0]											
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.21 CLMAnCMPH Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPH[11:0]	Specifies the upper threshold. ● The recommended value is $f_{CLMATMON\ (max)} / f_{CLMATSMP\ (min)} \times 16 + 1$. ● The minimum value is CLMAnCMPL + 0003_H.

13.5.4 CLMAnCMPL — CLMAn Compare Register L

This register specifies the lower limit of frequency.

It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

For details, see **Section 13.4.1(1), Calculation Method of the Thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + 08_H

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPL[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.22 CLMAnCMPL Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPL[11:0]	Specifies the lower threshold. ● The recommended value is $f_{CLMATMON\ (min)} / f_{CLMATSMP\ (max)} \times 16 - 1$. ● The minimum value is 0001_H.

13.5.5 CLMATEST — CLMA Test Register

This register is used to control self-test of CLMA0, CLMA1, CLMA2, and CLMA3.

The correct write sequence using the PROTCMDCLMA register is required in order to update this register.
For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 C100$_H$

Value after reset: 0000 0000$_H$ This register is initialized by a power-up reset PURES (power-on clear or debugger reset).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CLMA3 TESEN	CLMA2 TESEN	CLMA1 TESEN	CLMA0 TESEN	ERR MSK	MONCL KMSK	RES CLM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.23 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	CLMA3TESEN	RH850/F1KH-D8, RH850/F1KM-S4: CLMA3 Self-Test Enable/Disable 0: Test disabled 1: Test enabled RH850/F1KM-S1: When read, the value after reset is returned. When writing, write the value after reset.
5	CLMA2TESEN	RH850/F1KH-D8, RH850/F1KM-S4: CLMA2 Self-Test Enable/Disable 0: Test disabled 1: Test enabled RH850/F1KM-S1: CLMA3 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
4	CLMA1TESEN	CLMA1 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
3	CLMA0TESEN	CLMA0 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
2	ERRMSK	CLMA Test Error Mask Setting Masks a reset request to the reset controller when CLMAn detects an error. When the ERRMSK is set for CLMAn, that CLMAn does not issue a reset request to the reset controller even if it detects an error. The ERRMSK setting is valid only for the CLMAn for which CLMAnTESEN bit is set to 1. 0: Reset request signal generation enabled 1: Reset request signal generation disabled (masked)

Table 13.23 CLMATEST Register Contents

Bit Position	Bit Name	Function
1	MONCLKMSK	Monitor Clock Mask Setting Fixes the clock input to the CLMAn to low level. The MONCLKMSK setting is valid only for the CLMAn for which CLMAnTESEN bit is set to 1. 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM	CLMAn Test Reset Signal Control Initializes CLMAn forcibly. The RESCLM setting is valid only for the CLMAn for which CLMAnTESEN bit is set to 1. 0: Reset released 1: Reset executed

13.5.6 CLMATESTS — CLMA Test Status Register

This register is used to confirm the self-test result of CLMA0, CLMA1, CLMA2, and CLMA3.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	FFF8 C104$_H$
Value after reset:	0000 0000$_H$ This register is initialized by a power-up reset PURES (power-on clear or debugger reset).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLMA3 ERRS	CLMA2 ERRS	CLMA1 ERRS	CLMA0 ERRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.24 CLMATESTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	CLMA3ERRS	RH850/F1KH-D8, RH850/F1KM-S4: CLMA3 Error Status 0: Errors are not detected 1: Errors are detected RH850/F1KM-S1: When read, the value after reset is returned.
2	CLMA2ERRS	RH850/F1KH-D8, RH850/F1KM-S4: CLMA2 Error Status 0: Errors are not detected 1: Errors are detected RH850/F1KM-S1: CLMA3 Error Status 0: Errors are not detected 1: Errors are detected
1	CLMA1ERRS	CLMA1 Error Status 0: Errors are not detected 1: Errors are detected
0	CLMA0ERRS	CLMA0 Error Status 0: Errors are not detected 1: Errors are detected

13.5.7 CLMAnEMU0 — CLMAn Emulation Register 0

This register intentionally generates CLMAn error at emulation.

This register can be accessed only during break by debugger and is reset by break release.

Access: This register can be read or written in 8-bit units.

Address: <CLMAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLMAnSLFST	CLMAnSLSLW
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 13.25 CLMAnEMU0 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CLMAnSLFST	Specifies whether $f_{CLMATMON}$ is assumed to be high. 0: CLMATMON is assumed to be within the normal frequency range. 1: CLMATMON is assumed to exceed the upper threshold.
0	CLMAnSLSLW	Specifies whether $f_{CLMATMON}$ is assumed to be low. 0: CLMATMON is assumed to be within the normal frequency range. 1: CLMATMON is assumed to fall below the lower threshold.

CAUTION

It is prohibited to emulate a low and high CLMATMON at the same time. Thus CLMAnEMU0 must not be set to 03_H.

13.6 Usage Notes for CLMAn

Do not use a clock for which the CLMAn bit indicates an abnormality. The behavior and performance are not guaranteed if such a clock is used. When CLMA0 detects clock abnormality, modifying the clock domain settings is prohibited.

Table 13.26, Usage Notes for CLMAn provides usage notes for each CLMAn.

Table 13.26 Usage Notes for CLMAn

Monitor Clock	CPU Clock after CLMA Reset Release	Note
HS IntOSC (CLMA0)	EMCLK*1	Set ROSCE.ROSCDISTRG to 1.*1 Do not set the control registers of the MainOSC, the PLL0 and the PLL1. After $\overline{\text{CLMA0RES}}$ occurs, modifying any of the clock domain settings is prohibited.
MainOSC (CLMA1)	EMCLK (= HS IntOSC)	Do not set control registers of the MainOSC, the PLL0 (when it was a MainOSC source clock) and the PLL1 (when it was a MainOSC source clock). After occurrence of the $\overline{\text{CLMA1RES}}$, do not select a clock whose source clock is the MainOSC, the PLL0 or the PLL1.
PLL0OUT (CLMA2)	EMCLK (= HS IntOSC)	Do not set control registers of the PLL0. After occurrence of the $\overline{\text{CLMA2RES}}$, do not select a clock whose source clock is the PLL0.
PPLLOUT (CLMA3)	EMCLK (= HS IntOSC)	Do not set control registers of the PLL1. After occurrence of the $\overline{\text{CLMA3RES}}$, do not select a clock whose source clock is the PLL1.

Note 1. The state of EMCLK after reset by $\overline{\text{CLMA0RES}}$ depends on the state of HS IntOSC oscillation. If HS IntOSC is completely stopped, LS IntOSC is supplied as EMCLK. If HS IntOSC continues oscillating, the HS IntOSC will be supplied as the clock source of EMCLK regardless of the frequency of the HS IntOSC. Therefore, it is necessary to change the source clock of EMCLK to LS IntOSC. After setting the ROSCE.ROSCDISTRG bit to 1, EMCLK is switched from HS IntOSC to LS IntOSC.

Section 14 Stand-By Controller (STBC)

This section describes the functions, registers, and various stand-by modes of the stand-by controller (STBC).

14.1 Functions

14.1.1 Types of Stand-By Mode

The RH850/F1KH, RH850/F1KM supports STOP mode and DeepSTOP mode for system-level low power status. In addition, the RH850/F1KH, RH850/F1KM supports cyclic operation (Cyclic RUN mode and Cyclic STOP mode) which supports low-power operation of limited functions. Transition between each mode is described in the **Section 14.1.5, Transition to Stand-By Mode**.

- RUN mode
  RUN mode is a normal operation mode where the CPU is operating and all of other modules can operate. The CPU can enter “HALT” state by executing the “HALT” instruction to stop its operation in this mode.
- STOP mode
  STOP mode is a chip-level stand-by mode in which the clock supply to a certain clock domain can be stopped.
  STOP mode is entered when the STBC0STPT.STBC0STPTRG bit is set to 1.
  The clock supply to clock domains can continue even in STOP mode by setting CKSC_xxx_STPM.xxxSTPMSK = 1. For details on the CKSC_xxx_STPM register, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**, **Section 12C, Clock Controller of RH850/F1KM-S1**.
- DeepSTOP mode
  DeepSTOP mode is a chip-level stand-by mode to reduce power consumption further than STOP mode. In addition to the clock supply stop, the power supply to the Isolated area (ISO area) is switched off.
  DeepSTOP mode is entered when the STBC0PSC.STBC0DISTRG is set to 1.
- Cyclic RUN mode
  Cyclic RUN mode is a low-power operation mode in which limited modules can operate at low speed. In this mode, only the CPU1, the peripheral functions in the Always-On area (AWO area), RLIN3 and CSIG0 can operate.
  Data/Code flash memory and PLL are not available. The CPU2 is also not available.
  The CPU1 executes the instructions in the retention RAM.
  In this mode, the CPU1 can issue the “HALT” instruction to enter HALT state.
  The mode transition to Cyclic RUN mode from DeepSTOP mode is triggered by wake-up factor 2, and the mode transition to Cyclic RUN mode from Cyclic STOP mode is triggered by either wake-up factor 1 or wake-up factor 2.
  CSIG0 cannot be wake-up factor.
- Cyclic STOP mode
  Cyclic STOP mode is a STOP mode in cyclic operation, and the CPU1 halts its operation.
  This mode is entered by setting the STBC0STPT.STBC0STPTRG bit to 1 in Cyclic RUN mode.

14.1.2 Wake-Up Control

14.1.2.1 Wake-Up Factors for Stand-By Modes

The stand-by controller can initiate return from stand-by mode by the following wake-up factors.

Table 14.1 Wake-Up Factor 1 (RH850/F1KH-D8)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*[1]	Cyclic STOP → RUN*[1]
TNMI	Port	✓	✓	✓	✓
WDTA0NMI	WDTA0	✓	✓	✓	✓
INTLVIL*[3]	LVI	✓	✓	✓	✓
INTP0	Port	✓	✓	✓	✓
INTP1	Port	✓	✓	✓	✓
INTP2	Port	✓	✓	✓	✓
INTWDTA0	WDTA0	✓	✓	✓	✓
INTP3	Port	✓	✓	✓	✓
INTP4	Port	✓	✓	✓	✓
INTP5	Port	✓	✓	✓	✓
INTP10	Port	✓	✓	✓	✓
INTP11	Port	✓	✓	✓	✓
WUTRG1	LPS0	✓	✓	✓	✓
INTTAUJ0I0	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓	✓	✓
WUTRG0	LPS0	✓	✓	✓	✓
INTP6	Port	✓	✓	✓	✓
INTP7	Port	✓	✓	✓	✓
INTP8	Port	✓	✓	✓	✓
INTP12	Port	✓	✓	✓	✓
INTP9	Port	✓	✓	✓	✓
INTP13	Port	✓	✓	✓	✓
INTP14	Port	✓	✓	✓	✓
INTP15	Port	✓	✓	✓	✓
INTRTCA01S	RTCA0	✓	✓	✓	✓
INTRTCA0AL	RTCA0	✓	✓	✓	✓
INTRTCA0R	RTCA0	✓	✓	✓	✓
INTDCUTDI	JTAG	✓	✓	✓	✓
INTP16	Port	✓	✓	✓	✓
INTP17	Port	✓	✓	✓	✓
INTP18	Port	✓	✓	✓	✓
INTP19	Port	✓	✓	✓	✓
INTP20	Port	✓	✓	✓	✓
INTP21	Port	✓	✓	✓	✓
INTP22	Port	✓	✓	✓	✓

Table 14.1 Wake-Up Factor 1 (RH850/F1KH-D8)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*1	Cyclic STOP → RUN*1
INTP23	Port	✓	✓	✓	✓
INTTAUJ2I0	TAUJ2	✓	✓	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓	✓	✓
INTKR0	KR0	✓	—	—	—
INTRCANGRECC0*2	RCFDC0	✓	—	—	—
INTRCAN0REC*2	RCFDC0	✓	—	—	—
INTRCAN1REC*2	RCFDC0	✓	—	—	—
INTRCAN2REC*2	RCFDC0	✓	—	—	—
INTRCAN3REC*2	RCFDC0	✓	—	—	—
INTRCAN4REC*2	RCFDC0	✓	—	—	—
INTRCAN5REC*2	RCFDC0	✓	—	—	—
INTRCAN6REC*2	RCFDC0	✓	—	—	—
INTRCAN7REC*2	RCFDC0	✓	—	—	—
INTRCANGRECC1*2	RCFDC1	✓	—	—	—
INTRCAN8REC*2	RCFDC1	✓	—	—	—
INTRCAN9REC*2	RCFDC1	✓	—	—	—
INTRCAN10REC*2	RCFDC1	✓	—	—	—
INTRCAN11REC*2	RCFDC1	✓	—	—	—

Note 1. The mode returns to RUN mode via DeepSTOP mode. When the transition from Cyclic STOP to Cyclic RUN is made by wake-up factor 1, if the transition to DeepSTOP by STBC0PSC.STBC0DISTRG is made without clearing wake-up factor 1, the transition to RUN mode is made.

Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from stand-by modes such as DeepSTOP is possible. As the trigger for waking up from DeepSTOP, use a pin of port P0, P1, P2, P3, or P8, which is assigned to the Always-On area (AWO area).
To clear the wake-up factor flag, the interrupt request must be cleared in each CAN module.

Note 3. Cannot be cleared while REG0VCC is below the reference voltage (REG0VCC (min.) < VLVIn).
To clear the wake-up factor flag (WUF0[2]), WUFC0[2] must be set while WUFMSK0[2] = 1 and REG0VCC is above the reference voltage (REG0VCC (min.) > VLVIn). The INTLVIH interrupt can be used to check that REG0VCC is above the reference voltage.

Table 14.2 Wake-Up Factor 1 (RH850/F1KM-S4)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*[1]	Cyclic STOP → RUN*[1]
TNMI	Port	✓	✓	✓	✓
WDTA0NMI	WDTA0	✓	✓	✓	✓
INTLVIL*[3]	LVI	✓	✓	✓	✓
INTP0	Port	✓	✓	✓	✓
INTP1	Port	✓	✓	✓	✓
INTP2	Port	✓	✓	✓	✓
INTWDTA0	WDTA0	✓	✓	✓	✓
INTP3	Port	✓	✓	✓	✓
INTP4	Port	✓	✓	✓	✓
INTP5	Port	✓	✓	✓	✓
INTP10	Port	✓	✓	✓	✓
INTP11	Port	✓	✓	✓	✓
WUTRG1	LPS0	✓	✓	✓	✓
INTTAUJ0I0	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓	✓	✓
WUTRG0	LPS0	✓	✓	✓	✓
INTP6	Port	✓	✓	✓	✓
INTP7	Port	✓	✓	✓	✓
INTP8	Port	✓	✓	✓	✓
INTP12	Port	✓	✓	✓	✓
INTP9	Port	✓	✓	✓	✓
INTP13	Port	✓	✓	✓	✓
INTP14	Port	✓	✓	✓	✓
INTP15	Port	✓	✓	✓	✓
INTRTCA01S	RTCA0	✓	✓	✓	✓
INTRTCA0AL	RTCA0	✓	✓	✓	✓
INTRTCA0R	RTCA0	✓	✓	✓	✓
INTDCUTDI	JTAG	✓	✓	✓	✓
INTP16	Port	✓	✓	✓	✓
INTP17	Port	✓	✓	✓	✓
INTP18	Port	✓	✓	✓	✓
INTP19	Port	✓	✓	✓	✓
INTP20	Port	✓	✓	✓	✓
INTP21	Port	✓	✓	✓	✓
INTP22	Port	✓	✓	✓	✓

Table 14.2 Wake-Up Factor 1 (RH850/F1KM-S4)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*1	Cyclic STOP → RUN*1
INTP23	Port	✓	✓	✓	✓
INTTAUJ2I0	TAUJ2	✓	✓	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓	✓	✓
INTKR0	KR0	✓	—	—	—
INTRCANGRECC0*2	RCFDC0	✓	—	—	—
INTRCAN0REC*2	RCFDC0	✓	—	—	—
INTRCAN1REC*2	RCFDC0	✓	—	—	—
INTRCAN2REC*2	RCFDC0	✓	—	—	—
INTRCAN3REC*2	RCFDC0	✓	—	—	—
INTRCAN4REC*2	RCFDC0	✓	—	—	—
INTRCAN5REC*2	RCFDC0	✓	—	—	—
INTRCAN6REC*2	RCFDC0	✓	—	—	—
INTRCAN7REC*2	RCFDC0	✓	—	—	—

Note 1. The mode returns to RUN mode via DeepSTOP mode. When the transition from Cyclic STOP to Cyclic RUN is made by wake-up factor 1, if the transition to DeepSTOP by STBC0PSC.STBC0DISTRG is made without clearing wake-up factor 1, the transition to RUN mode is made.

Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from stand-by modes such as DeepSTOP is possible. As the trigger for waking up from DeepSTOP, use a pin of port P0, P1, P2, or P8, which is assigned to the Always-On area (AWO area).
To clear the wake-up factor flag, the interrupt request must be cleared in each CAN module.

Note 3. Cannot be cleared while REGVCC is below the reference voltage (REGVCC (min.) < VLVIn).
To clear the wake-up factor flag (WUF0[2]), WUFC0[2] must be set while WUFMSK0[2] = 1 and REGVCC is above the reference voltage (REGVCC (min.) > VLVIn). The INTLVIH interrupt can be used to check that REGVCC is above the reference voltage.

Table 14.3 Wake-Up Factor 1 (RH850/F1KM-S1)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*1	Cyclic STOP → RUN*1
TNMI	Port	✓	✓	✓	✓
WDTA0NMI	WDTA0	✓	✓	✓	✓
INTLVIL*3	LVI	✓	✓	✓	✓
INTP0	Port	✓	✓	✓	✓
INTP1	Port	✓	✓	✓	✓
INTP2	Port	✓	✓	✓	✓
INTWDTA0	WDTA0	✓	✓	✓	✓
INTP3	Port	✓	✓	✓	✓
INTP4	Port	✓	✓	✓	✓
INTP5	Port	✓	✓	✓	✓
INTP10	Port	✓	✓	✓	✓
INTP11	Port	✓	✓	✓	✓
WUTRG1	LPS0	✓	✓	✓	✓
INTTAUJ0I0	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓	✓	✓

Table 14.3 Wake-Up Factor 1 (RH850/F1KM-S1)

Wake-Up Factor	Unit	STOP → RUN	DeepSTOP → RUN	Cyclic RUN → RUN*1	Cyclic STOP → RUN*1
INTTAUJ0I2	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓	✓	✓
WUTRG0	LPS0	✓	✓	✓	✓
INTP6	Port	✓	✓	✓	✓
INTP7	Port	✓	✓	✓	✓
INTP8	Port	✓	✓	✓	✓
INTP12	Port	✓	✓	✓	✓
INTP13	Port	✓	—	✓	✓
INTRTCA01S	RTCA0	✓	✓	✓	✓
INTRTCA0AL	RTCA0	✓	✓	✓	✓
INTRTCA0R	RTCA0	✓	✓	✓	✓
INTDCUTDI	JTAG	✓	✓	✓	✓
INTTAUJ2I0	TAUJ2	✓	✓	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓	✓	✓
INTKR0	KR0	✓	—	—	—
INTRCANGRECC0*2	RCFDC0	✓	—	—	—
INTRCAN0REC*2	RCFDC0	✓	—	—	—
INTRCAN1REC*2	RCFDC0	✓	—	—	—
INTRCAN2REC*2	RCFDC0	✓	—	—	—
INTRCAN3REC*2	RCFDC0	✓	—	—	—
INTRCAN4REC*2	RCFDC0	✓	—	—	—
INTRCAN5REC*2	RCFDC0	✓	—	—	—

Note 1. The mode returns to RUN mode via DeepSTOP mode. When the transition from Cyclic STOP to Cyclic RUN is made by wake-up factor 1, if the transition to DeepSTOP by STBC0PSC.STBC0DISTRG is made without clearing wake-up factor 1, the transition to RUN mode is made.

Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from stand-by modes such as DeepSTOP is possible. As the trigger for waking up from DeepSTOP, use a pin of port P0, which is assigned to the Always-On area (AWO area).
To clear the wake-up factor flag, the interrupt request must be cleared in each CAN module.

Note 3. Cannot be cleared while REGVCC is below the reference voltage (REGVCC (min.) < VLVIn).
To clear the wake-up factor flag (WUF0[2]), WUFC0[2] must be set while WUFMSK0[2] = 1 and REGVCC is above the reference voltage (REGVCC (min.) > VLVIn). The INTLVIH interrupt can be used to check that REGVCC is above the reference voltage.

Table 14.4 Wake-Up Factor 2 (RH850/F1KH-D8)

Wake-Up Factor	Unit	DeepSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	✓*1	✓*1
INTADCA0I1	ADCA0	✓*1	✓*1
INTADCA0I2	ADCA0	✓*1	✓*1
INTRLIN30	RLIN30	—	✓
INTTAUJ0I0	TAUJ0	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓
INTRLIN31	RLIN31	—	✓
INTRLIN32	RLIN32	—	✓
INTRTCA01S	RTCA0	✓	✓
INTRTCA0AL	RTCA0	✓	✓
INTRTCA0R	RTCA0	✓	✓
INTRLIN33	RLIN33	—	✓
INTRLIN34	RLIN34	—	✓
INTRLIN35	RLIN35	—	✓
INTRLIN36	RLIN36	—	✓
INTRLIN37	RLIN37	—	✓
INTTAUJ2I0	TAUJ2	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓

Note 1. These wake-up factors are only available in LPS analog input mode.

Table 14.5 Wake-Up Factor 2 (RH850/F1KM-S4)

Wake-Up Factor	Unit	DeepSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	✓*1	✓*1
INTADCA0I1	ADCA0	✓*1	✓*1
INTADCA0I2	ADCA0	✓*1	✓*1
INTRLIN30	RLIN30	—	✓
INTTAUJ0I0	TAUJ0	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓
INTRLIN31	RLIN31	—	✓
INTRLIN32	RLIN32	—	✓
INTRTCA01S	RTCA0	✓	✓
INTRTCA0AL	RTCA0	✓	✓
INTRTCA0R	RTCA0	✓	✓
INTRLIN33	RLIN33	—	✓
INTRLIN34	RLIN34	—	✓
INTRLIN35	RLIN35	—	✓
INTRLIN36	RLIN36	—	✓
INTRLIN37	RLIN37	—	✓
INTTAUJ2I0	TAUJ2	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓

Note 1. These wake-up factors are only available in LPS analog input mode.

Table 14.6 Wake-Up Factor 2 (RH850/F1KM-S1)

Wake-Up Factor	Unit	DeepSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	✓*1	✓*1
INTADCA0I1	ADCA0	✓*1	✓*1
INTADCA0I2	ADCA0	✓*1	✓*1
INTRLIN30	RLIN30	—	✓
INTTAUJ0I0	TAUJ0	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓
INTRLIN31	RLIN31	—	✓
INTRLIN32	RLIN32	—	✓
INTRTCA01S	RTCA0	✓	✓
INTRTCA0AL	RTCA0	✓	✓
INTRTCA0R	RTCA0	✓	✓
INTRLIN33	RLIN33	—	✓
INTTAUJ2I0	TAUJ2	✓	✓
INTTAUJ2I1	TAUJ2	✓	✓
INTTAUJ2I2	TAUJ2	✓	✓
INTTAUJ2I3	TAUJ2	✓	✓

Note 1. These wake-up factors are only available in LPS analog input mode.

CAUTION

For the pins of the function used for the wake-up factors from DeepSTOP, use the multiplexed functions of the ports assigned to the Always-On area (AWO area).

14.1.2.2 Setting of Wake-Up Factors

Wake-up factors for returning from stand-by modes are controlled by the following stand-by controller registers:

- Wake-up factor registers: WUF0, WUF1, WUF20, WUF_ISO0

  Upon occurrence of an effective wake-up factor, the associated wake-up factor flag is set to 1. By checking these registers and their flags, it is possible to identify the wake-up factor.

- Wake-up factor mask registers: WUFMSK0, WUFMSK1, WUFMSK20, WUFMSK_ISO0

  Each bit of these registers is assigned to a certain wake-up factor. Wake-up by this factor is enabled if its mask bit is set to 0. Wake-up factors assigned to both wake-up factor 1 and 2 should not be enabled at the same time.

- Wake-up factor clear registers: WUFC0, WUFC1, WUFC20, WUFC_ISO0

  By setting the applicable bits in these registers to 1, the wake-up factor bit (WUFy) in the wake-up factor registers (WUF0, WUF1, WUF20, WUF_ISO0) can be cleared.

NOTE

The wake-up factor flags in the wake-up factor registers (WUF0, WUF1, WUF20, and WUF_ISO0) only indicate the occurrence of wake-up factor. These flags do not indicate a transition from stand-by mode to normal mode.

The assignment of the wake-up factors to the control register bits and status register bits are shown in the following tables.

For details about the wake-up control and status registers, see **Section 14.2.2.3, WUF0/WUF1/WUF20/WUF_ISO0 — Wake-Up Factor Registers**, **Section 14.2.2.4, WUFMSK0/WUFMSK1/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers**, and **Section 14.2.2.5, WUFC0/WUFC1/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers**.

Table 14.7 Wake-Up Factor 1 Register Assignment (RH850/F1KH-D8)

Wake-Up Factor	Register Bit Assignment			Unit	176 Pins	233 Pins	272 Pins	324 Pins
TNMI	WUF0[0]	WUFMSK0[0]	WUFC0[0]	Port	✓	✓	—	✓
WDTA0NMI	WUF0[1]	WUFMSK0[1]	WUFC0[1]	WDTA0	✓	✓	—	✓
INTLVIL	WUF0[2]	WUFMSK0[2]	WUFC0[2]	LVI	✓	✓	—	✓
INTP0	WUF0[5]	WUFMSK0[5]	WUFC0[5]	Port	✓	✓	—	✓
INTP1	WUF0[6]	WUFMSK0[6]	WUFC0[6]	Port	✓	✓	—	✓
INTP2	WUF0[7]	WUFMSK0[7]	WUFC0[7]	Port	✓	✓	—	✓
INTWDTA0	WUF0[8]	WUFMSK0[8]	WUFC0[8]	WDTA0	✓	✓	—	✓
INTP3	WUF0[9]	WUFMSK0[9]	WUFC0[9]	Port	✓	✓	—	✓
INTP4	WUF0[10]	WUFMSK0[10]	WUFC0[10]	Port	✓	✓	—	✓
INTP5	WUF0[11]	WUFMSK0[11]	WUFC0[11]	Port	✓	✓	—	✓
INTP10	WUF0[12]	WUFMSK0[12]	WUFC0[12]	Port	✓	✓	—	✓
INTP11	WUF0[13]	WUFMSK0[13]	WUFC0[13]	Port	✓	✓	—	✓
WUTRG1	WUF0[14]	WUFMSK0[14]	WUFC0[14]	LPS0	✓	✓	—	✓
INTTAUJ0I0	WUF0[15]	WUFMSK0[15]	WUFC0[15]	TAUJ0	✓	✓	—	✓
INTTAUJ0I1	WUF0[16]	WUFMSK0[16]	WUFC0[16]	TAUJ0	✓	✓	—	✓
INTTAUJ0I2	WUF0[17]	WUFMSK0[17]	WUFC0[17]	TAUJ0	✓	✓	—	✓
INTTAUJ0I3	WUF0[18]	WUFMSK0[18]	WUFC0[18]	TAUJ0	✓	✓	—	✓
WUTRG0	WUF0[19]	WUFMSK0[19]	WUFC0[19]	LPS0	✓	✓	—	✓
INTP6	WUF0[20]	WUFMSK0[20]	WUFC0[20]	Port	✓	✓	—	✓
INTP7	WUF0[21]	WUFMSK0[21]	WUFC0[21]	Port	✓	✓	—	✓
INTP8	WUF0[22]	WUFMSK0[22]	WUFC0[22]	Port	✓	✓	—	✓
INTP12	WUF0[23]	WUFMSK0[23]	WUFC0[23]	Port	✓	✓	—	✓
INTP9	WUF0[24]	WUFMSK0[24]	WUFC0[24]	Port	✓	✓	—	✓
INTP13	WUF0[25]	WUFMSK0[25]	WUFC0[25]	Port	✓	✓	—	✓
INTP14	WUF0[26]	WUFMSK0[26]	WUFC0[26]	Port	✓	✓	—	✓
INTP15	WUF0[27]	WUFMSK0[27]	WUFC0[27]	Port	✓	✓	—	✓
INTRTCA01S	WUF0[28]	WUFMSK0[28]	WUFC0[28]	RTCA0	✓	✓	—	✓
INTRTCA0AL	WUF0[29]	WUFMSK0[29]	WUFC0[29]	RTCA0	✓	✓	—	✓
INTRTCA0R	WUF0[30]	WUFMSK0[30]	WUFC0[30]	RTCA0	✓	✓	—	✓
INTDCUTDI	WUF0[31]	WUFMSK0[31]	WUFC0[31]	JTAG	✓	✓	—	✓
INTP16	WUF1[0]	WUFMSK1[0]	WUFC1[0]	Port	✓	✓	—	✓
INTP17	WUF1[1]	WUFMSK1[1]	WUFC1[1]	Port	✓	✓	—	✓
INTP18	WUF1[2]	WUFMSK1[2]	WUFC1[2]	Port	✓	✓	—	✓
INTP19	WUF1[3]	WUFMSK1[3]	WUFC1[3]	Port	✓	✓	—	✓
INTP20	WUF1[4]	WUFMSK1[4]	WUFC1[4]	Port	✓	✓	—	✓
INTP21	WUF1[5]	WUFMSK1[5]	WUFC1[5]	Port	✓	✓	—	✓
INTP22	WUF1[6]	WUFMSK1[6]	WUFC1[6]	Port	✓	✓	—	✓
INTP23	WUF1[7]	WUFMSK1[7]	WUFC1[7]	Port	✓	✓	—	✓
INTTAUJ2I0	WUF1[8]	WUFMSK1[8]	WUFC1[8]	TAUJ2	✓	✓	—	✓
INTTAUJ2I1	WUF1[9]	WUFMSK1[9]	WUFC1[9]	TAUJ2	✓	✓	—	✓
INTTAUJ2I2	WUF1[10]	WUFMSK1[10]	WUFC1[10]	TAUJ2	✓	✓	—	✓
INTTAUJ2I3	WUF1[11]	WUFMSK1[11]	WUFC1[11]	TAUJ2	✓	✓	—	✓
INTKR0	WUF_ISO0[1]	WUFMSK_ISO0[1]	WUFC_ISO0[1]	KR0	✓	✓	—	✓
INTRCANGRECC0	WUF_ISO0[2]	WUFMSK_ISO0[2]	WUFC_ISO0[2]	RCFDC0	✓	✓	—	✓

Table 14.7 Wake-Up Factor 1 Register Assignment (RH850/F1KH-D8)

Wake-Up Factor	Register Bit Assignment			Unit	176 Pins	233 Pins	272 Pins	324 Pins
INTRCAN0REC	WUF_ISO0[3]	WUFMSK_ISO0[3]	WUFC_ISO0[3]	RCFDC0	✓	✓	—	✓
INTRCAN1REC	WUF_ISO0[4]	WUFMSK_ISO0[4]	WUFC_ISO0[4]	RCFDC0	✓	✓	—	✓
INTRCAN2REC	WUF_ISO0[5]	WUFMSK_ISO0[5]	WUFC_ISO0[5]	RCFDC0	✓	✓	—	✓
INTRCAN3REC	WUF_ISO0[6]	WUFMSK_ISO0[6]	WUFC_ISO0[6]	RCFDC0	✓	✓	—	✓
INTRCAN4REC	WUF_ISO0[7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RCFDC0	✓	✓	—	✓
INTRCAN5REC	WUF_ISO0[8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RCFDC0	✓	✓	—	✓
INTRCAN6REC	WUF_ISO0[10]	WUFMSK_ISO0[10]	WUFC_ISO0[10]	RCFDC0	✓	✓	—	✓
INTRCAN7REC	WUF_ISO0[11]	WUFMSK_ISO0[11]	WUFC_ISO0[11]	RCFDC0	✓	✓	—	✓
INTRCANGRECC1	WUF_ISO0[12]	WUFMSK_ISO0[12]	WUFC_ISO0[12]	RCFDC1	—	—	—	✓
INTRCAN8REC	WUF_ISO0[13]	WUFMSK_ISO0[13]	WUFC_ISO0[13]	RCFDC1	—	—	—	✓
INTRCAN9REC	WUF_ISO0[14]	WUFMSK_ISO0[14]	WUFC_ISO0[14]	RCFDC1	—	—	—	✓
INTRCAN10REC	WUF_ISO0[15]	WUFMSK_ISO0[15]	WUFC_ISO0[15]	RCFDC1	—	—	—	✓
INTRCAN11REC	WUF_ISO0[16]	WUFMSK_ISO0[16]	WUFC_ISO0[16]	RCFDC1	—	—	—	✓

Table 14.8 Wake-Up Factor 1 Register Assignment (RH850/F1KM-S4)

Wake-Up Factor	Register Bit Assignment			Unit	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
TNMI	WUF0[0]	WUFMSK0[0]	WUFC0[0]	Port	✓	✓	✓	✓	✓
WDTA0NMI	WUF0[1]	WUFMSK0[1]	WUFC0[1]	WDTA0	✓	✓	✓	✓	✓
INTLVIL	WUF0[2]	WUFMSK0[2]	WUFC0[2]	LVI	✓	✓	✓	✓	✓
INTP0	WUF0[5]	WUFMSK0[5]	WUFC0[5]	Port	✓	✓	✓	✓	✓
INTP1	WUF0[6]	WUFMSK0[6]	WUFC0[6]	Port	✓	✓	✓	✓	✓
INTP2	WUF0[7]	WUFMSK0[7]	WUFC0[7]	Port	✓	✓	✓	✓	✓
INTWDTA0	WUF0[8]	WUFMSK0[8]	WUFC0[8]	WDTA0	✓	✓	✓	✓	✓
INTP3	WUF0[9]	WUFMSK0[9]	WUFC0[9]	Port	✓	✓	✓	✓	✓
INTP4	WUF0[10]	WUFMSK0[10]	WUFC0[10]	Port	✓	✓	✓	✓	✓
INTP5	WUF0[11]	WUFMSK0[11]	WUFC0[11]	Port	✓	✓	✓	✓	✓
INTP10	WUF0[12]	WUFMSK0[12]	WUFC0[12]	Port	✓	✓	✓	✓	✓
INTP11	WUF0[13]	WUFMSK0[13]	WUFC0[13]	Port	✓	✓	✓	✓	✓
WUTRG1	WUF0[14]	WUFMSK0[14]	WUFC0[14]	LPS0	✓	✓	✓	✓	✓
INTTAUJ0I0	WUF0[15]	WUFMSK0[15]	WUFC0[15]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I1	WUF0[16]	WUFMSK0[16]	WUFC0[16]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I2	WUF0[17]	WUFMSK0[17]	WUFC0[17]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I3	WUF0[18]	WUFMSK0[18]	WUFC0[18]	TAUJ0	✓	✓	✓	✓	✓
WUTRG0	WUF0[19]	WUFMSK0[19]	WUFC0[19]	LPS0	✓	✓	✓	✓	✓
INTP6	WUF0[20]	WUFMSK0[20]	WUFC0[20]	Port	✓	✓	✓	✓	✓
INTP7	WUF0[21]	WUFMSK0[21]	WUFC0[21]	Port	✓	✓	✓	✓	✓
INTP8	WUF0[22]	WUFMSK0[22]	WUFC0[22]	Port	✓	✓	✓	✓	✓
INTP12	WUF0[23]	WUFMSK0[23]	WUFC0[23]	Port	✓	✓	✓	✓	✓
INTP9	WUF0[24]	WUFMSK0[24]	WUFC0[24]	Port	✓	✓	✓	✓	✓
INTP13	WUF0[25]	WUFMSK0[25]	WUFC0[25]	Port	✓	✓	✓	✓	✓
INTP14	WUF0[26]	WUFMSK0[26]	WUFC0[26]	Port	—	✓	✓	✓	✓
INTP15	WUF0[27]	WUFMSK0[27]	WUFC0[27]	Port	—	✓	✓	✓	✓
INTRTCA01S	WUF0[28]	WUFMSK0[28]	WUFC0[28]	RTCA0	✓	✓	✓	✓	✓
INTRTCA0AL	WUF0[29]	WUFMSK0[29]	WUFC0[29]	RTCA0	✓	✓	✓	✓	✓
INTRTCA0R	WUF0[30]	WUFMSK0[30]	WUFC0[30]	RTCA0	✓	✓	✓	✓	✓
INTDCUTDI	WUF0[31]	WUFMSK0[31]	WUFC0[31]	JTAG	✓	✓	✓	✓	✓
INTP16	WUF1[0]	WUFMSK1[0]	WUFC1[0]	Port	—	✓	✓	✓	✓
INTP17	WUF1[1]	WUFMSK1[1]	WUFC1[1]	Port	—	✓	✓	✓	✓
INTP18	WUF1[2]	WUFMSK1[2]	WUFC1[2]	Port	—	✓	✓	✓	✓
INTP19	WUF1[3]	WUFMSK1[3]	WUFC1[3]	Port	—	✓	✓	✓	✓
INTP20	WUF1[4]	WUFMSK1[4]	WUFC1[4]	Port	—	✓	✓	✓	✓
INTP21	WUF1[5]	WUFMSK1[5]	WUFC1[5]	Port	—	✓	✓	✓	✓
INTP22	WUF1[6]	WUFMSK1[6]	WUFC1[6]	Port	—	✓	✓	✓	✓
INTP23	WUF1[7]	WUFMSK1[7]	WUFC1[7]	Port	—	✓	✓	✓	✓
INTTAUJ2I0	WUF1[8]	WUFMSK1[8]	WUFC1[8]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I1	WUF1[9]	WUFMSK1[9]	WUFC1[9]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I2	WUF1[10]	WUFMSK1[10]	WUFC1[10]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I3	WUF1[11]	WUFMSK1[11]	WUFC1[11]	TAUJ2	✓	✓	✓	✓	✓
INTKR0	WUF_ISO0[1]	WUFMSK_ISO0[1]	WUFC_ISO0[1]	KR0	✓	✓	✓	✓	✓
INTRCANGRECC0	WUF_ISO0[2]	WUFMSK_ISO0[2]	WUFC_ISO0[2]	RCFDC0	✓	✓	✓	✓	✓

Table 14.8 Wake-Up Factor 1 Register Assignment (RH850/F1KM-S4)

Wake-Up Factor	Register Bit Assignment			Unit	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
INTRCAN0REC	WUF_ISO0[3]	WUFMSK_ISO0[3]	WUFC_ISO0[3]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN1REC	WUF_ISO0[4]	WUFMSK_ISO0[4]	WUFC_ISO0[4]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN2REC	WUF_ISO0[5]	WUFMSK_ISO0[5]	WUFC_ISO0[5]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN3REC	WUF_ISO0[6]	WUFMSK_ISO0[6]	WUFC_ISO0[6]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN4REC	WUF_ISO0[7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN5REC	WUF_ISO0[8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN6REC	WUF_ISO0[10]	WUFMSK_ISO0[10]	WUFC_ISO0[10]	RCFDC0	✓	✓	✓	✓	✓
INTRCAN7REC	WUF_ISO0[11]	WUFMSK_ISO0[11]	WUFC_ISO0[11]	RCFDC0	✓	✓	✓	✓	✓

Table 14.9 Wake-Up Factor 1 Register Assignment (RH850/F1KM-S1)

Wake-Up Factor	Register Bit Assignment			Unit	48 Pins	64 Pins	80 Pins	100 Pins
TNMI	WUF0[0]	WUFMSK0[0]	WUFC0[0]	Port	✓	✓	✓	✓
WDTA0NMI	WUF0[1]	WUFMSK0[1]	WUFC0[1]	WDTA0	✓	✓	✓	✓
INTLVIL	WUF0[2]	WUFMSK0[2]	WUFC0[2]	LVI	✓	✓	✓	✓
INTP0	WUF0[5]	WUFMSK0[5]	WUFC0[5]	Port	✓	✓	✓	✓
INTP1	WUF0[6]	WUFMSK0[6]	WUFC0[6]	Port	✓	✓	✓	✓
INTP2	WUF0[7]	WUFMSK0[7]	WUFC0[7]	Port	✓	✓	✓	✓
INTWDTA0	WUF0[8]	WUFMSK0[8]	WUFC0[8]	WDTA0	✓	✓	✓	✓
INTP3	WUF0[9]	WUFMSK0[9]	WUFC0[9]	Port	✓	✓	✓	✓
INTP4	WUF0[10]	WUFMSK0[10]	WUFC0[10]	Port	✓	✓	✓	✓
INTP5	WUF0[11]	WUFMSK0[11]	WUFC0[11]	Port	✓	✓	✓	✓
INTP10	WUF0[12]	WUFMSK0[12]	WUFC0[12]	Port	✓	✓	✓	✓
INTP11	WUF0[13]	WUFMSK0[13]	WUFC0[13]	Port	✓	✓	✓	✓
WUTRG1	WUF0[14]	WUFMSK0[14]	WUFC0[14]	LPS0	✓	✓	✓	✓
INTTAUJ0I0	WUF0[15]	WUFMSK0[15]	WUFC0[15]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	WUF0[16]	WUFMSK0[16]	WUFC0[16]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	WUF0[17]	WUFMSK0[17]	WUFC0[17]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	WUF0[18]	WUFMSK0[18]	WUFC0[18]	TAUJ0	✓	✓	✓	✓
WUTRG0	WUF0[19]	WUFMSK0[19]	WUFC0[19]	LPS0	✓	✓	✓	✓
INTP6	WUF0[20]	WUFMSK0[20]	WUFC0[20]	Port	—	—	✓	✓
INTP7	WUF0[21]	WUFMSK0[21]	WUFC0[21]	Port	—	—	✓	✓
INTP8	WUF0[22]	WUFMSK0[22]	WUFC0[22]	Port	—	—	✓	✓
INTP12	WUF0[23]	WUFMSK0[23]	WUFC0[23]	Port	—	—	✓	✓
INTP13	WUF0[25]	WUFMSK0[25]	WUFC0[25]	Port	—	—	—	✓
INTRTCA01S	WUF0[28]	WUFMSK0[28]	WUFC0[28]	RTCA0	✓	✓	✓	✓
INTRTCA0AL	WUF0[29]	WUFMSK0[29]	WUFC0[29]	RTCA0	✓	✓	✓	✓
INTRTCA0R	WUF0[30]	WUFMSK0[30]	WUFC0[30]	RTCA0	✓	✓	✓	✓
INTDCUTDI	WUF0[31]	WUFMSK0[31]	WUFC0[31]	JTAG	✓	✓	✓	✓
INTTAUJ2I0	WUF1[8]	WUFMSK1[8]	WUFC1[8]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I1	WUF1[9]	WUFMSK1[9]	WUFC1[9]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I2	WUF1[10]	WUFMSK1[10]	WUFC1[10]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I3	WUF1[11]	WUFMSK1[11]	WUFC1[11]	TAUJ2	✓	✓	✓	✓
INTKR0	WUF_ISO0[1]	WUFMSK_ISO0[1]	WUFC_ISO0[1]	KR0	✓	✓	✓	✓
INTRCANGRECC0	WUF_ISO0[2]	WUFMSK_ISO0[2]	WUFC_ISO0[2]	RCFDC0	✓	✓	✓	✓

Table 14.9 Wake-Up Factor 1 Register Assignment (RH850/F1KM-S1)

Wake-Up Factor	Register Bit Assignment			Unit	48 Pins	64 Pins	80 Pins	100 Pins
INTRCAN0REC	WUF_ISO0[3]	WUFMSK_ISO0[3]	WUFC_ISO0[3]	RCFDC0	✓	✓	✓	✓
INTRCAN1REC	WUF_ISO0[4]	WUFMSK_ISO0[4]	WUFC_ISO0[4]	RCFDC0	—	✓	✓	✓
INTRCAN2REC	WUF_ISO0[5]	WUFMSK_ISO0[5]	WUFC_ISO0[5]	RCFDC0	—	✓	✓	✓
INTRCAN3REC	WUF_ISO0[6]	WUFMSK_ISO0[6]	WUFC_ISO0[6]	RCFDC0	—	—	—	✓
INTRCAN4REC	WUF_ISO0[7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RCFDC0	—	—	—	✓
INTRCAN5REC	WUF_ISO0[8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RCFDC0	—	—	—	✓

Table 14.10 Wake-Up Factor 2 Register Assignment (RH850/F1KH-D8)

Wake-Up Factor	Register Bit Assignment			Unit	176 Pins	233 Pins	272 Pins	324 Pins
INTADCA0I0	WUF20[0]	WUFMSK20[0]	WUFC20[0]	ADCA0	✓	✓	—	✓
INTADCA0I1	WUF20[1]	WUFMSK20[1]	WUFC20[1]	ADCA0	✓	✓	—	✓
INTADCA0I2	WUF20[2]	WUFMSK20[2]	WUFC20[2]	ADCA0	✓	✓	—	✓
INTRLIN30	WUF20[3]	WUFMSK20[3]	WUFC20[3]	RLIN30	✓	✓	—	✓
INTTAUJ0I0	WUF20[4]	WUFMSK20[4]	WUFC20[4]	TAUJ0	✓	✓	—	✓
INTTAUJ0I1	WUF20[5]	WUFMSK20[5]	WUFC20[5]	TAUJ0	✓	✓	—	✓
INTTAUJ0I2	WUF20[6]	WUFMSK20[6]	WUFC20[6]	TAUJ0	✓	✓	—	✓
INTTAUJ0I3	WUF20[7]	WUFMSK20[7]	WUFC20[7]	TAUJ0	✓	✓	—	✓
INTRLIN31	WUF20[8]	WUFMSK20[8]	WUFC20[8]	RLIN31	✓	✓	—	✓
INTRLIN32	WUF20[9]	WUFMSK20[9]	WUFC20[9]	RLIN32	✓	✓	—	✓
INTRTCA01S	WUF20[10]	WUFMSK20[10]	WUFC20[10]	RTCA0	✓	✓	—	✓
INTRTCA0AL	WUF20[11]	WUFMSK20[11]	WUFC20[11]	RTCA0	✓	✓	—	✓
INTRTCA0R	WUF20[12]	WUFMSK20[12]	WUFC20[12]	RTCA0	✓	✓	—	✓
INTRLIN33	WUF20[13]	WUFMSK20[13]	WUFC20[13]	RLIN33	✓	✓	—	✓
INTRLIN34	WUF20[14]	WUFMSK20[14]	WUFC20[14]	RLIN34	✓	✓	—	✓
INTRLIN35	WUF20[15]	WUFMSK20[15]	WUFC20[15]	RLIN35	✓	✓	—	✓
INTRLIN36	WUF20[16]	WUFMSK20[16]	WUFC20[16]	RLIN36	✓	✓	—	✓
INTRLIN37	WUF20[17]	WUFMSK20[17]	WUFC20[17]	RLIN37	✓	✓	—	✓
INTTAUJ2I0	WUF20[18]	WUFMSK20[18]	WUFC20[18]	TAUJ2	✓	✓	—	✓
INTTAUJ2I1	WUF20[19]	WUFMSK20[19]	WUFC20[19]	TAUJ2	✓	✓	—	✓
INTTAUJ2I2	WUF20[20]	WUFMSK20[20]	WUFC20[20]	TAUJ2	✓	✓	—	✓
INTTAUJ2I3	WUF20[21]	WUFMSK20[21]	WUFC20[21]	TAUJ2	✓	✓	—	✓

Table 14.11 Wake-Up Factor 2 Register Assignment (RH850/F1KM-S4)

Wake-Up Factor	Register Bit Assignment			Unit	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
INTADCA0I0	WUF20[0]	WUFMSK20[0]	WUFC20[0]	ADCA0	✓	✓	✓	✓	✓
INTADCA0I1	WUF20[1]	WUFMSK20[1]	WUFC20[1]	ADCA0	✓	✓	✓	✓	✓
INTADCA0I2	WUF20[2]	WUFMSK20[2]	WUFC20[2]	ADCA0	✓	✓	✓	✓	✓
INTRLIN30	WUF20[3]	WUFMSK20[3]	WUFC20[3]	RLIN30	✓	✓	✓	✓	✓
INTTAUJ0I0	WUF20[4]	WUFMSK20[4]	WUFC20[4]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I1	WUF20[5]	WUFMSK20[5]	WUFC20[5]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I2	WUF20[6]	WUFMSK20[6]	WUFC20[6]	TAUJ0	✓	✓	✓	✓	✓
INTTAUJ0I3	WUF20[7]	WUFMSK20[7]	WUFC20[7]	TAUJ0	✓	✓	✓	✓	✓
INTRLIN31	WUF20[8]	WUFMSK20[8]	WUFC20[8]	RLIN31	✓	✓	✓	✓	✓
INTRLIN32	WUF20[9]	WUFMSK20[9]	WUFC20[9]	RLIN32	✓	✓	✓	✓	✓
INTRTCA01S	WUF20[10]	WUFMSK20[10]	WUFC20[10]	RTCA0	✓	✓	✓	✓	✓
INTRTCA0AL	WUF20[11]	WUFMSK20[11]	WUFC20[11]	RTCA0	✓	✓	✓	✓	✓
INTRTCA0R	WUF20[12]	WUFMSK20[12]	WUFC20[12]	RTCA0	✓	✓	✓	✓	✓
INTRLIN33	WUF20[13]	WUFMSK20[13]	WUFC20[13]	RLIN33	—	✓	✓	✓	✓
INTRLIN34	WUF20[14]	WUFMSK20[14]	WUFC20[14]	RLIN34	—	✓	✓	✓	✓
INTRLIN35	WUF20[15]	WUFMSK20[15]	WUFC20[15]	RLIN35	—	✓	✓	✓	✓
INTRLIN36	WUF20[16]	WUFMSK20[16]	WUFC20[16]	RLIN36	—	—	✓	✓	✓
INTRLIN37	WUF20[17]	WUFMSK20[17]	WUFC20[17]	RLIN37	—	—	✓	✓	✓
INTTAUJ2I0	WUF20[18]	WUFMSK20[18]	WUFC20[18]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I1	WUF20[19]	WUFMSK20[19]	WUFC20[19]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I2	WUF20[20]	WUFMSK20[20]	WUFC20[20]	TAUJ2	✓	✓	✓	✓	✓
INTTAUJ2I3	WUF20[21]	WUFMSK20[21]	WUFC20[21]	TAUJ2	✓	✓	✓	✓	✓

Table 14.12 Wake-Up Factor 2 Register Assignment (RH850/F1KM-S1)

Wake-Up Factor	Register Bit Assignment			Unit	48 Pins	64 Pins	80 Pins	100 Pins
INTADCA0I0	WUF20[0]	WUFMSK20[0]	WUFC20[0]	ADCA0	✓	✓	✓	✓
INTADCA0I1	WUF20[1]	WUFMSK20[1]	WUFC20[1]	ADCA0	✓	✓	✓	✓
INTADCA0I2	WUF20[2]	WUFMSK20[2]	WUFC20[2]	ADCA0	✓	✓	✓	✓
INTRLIN30	WUF20[3]	WUFMSK20[3]	WUFC20[3]	RLIN30	✓	✓	✓	✓
INTTAUJ0I0	WUF20[4]	WUFMSK20[4]	WUFC20[4]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	WUF20[5]	WUFMSK20[5]	WUFC20[5]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	WUF20[6]	WUFMSK20[6]	WUFC20[6]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	WUF20[7]	WUFMSK20[7]	WUFC20[7]	TAUJ0	✓	✓	✓	✓
INTRLIN31	WUF20[8]	WUFMSK20[8]	WUFC20[8]	RLIN31	—	✓	✓	✓
INTRLIN32	WUF20[9]	WUFMSK20[9]	WUFC20[9]	RLIN32	—	—	✓	✓
INTRTCA01S	WUF20[10]	WUFMSK20[10]	WUFC20[10]	RTCA0	✓	✓	✓	✓
INTRTCA0AL	WUF20[11]	WUFMSK20[11]	WUFC20[11]	RTCA0	✓	✓	✓	✓
INTRTCA0R	WUF20[12]	WUFMSK20[12]	WUFC20[12]	RTCA0	✓	✓	✓	✓
INTRLIN33	WUF20[13]	WUFMSK20[13]	WUFC20[13]	RLIN33	—	—	—	✓
INTTAUJ2I0	WUF20[18]	WUFMSK20[18]	WUFC20[18]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I1	WUF20[19]	WUFMSK20[19]	WUFC20[19]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I2	WUF20[20]	WUFMSK20[20]	WUFC20[20]	TAUJ2	✓	✓	✓	✓
INTTAUJ2I3	WUF20[21]	WUFMSK20[21]	WUFC20[21]	TAUJ2	✓	✓	✓	✓

14.1.3 On-Chip Debug Wake-Up

The On-Chip Debug unit (OCD) generates a wake-up event while the microcontroller runs the application program in the following cases:

- The debugger issues a stop request
- A breakpoint is hit

In either case all stand-by modes are terminated, provided the OCD debug event is enabled as a wake-up factor via the WUFMSK0 register.

CAUTION

If the OCD wake-up event is disabled, it is not possible to wake up from stand-by modes via an On-chip debugger request.
The OCD wake-up event can be enabled as a wake-up factor for all stand-by modes by setting WUFMSK0[31] = 0.
When the hot plug-in function is used, make sure to enable the OCD wake-up event and return from stand-by mode by INTDCUTDI interrupt.

14.1.4 I/O Buffer Control

This section describes the behavior of the I/O buffers during various stand-by modes.

The port groups in the Isolated area (ISO area) support the I/O buffer hold state. The port groups in the Always-On area (AWO area) remain their state before entering stand-by mode.
For details on the port group assignment to the Isolated area (ISO area) and to the Always-On area (AWO area), see **Section 10A, Power Supply Circuit of RH850/F1KH-D8**, **Section 10B, Power Supply Circuit of RH850/F1KM-S4**, **Section 10C, Power Supply Circuit of RH850/F1KM-S1**.

14.1.4.1 I/O Buffer Hold State

During the I/O buffer hold state, the I/O buffers maintains the state it was in before entering this state. Therefore, no external or internal signal can change the state of the I/O buffer until the I/O buffer hold state is terminated.

14.1.4.2 I/O Buffers during STOP Mode

In STOP mode, the I/O buffers remain their state before entering STOP mode (I/O buffer hold state is not entered).

14.1.4.3 I/O Buffers during DeepSTOP Mode

In DeepSTOP mode, the I/O buffers of port groups in the Isolated area (ISO area) transition to I/O buffer hold state.

After wake-up from DeepSTOP, the I/O buffers remain in I/O buffer hold state until the state is canceled by software. To cancel I/O buffer hold state, follow the steps shown below.

1. Re-configure the peripheral or port function.
2. Set IOHOLD.IOHOLD = 0.

The following table is a summary of the I/O buffer in the Isolated area (ISO area) during stand-by mode and after wake-up.

Table 14.13 Buffer Operation during Stand-by Mode and after Wake-Up (I/O buffers in the Isolated area (ISO area))

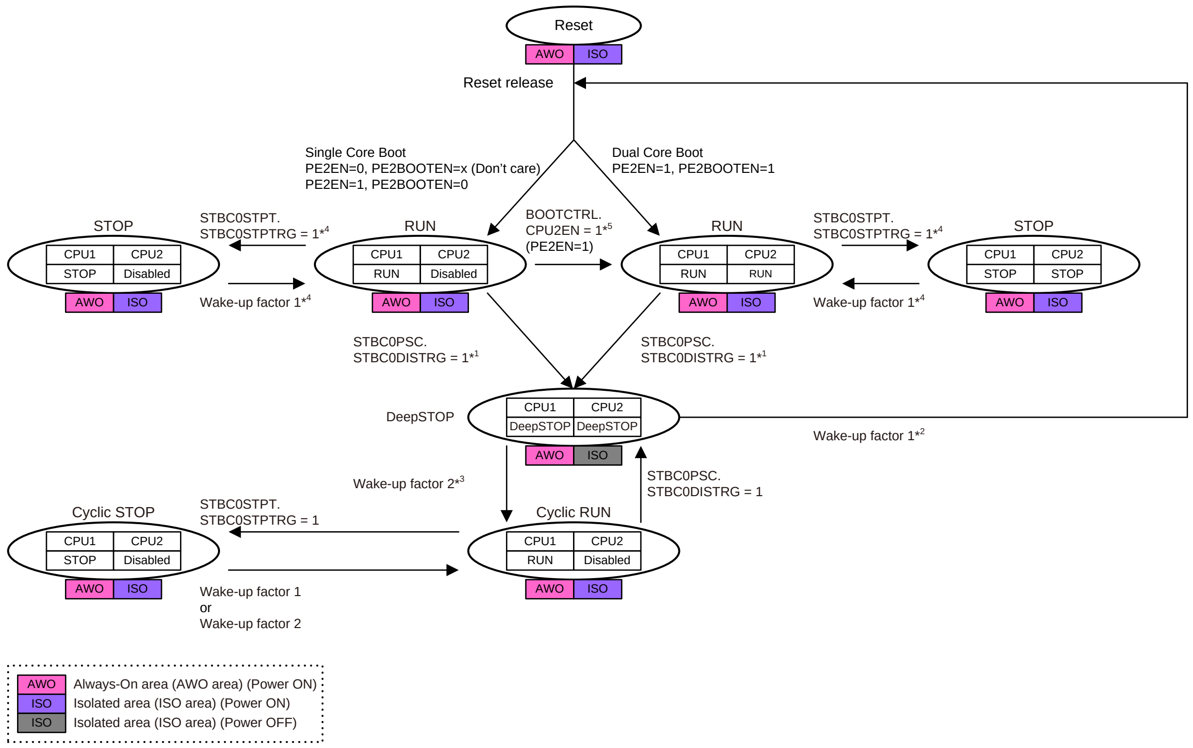
	Before Stand-By	During Stand-By	After Wake-Up
STOP mode	Normal operation		
DeepSTOP mode	Normal operation	I/O buffer hold state	I/O buffer hold state*[1]

Note 1. Set the IOHOLD.IOHOLD bit to “0” to release the I/O buffer hold state.

The port groups in the Always-On area (AWO area) don’t support I/O buffer hold state. They continue operation and remain its state before entering DeepSTOP. In the case an alternative function of modules in Isolated area (ISO area) is assigned to the pin in the Always-On area (AWO area), the state of the I/O buffer may change in the transition to the DeepSTOP due to initialization of the modules in the Isolated area (ISO area) by ISORES. To avoid this behavior, it is recommended to change to function of modules in Always-On area (AWO area) (e. g. Port mode) before entering DeepSTOP.

14.1.5 Transition to Stand-By Mode

The figure below shows transition between RUN mode and stand-by mode.



Note: The return time from stand-by mode can be shortened by setting the AWDTADSTPMSK bit of the CKSC_AWDTAD_STPM register to 1.

Note 1. Set a clock other than the PLL clock as the CPU clock source.

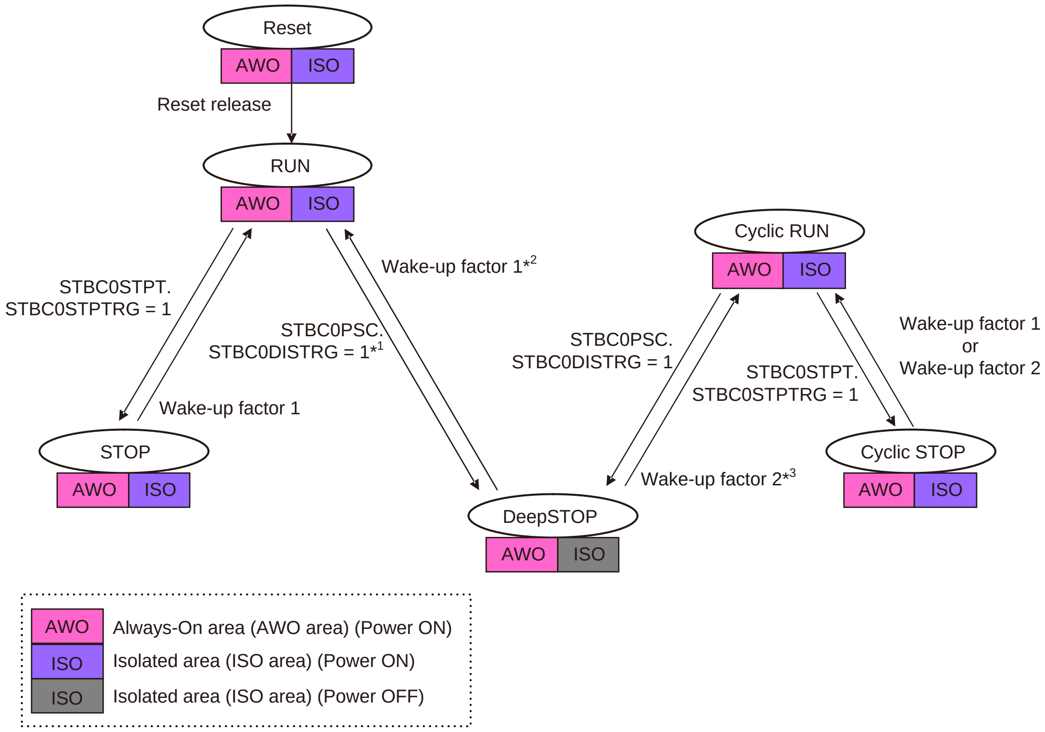
Note 2. To return from DeepSTOP to RUN, ISO peripheral function registers must be set again.
The CPU1/CPU2 starts the operation from the reset address on the code flash.

Note 3. The CPU1 starts the operation from the reset address on the retention RAM.

Note 4. Set a clock according to the method described in **Section 12AB.5.5, CPUCLK Setting in STOP Mode Method**.

Note 5. For details on the BOOTCTRL register, see **Section 3A.4, CPU2 Boot Up Operation**.

Figure 14.1 Transition to Stand-By Mode (RH850/F1KH-D8)



Note: The return time from stand-by mode can be shortened by setting the AWDTADSTPMSK bit of the CKSC_AWDTAD_STPM register to 1.

Note 1. Set a clock other than the PLL clock as the CPU clock source.

Note 2. To return from DeepSTOP to RUN, ISO peripheral function registers must be set again.
The CPU starts the operation from the reset address on the code flash.

Note 3. The CPU starts the operation from the reset address on the retention RAM.

Figure 14.2 Transition to Stand-By Mode (RH850/F1KM-S4, RH850/F1KM-S1)

14.1.6 Clock Supply

The clock supply to the stand-by controller is shown in the following table.

Table 14.14 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
Stand-by controller	Register access clock	CPUCLK_L, EMCLK

14.2 Registers

14.2.1 List of Registers

The following table lists the stand-by controller registers.

Table 14.15 List of Registers

Module Name	Register Name	Symbol	Address
STBC0	Power save control register	STBC0PSC	FFF8 0100_H
	Power stop trigger register	STBC0STPT	FFF8 0110_H
STBC_WUF0	Wake-up factor registers	WUF0	FFF8 0400_H
STBC_WUF1		WUF1	FFF8 0410_H
STBC_WUF20		WUF20	FFF8 0520_H
STBC_WUFISO		WUF_ISO0	FFF8 8110_H
STBC_WUF0	Wake-up factor mask registers	WUFMSK0	FFF8 0404_H
STBC_WUF1		WUFMSK1	FFF8 0414_H
STBC_WUF20		WUFMSK20	FFF8 0524_H
STBC_WUFISO		WUFMSK_ISO0	FFF8 8114_H
STBC_WUF0	Wake-up factor clear registers	WUFC0	FFF8 0408_H
STBC_WUF1		WUFC1	FFF8 0418_H
STBC_WUF20		WUFC20	FFF8 0528_H
STBC_WUFISO		WUFC_ISO0	FFF8 8118_H
STBC_IOHOLD	I/O buffer hold control register	IOHOLD	FFF8 $0B00_H$

14.2.2 Details of Stand-By Controller Control Registers

14.2.2.1 STBC0PSC — Power Save Control Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 0100_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0DISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 14.16 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	STBC0DISTRG	0: No effect 1: Transition to DeepSTOP mode This bit is cleared automatically after transition to the DeepSTOP mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

14.2.2.2 STBC0STPT — Power Stop Trigger Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 0110_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0STPTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 14.17 STBC0STPT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STBC0STPTRG	0: No effect. 1: Transition to STOP mode – In RUN mode: Transition to STOP mode – In Cyclic RUN mode: Transition to Cyclic STOP mode This bit is cleared automatically after transition to the STOP / Cyclic STOP mode.

14.2.2.3 WUF0/WUF1/WUF20/WUF_ISO0 — Wake-Up Factor Registers

These registers indicate the generation of wake-up factors.

WUF0, WUF1 and WUF20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUF_ISO0 is initialized by all reset sources (ISORES).

Access: These registers are read-only registers that can be read in 32-bit units.

Address: WUF0: FFF8 0400$_H$
WUF1: FFF8 0410$_H$
WUF20: FFF8 0520$_H$
WUF_ISO0: FFF8 8110$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUF31	WUF30	WUF29	WUF28	WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF09	WUF08	WUF07	WUF06	WUF05	WUF04	WUF03	WUF02	WUF01	WUF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.18 WUF0/WUF1/WUF20/WUF_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFy	Indicates the generation of a wake-up factor. 0: Wake-up factor is not generated 1: Wake-up factor is generated

NOTE

While the WUFMSKy bit in the wake-up factor mask register is 1, WUFy is not set to 1 at the generation of a wake-up factor.

Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 14.7, Wake-Up Factor 1 Register Assignment (RH850/F1KH-D8)**, **Table 14.8, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S4)**, **Table 14.9, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S1)**, **Table 14.10, Wake-Up Factor 2 Register Assignment (RH850/F1KH-D8)**, **Table 14.11, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S4)**, and **Table 14.12, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S1)**.

The bit to which a wake-up factor is not assigned is read as the value “0”.

14.2.2.4 WUFMSK0/WUFMSK1/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers

These registers enable wake-up factors.

WUFMSK0, WUFMSK1 and WUFMSK20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUFMSK_ISO0 is initialized by all reset sources (ISORES).

Access: These registers can be read or written in 32-bit units.

Address:
WUFMSK0: FFF8 0404$_H$
WUFMSK1: FFF8 0414$_H$
WUFMSK20: FFF8 0524$_H$
WUFMSK_ISO0: FFF8 8114$_H$

Value after reset:
WUFMSK0: FFFF FFFF$_H$
WUFMSK1: FFFF FFFF$_H$
WUFMSK20: FFFF FFFF$_H$
WUFMSK_ISO0: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFMSK31	WUFMSK30	WUFMSK29	WUFMSK28	WUFMSK27	WUFMSK26	WUFMSK25	WUFMSK24	WUFMSK23	WUFMSK22	WUFMSK21	WUFMSK20	WUFMSK19	WUFMSK18	WUFMSK17	WUFMSK16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFMSK15	WUFMSK14	WUFMSK13	WUFMSK12	WUFMSK11	WUFMSK10	WUFMSK09	WUFMSK08	WUFMSK07	WUFMSK06	WUFMSK05	WUFMSK04	WUFMSK03	WUFMSK02	WUFMSK01	WUFMSK00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.19 WUFMSK0/WUFMSK1/WUFMSK20/WUFMSK_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFMSKy	Enables/disables a wake-up factor. 0: Wake-up factor is enabled 1: Wake-up factor is disabled

NOTE

While the WUFMSKy bit is 1, WUFy of the wake-up factor register is not set to 1 at the generation of a wake-up factor.

Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 14.7, Wake-Up Factor 1 Register Assignment (RH850/F1KH-D8)**, **Table 14.8, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S4)**, **Table 14.9, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S1)**, **Table 14.10, Wake-Up Factor 2 Register Assignment (RH850/F1KH-D8)**, **Table 14.11, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S4)**, and **Table 14.12, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S1)**.

When writing to these registers, write the value “1” to the bits to which wake-up factors are not assigned.

14.2.2.5 WUFC0/WUFC1/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers

These registers clear the WUFy bits in the wake-up factor registers.

Access: These registers are write-only registers that can be written in 32-bit units.

Address: WUFC0: FFF8 0408$_H$

WUFC1: FFF8 0418$_H$

WUFC20: FFF8 0528$_H$

WUFC_ISO0: FFF8 8118$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFC 31	WUFC 30	WUFC 29	WUFC 28	WUFC 27	WUFC 26	WUFC 25	WUFC 24	WUFC 23	WUFC 22	WUFC 21	WUFC 20	WUFC 19	WUFC 18	WUFC 17	WUFC 16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFC 15	WUFC 14	WUFC 13	WUFC 12	WUFC 11	WUFC 10	WUFC 09	WUFC 08	WUFC 07	WUFC 06	WUFC 05	WUFC 04	WUFC 03	WUFC 02	WUFC 01	WUFC 00
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 14.20 WUFC0/WUFC1/WUFC20/WUFC_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFCy	Clears the wake-up factor bit WUFy in the wake-up factor registers. 0: WUFy is not modified 1: WUFy is cleared

Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 14.7, Wake-Up Factor 1 Register Assignment (RH850/F1KH-D8)**, **Table 14.8, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S4)**, **Table 14.9, Wake-Up Factor 1 Register Assignment (RH850/F1KM-S1)**, **Table 14.10, Wake-Up Factor 2 Register Assignment (RH850/F1KH-D8)**, **Table 14.11, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S4)**, and **Table 14.12, Wake-Up Factor 2 Register Assignment (RH850/F1KM-S1)**.

When writing to these registers, write the value “0” to the bits to which wake-up factors are not assigned.

14.2.2.6 IOHOLD — I/O Buffer Hold Control Register

This register specifies the hold state of the I/O buffer in DeepSTOP mode. The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access:	This register can be read or written in 32-bit units.
Address:	FFF8 0B00$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOHOLD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 14.21 IOHOLD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IOHOLD	0: I/O hold state is released 1: I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is prohibited. To release the I/O hold state after the wake-up, set this bit to 0 by software.

14.3 Mode Transition

This section explains the mode transition procedures.

CAUTION

Transition to stand-by mode should be performed by CPU1. When CPU1 shifts to stand-by mode, CPU2 will also shift to stand-by mode.

14.3.1 STOP Mode

In STOP mode, most of the clock supplies to the Always-On area (AWO area) and the Isolated area (ISO area) are stopped. The clock stop mask registers control clock supply to related clock domains in stand-by mode. Stop all of the peripheral functions before transition to STOP mode if the clock supply to the function will be stopped in STOP mode.

The transition procedure (example) to STOP mode is shown below.

Preparation for stand-by

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupts for non-wake-up factors (ICxxx.MKxxx = 1).
  - Release the masks of the interrupts for wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC0/WUFC1/WUFC_ISO0 registers).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK1/WUFMSK_ISO0 registers).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK1/WUFMSK_ISO0 registers).
- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC_xxx_STPM.xxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).
- RH850/F1KH-D8
  Set a clock according to the method described in **Section 12AB.5.5, CPUCLK Setting in STOP Mode Method**.

Start of stand-by

Set the STBC0STPTRG bit in the STBC0STPT register to 1 to transition to STOP mode.

End of stand-by

When a wake-up factor is generated, the microcontroller returns from STOP mode.

Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0, WUF1, WUF_ISO0).

When an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.

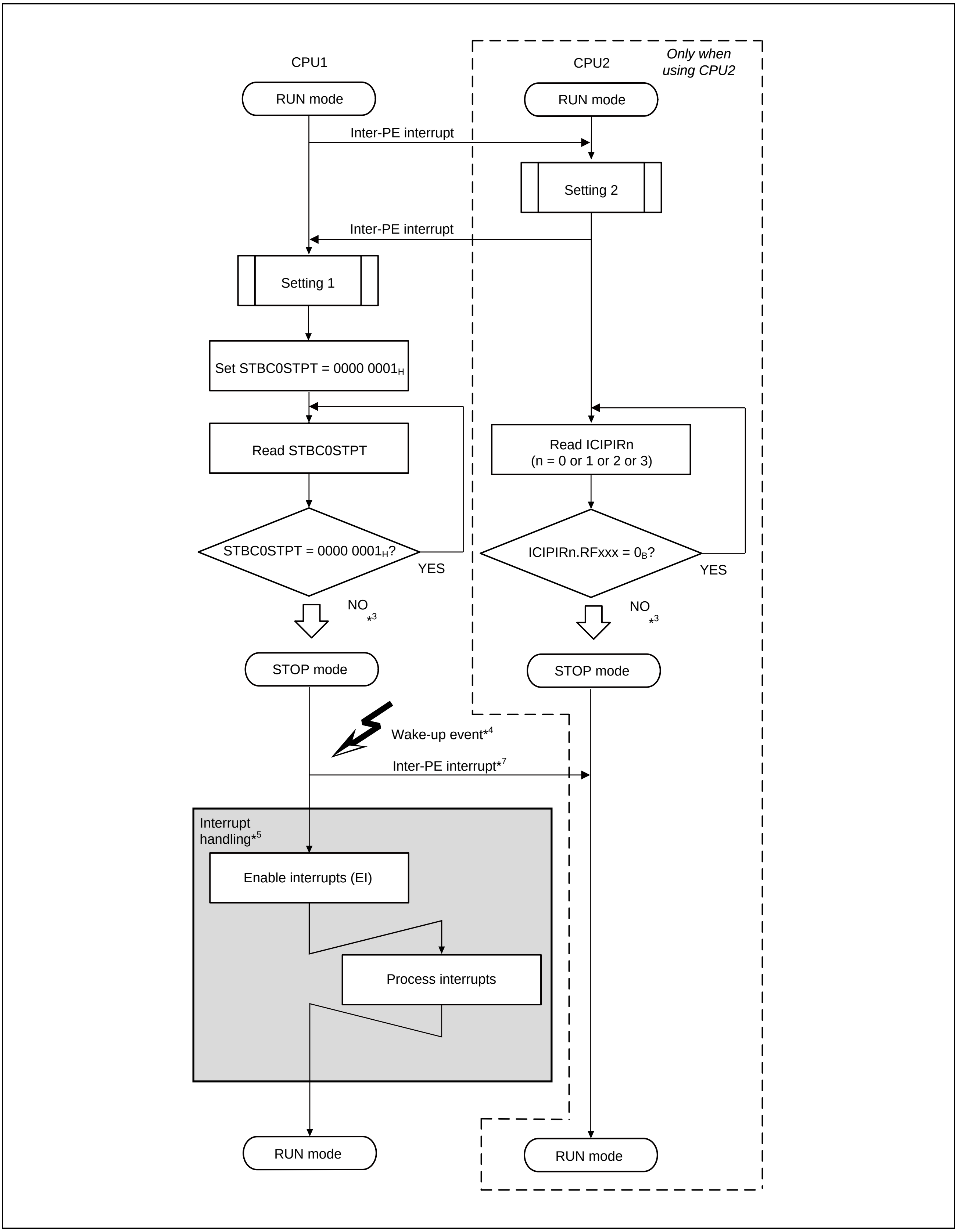
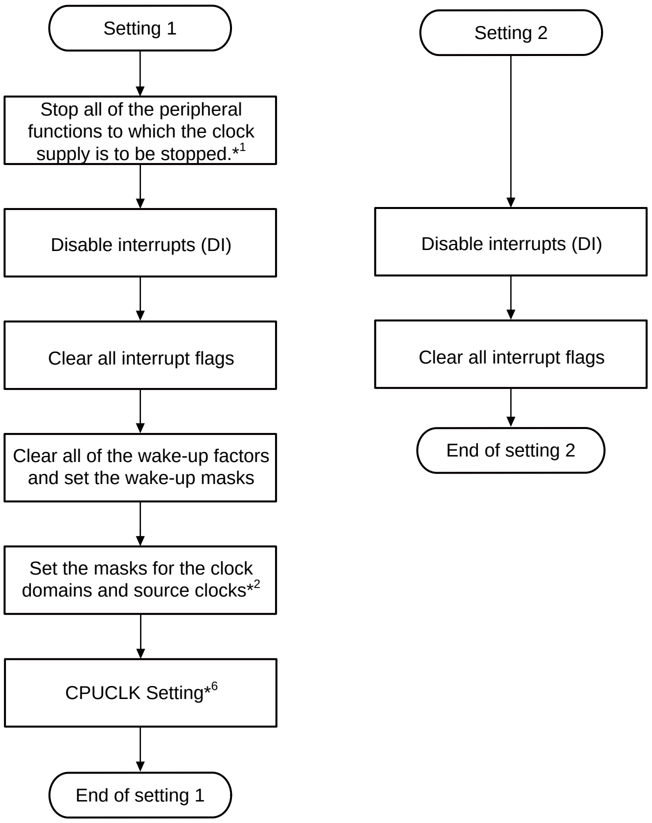


Figure 14.3 Example of STOP Mode Transition (RH850/F1KH-D8) (1/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the STOP mode, the operation of the peripheral function may be incorrect. Before the transition to the STOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. Though the clock mask can be set before this flow is started, it must be set before 0000 0001_H is written to STBC0STPT.

Note 3. The clock supply to the CPU is stopped and the operation transitions to the STOP mode while checking that STBC0STPT = 0000 0001_H and ICIPIRn.RFxxx = 0_B.

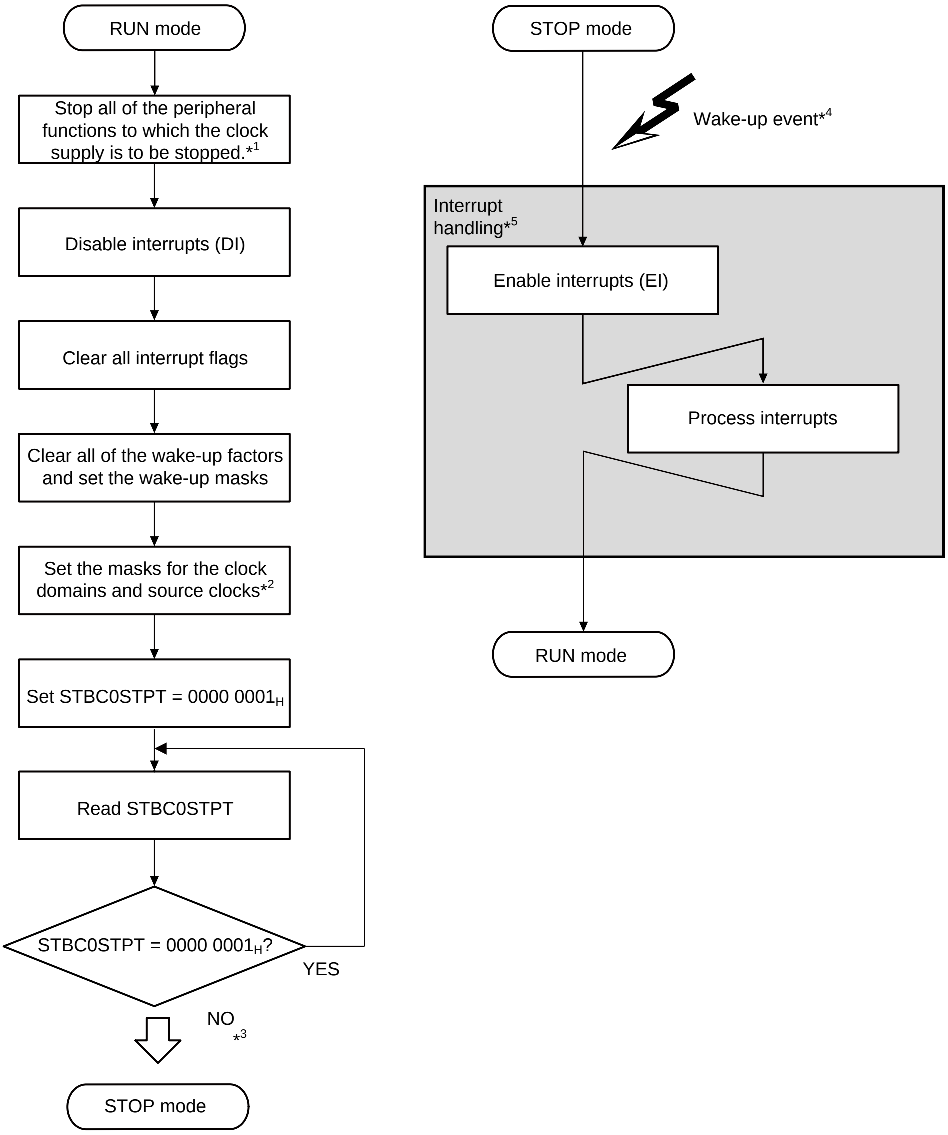
Note 4. STBC0STPT is set to 0000 0000_H at the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF0, WUF1 and WUF_ISO0 registers.

Note 5. This processing is optional. It is required when executing the interrupt handling after the wake-up.

Note 6. For details on the CPUCLK Setting, see **Section, 12AB.5.4 CPUCLK Source and Divided Clock Selection Method**, **Section, 12AB.5.5 CPUCLK Setting in STOP Mode Method**.

Note 7. ICIPIRn.RFxxx can be set to 1_B by the Inter-PE interrupt. ICIPIRn.RFxxx should be cleared by software.

Figure 14.3 Example of STOP Mode Transition (RH850/F1KH-D8) (2/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the STOP mode, the operation of the peripheral function may be incorrect. Before the transition to the STOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. Though the clock mask can be set before this flow is started, it must be set before 0000 0001_H is written to STBC0STPT.

Note 3. The clock supply to the CPU is stopped and the operation transitions to the STOP mode while checking that STBC0STPT = 0000 0001_H.

Note 4. STBC0STPT is set to 0000 0000_H at the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF0, WUF1 and WUF_ISO0 registers.

Note 5. This processing is optional. It is required when executing the interrupt handling after the wake-up.

Figure 14.4 Example of STOP Mode Transition (RH850/F1KM-S4, RH850/F1KM-S1)

14.3.2 DeepSTOP Mode

In DeepSTOP mode, the clock supply to all areas and the power supply to the Isolated area (ISO area) are stopped. However, clock supply to peripheral functions in the Always-On area (AWO area) can be continued by setting the clock stop mask register.

Select the clock other than the PLL as the CPU operating clock, before the transition to DeepSTOP mode.

The transition procedure (example) to DeepSTOP mode is shown below.

Preparation for stand-by

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupts for non-wake-up factors (ICxxx.MKxxx = 1).
  - Release the masks of the interrupts for wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC0/WUFC1/WUFC20 registers).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK1/WUFMSK20 registers).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK1/WUFMSK20 registers).

CAUTION

When a wake-up factor is assigned to both wake-up factor 1 registers and wake-up factor 2 registers, it can be used only in one of them.

- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC_xxx_STPM.xxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).

Start of stand-by

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to transition to DeepSTOP mode.

End of stand-by

When a wake-up factor is generated, the microcontroller returns from DeepSTOP mode.

Wake-up handling

- When returned from DeepSTOP mode due to wake-up factor 1, the microcontroller starts the operation from the reset vector address.

  If one of the following interrupts has been generated before recovery from DeepSTOP mode to RUN mode, the microcontroller restarts operation from the exception handler address:

  - FENMI: FENMI handler address ($E0_H$)
  - FEINT: FEINT handler address ($F0_H$)

  RH850/F1KH-D8, RH850/F1KM-S4:

  Note that the general-purpose registers, local RAM and global RAM are undefined value after return from DeepSTOP mode.

  RH850/F1KM-S1:

  Note that the general-purpose registers and local RAM are undefined value after return from DeepSTOP mode.

- The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0, WUF1).

- The ports in the Isolated area (ISO area) maintain the I/O buffer hold state.
  Release the I/O buffer hold state by executing the following steps:
  1. Re-configure the peripheral functions and port functions.
  2. Set IOHOLD.IOHOLD = 0.

- To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. In addition, release the masking of the interrupt used as wake-up factors. Then, when an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.

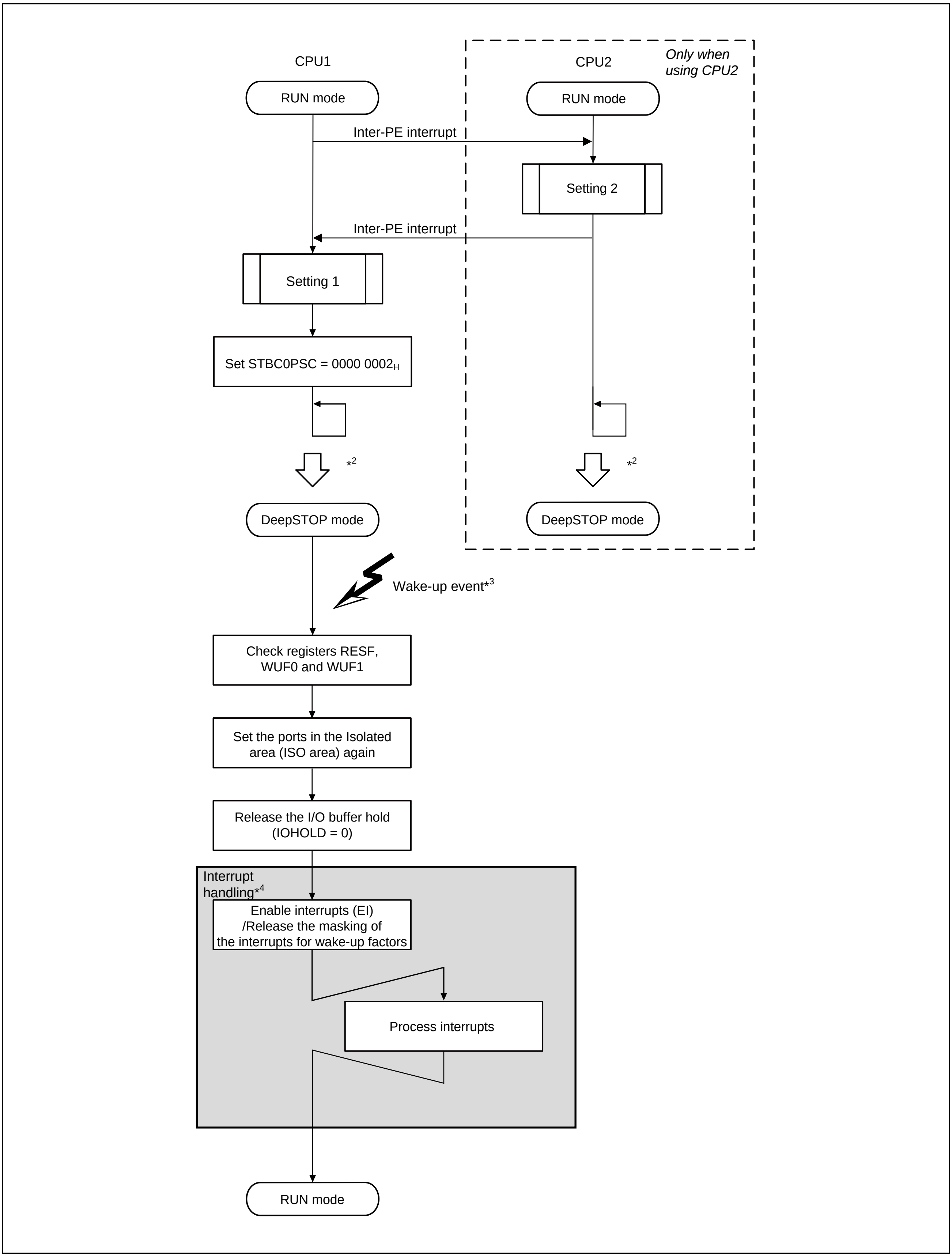
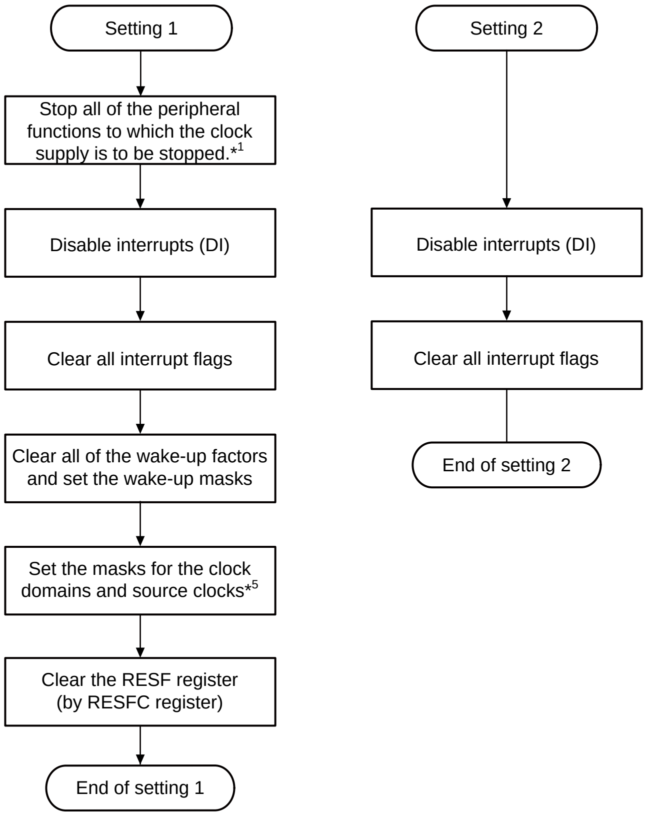


Figure 14.5 Example of DeepSTOP Mode Transition (RH850/F1KH-D8) (1/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DeepSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DeepSTOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

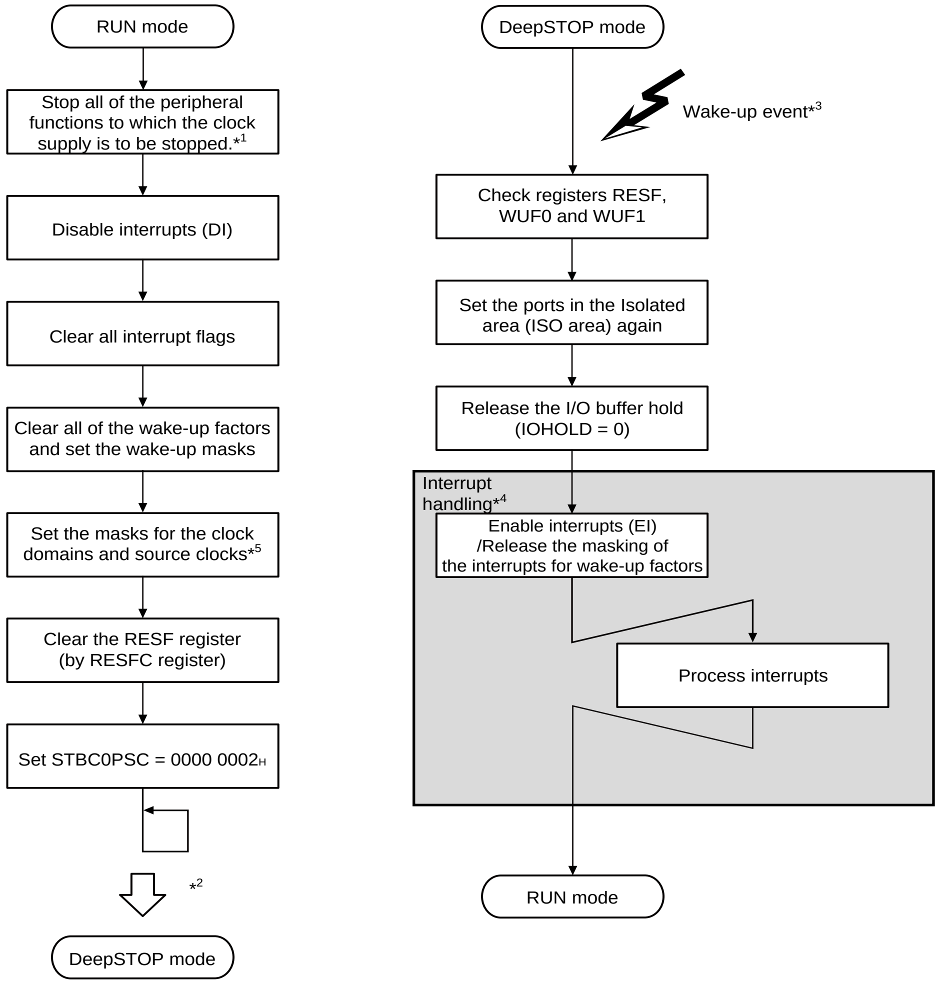
Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DeepSTOP mode by the unconditional loop.

Note 3. The CPU starts the program from the reset vector after the generation of a wake-up factor. The return from the DeepSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up factor can be checked by the WUF0 and WUF1 registers.

Note 4. This processing is optional. It is required when executing the interrupt handling after the wake-up. To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. After that, the interrupt should be enabled.

Note 5. Though the clock mask can be set before this flow is started, it must be set before 0000 0002_H is written to STBC0PSC.

Figure 14.5 Example of DeepSTOP Mode Transition (RH850/F1KH-D8) (2/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DeepSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DeepSTOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DeepSTOP mode by the unconditional loop.

Note 3. The CPU starts the program from the reset vector after the generation of a wake-up factor. The return from the DeepSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up factor can be checked by the WUF0 and WUF1 registers.

Note 4. This processing is optional. It is required when executing the interrupt handling after the wake-up. To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. After that, the interrupt should be enabled.

Note 5. Though the clock mask can be set before this flow is started, it must be set before 0000 0002_H is written to STBC0PSC.

Figure 14.6 Example of DeepSTOP Mode Transition (RH850/F1KM-S4, RH850/F1KM-S1)

14.3.3 Cyclic RUN Mode

In Cyclic RUN mode, the functions except the CPU1, peripheral functions in Always-On area (AWO area), RLIN3, and CSIG0 are stopped. In this mode, PLL and Flash Memory are not available. The CPU2 is also not available. The transition procedure (example) to Cyclic RUN mode is shown below.

Preparation of Cyclic RUN

- RH850/F1KH-D8, RH850/F1KM-S4
  Allocate the program for Cyclic RUN to the retention RAM. The reset vector base address (RBASE) in Cyclic RUN operation is set to the first address of the retention RAM (FEF0 0000_H). Note that neither the code flash memory nor the data flash memory is available in Cyclic RUN mode.
- RH850/F1KM-S1
  Allocate the program for Cyclic RUN to the retention RAM. The reset vector base address (RBASE) in Cyclic RUN operation should be specified in the CYCRBASE register described in **Section 9BC, Reset Controller of RH850/F1KM**. Note that neither the code flash memory nor the data flash memory is available in Cyclic RUN mode.

  The instruction to transition to DeepSTOP mode should be arranged in the interrupt exception handler or a polling routine of interrupt request which is used as the source of returning to the RUN mode.

  For details on the exception vector, see *the RH850G3KH User's Manual: Software*.

CAUTION

Do not change the PSW.EBV bit from its value after reset in Cyclic RUN mode (Do not set the PSW.EBV bit to 1 in Cyclic RUN mode).

- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK20 register).
- Transition to DeepSTOP mode. For details on the transition to DeepSTOP mode, see **Section 14.3.2, DeepSTOP Mode**.

Start of Cyclic RUN

The operation transitions to Cyclic RUN mode from DeepSTOP mode at the generation of wake-up factor 2.

The operation transitions to Cyclic RUN mode from Cyclic STOP mode at the generation of wake-up factors 1 and 2.

- RH850/F1KH-D8, RH850/F1KM-S4
  The microcontroller starts operation from the reset vector address of Cyclic RUN mode (the first address of the retention RAM (FEF0 0000_H)). If one of the following interrupts has been generated during recovery from DeepSTOP mode to Cyclic RUN mode, the microcontroller restart operation from the exception handler address:
  - FENMI: FENMI handler address in Cyclic RUN mode (FEF0 0000_H + $E0_H$)
  - FEINT: FEINT handler address in Cyclic RUN mode (FEF0 0000_H + $F0_H$)

  Note that the general-purpose registers, local RAM and global RAM are undefined value after the transition to Cyclic RUN mode from DeepSTOP mode.

- RH850/F1KM-S1
  The microcontroller starts operation from the reset vector address of Cyclic RUN mode specified by the CYCRBASE register. If one of the following interrupts has been generated during recovery from DeepSTOP mode to Cyclic RUN mode, the microcontroller restart operation from the exception handler address:
  - FENMI: FENMI handler address in Cyclic RUN mode (CYCRBASE + $E0_H$)
  - FEINT: FEINT handler address in Cyclic RUN mode (CYCRBASE + $F0_H$)

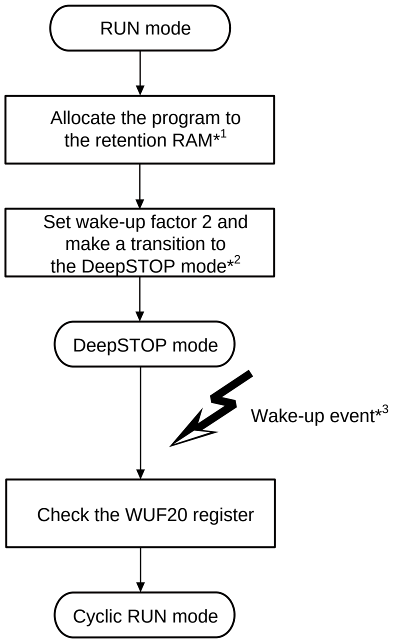
  Note that the general-purpose registers and local RAM are undefined value after the transition to Cyclic RUN mode from DeepSTOP mode.

End of Cyclic RUN

The Cyclic RUN mode ends at the transition to the Cyclic STOP mode by setting the STBC0STPT.STBC0STPTRG bit to 1, or at the transition to the DeepSTOP mode by setting the STBC0PSC.STBC0DISTRG bit to 1.

Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF20).



Note 1. When the mode transitions from the Cyclic RUN mode to the RUN mode by a wake-up factor, the transition to the DeepSTOP mode should be made in the processing of the interrupt vector for the wake-up factor. In that case, allocate the interrupt processing program to the retention RAM as well.

Note 2. Before the transition to the DeepSTOP mode, clear the flag for wake-up factor 2 in the WUFC20 register and set wake-up factor 2 that is to be used in the WUFMSK20 register. Other parts of the procedure for transition to DeepSTOP mode are the same as the normal procedure.

Note 3.

- RH850/F1KH-D8, RH850/F1KM-S4
  The CPU starts the program from the top address (FEF0 0000_H) of the retention RAM after the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF20 register.
- RH850/F1KM-S1
  The CPU starts the program from the address specified by CYCRBASE register after the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF20 register.

Figure 14.7 Example of Cyclic RUN Mode Transition

14.3.4 Cyclic STOP Mode

In Cyclic STOP mode, the functions except the peripheral functions in the Always-On area (AWO area) and RLIN3 are stopped.

The transition procedure (example) to Cyclic STOP mode is shown below.

Preparation for Cyclic STOP

- Transition to Cyclic RUN mode.
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC0/WUFC1/WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK1/WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK1/WUFMSK20 register).

Start of Cyclic STOP

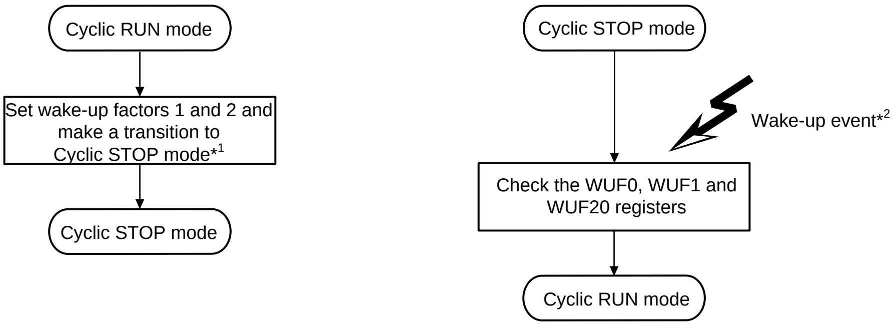
Set the STBC0STPT.STBC0STPTRG bit to 1 to transition to Cyclic STOP mode.

End of Cyclic STOP

The operation transitions to Cyclic RUN mode at the generation of wake-up factor 1 or 2.

Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0, WUF1, WUF20).



Note 1. Set wake-up factors 1 and 2 to be used for transitioning to RUN mode and Cyclic RUN mode, respectively. When the mode transitions to RUN mode by wake-up factor 1, the transition processing to DeepSTOP mode should be added in Cyclic RUN mode.

Note 2. When a wake-up factor is generated in Cyclic STOP mode, the mode transitions to Cyclic RUN mode and the operation starts immediately after the processing shifted to Cyclic STOP mode. The generated wake-up factors can be checked by the WUF0, WUF1 and WUF20 registers.

Figure 14.8 Example of Cyclic STOP Mode Transition

14.4 Writing to the Stand-By Controller Related Registers

The following stand-by controller registers are write-protected registers.

- STBC0PSC register
- STBC0STPT register
- IOHOLD register

The write-protected registers are protected against the illegal writing due to an incorrect program operation.

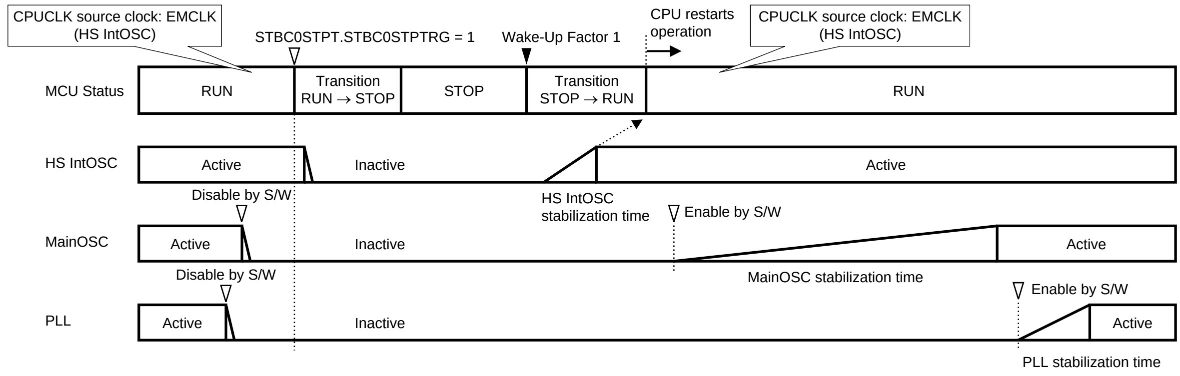
For details on the write-protected sequence, see **Section 5, Write-Protected Registers**.

14.5 Clock Oscillator Behavior During Stand-By Mode Transition

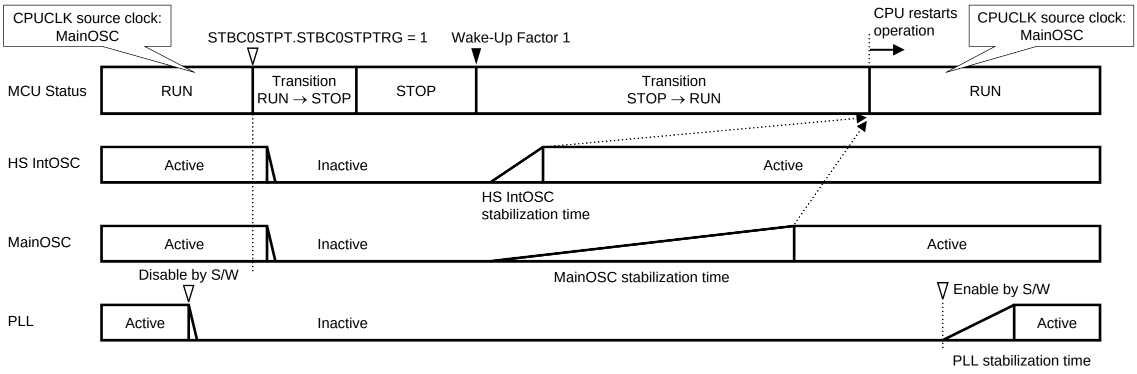
The following figures explain clock oscillator behavior during stand-by mode transition. The clock oscillators restart operation automatically if they are used before entering stand-by.

- If MainOSC and PLL are enabled before entering STOP mode, they restart oscillation automatically during wake-up from STOP mode, and CPU restarts operation after oscillations of these clock sources become stable.
- If MainOSC is enabled before entering DeepSTOP mode, it restarts oscillation automatically during wake-up from DeepSTOP mode, and CPU restarts operation after oscillation of MainOSC becomes stable. PLL is not restarted automatically even if it is enabled before entering DeepSTOP mode.
- If MainOSC is enabled before entering Cyclic STOP mode, it restarts oscillation automatically during wake-up from Cyclic STOP mode. CPU restarts operation after oscillation of MainOSC becomes stable.

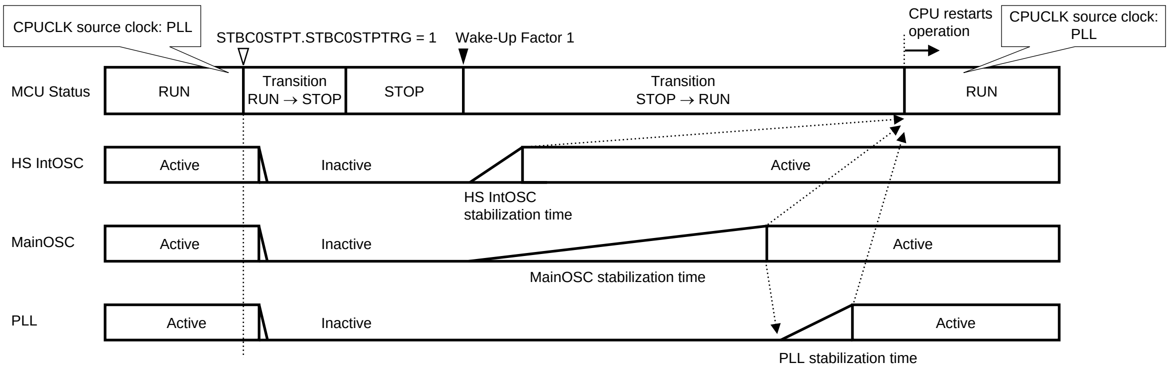
Note that behavior of HS IntOSC and MainOSC in the following figures is in the case they stop oscillation during stand-by mode. HS IntOSC and MainOSC continue oscillation during stand-by mode if their stop mask register is set to 1 (ROSCSTPMSK bit of ROSCSTPM register and MOSCSTPMSK bit of MOSCSTPM register respectively) or there is a clock domain which uses the source clock during stand-by by setting its stop mask bit (CKSC_xxxx_STPM = 0000 0003_H).



(a) PLL and MainOSC are disabled before entering STOP mode.
Restart MainOSC and PLL by software after wake-up.
CPUCLK source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

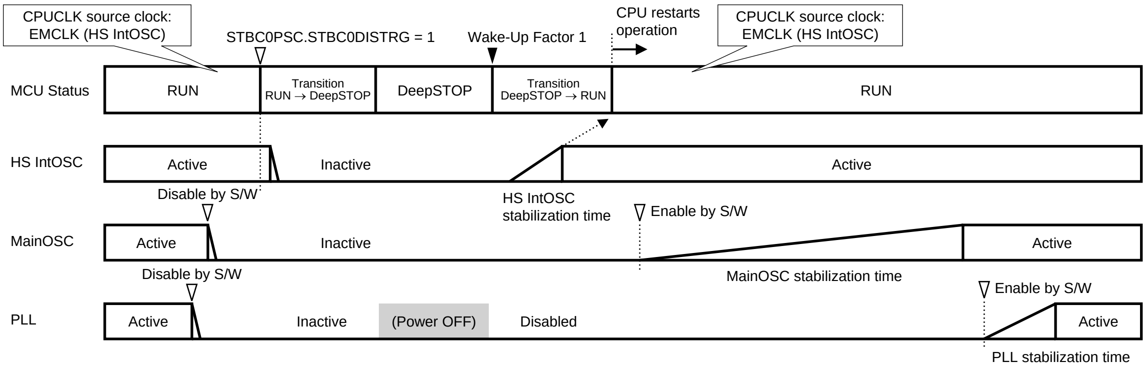


(b) PLL is disabled, and MainOSC is enabled before entering STOP mode.
MainOSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CPUCLK source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

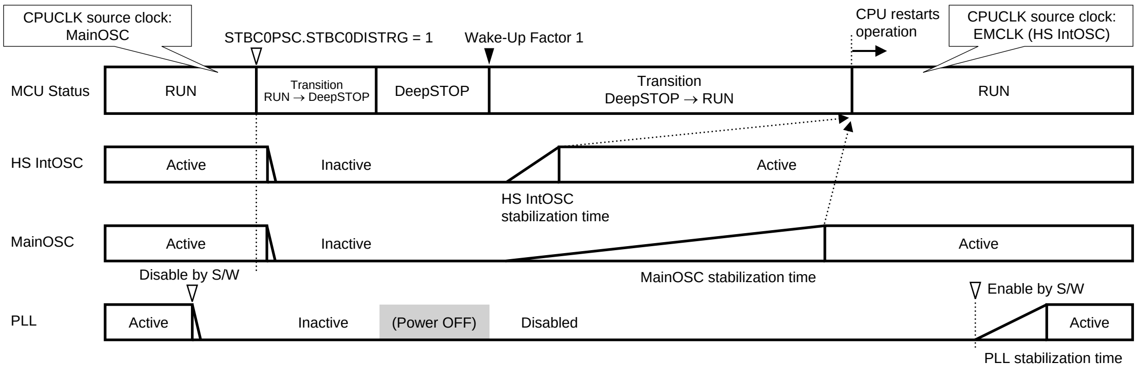


(c) PLL and MainOSC are enabled before entering STOP mode.
PLL and MainOSC are restarted automatically during transition to RUN mode, and become stable CPU restarts operation.
CPUCLK source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

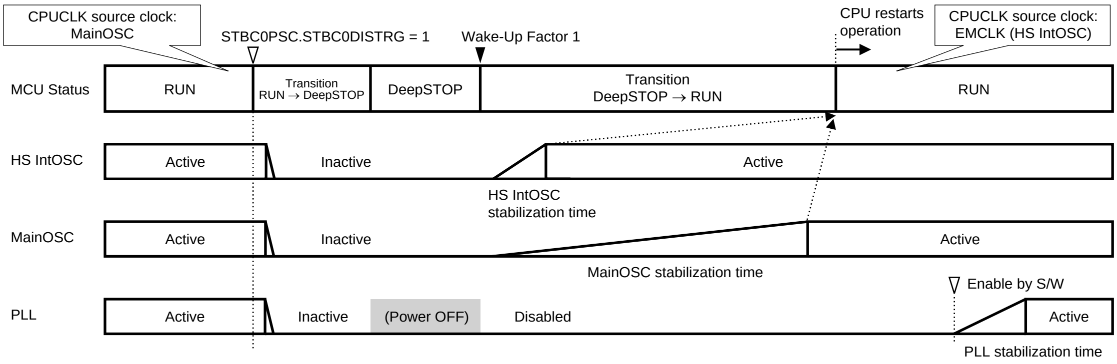
Figure 14.9 Clock Oscillators Behavior in Stand-By Mode Transition
(RUN → STOP → RUN)



(a) PLL and MainOSC are disabled before entering DeepSTOP mode.
Restart MainOSC and PLL by software after wake-up.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

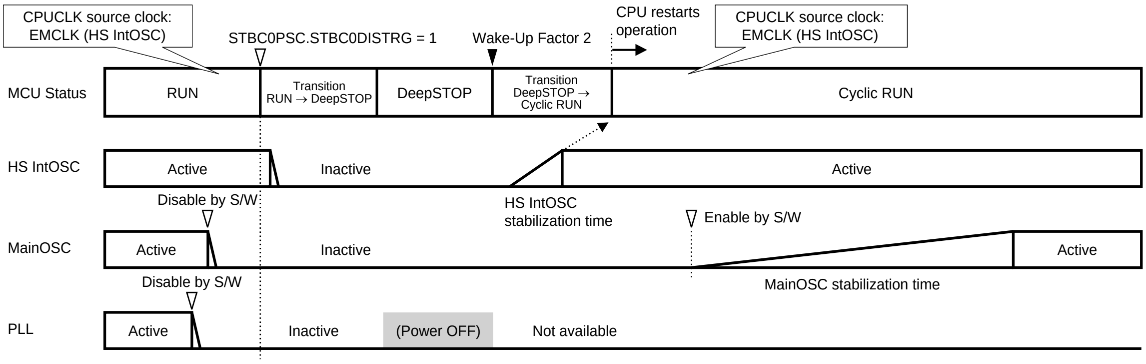


(b) PLL is disabled, and MainOSC is enabled before entering DeepSTOP mode.
MainOSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

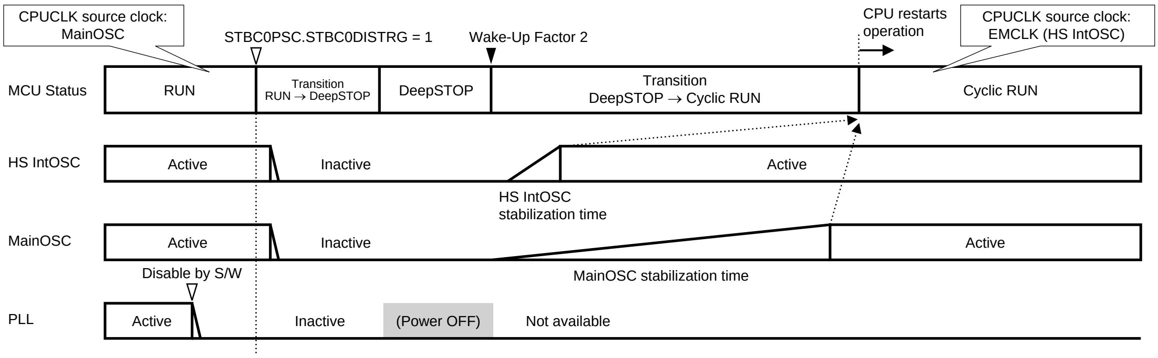


(c) PLL and MainOSC are enabled before entering DeepSTOP mode.
MainOSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

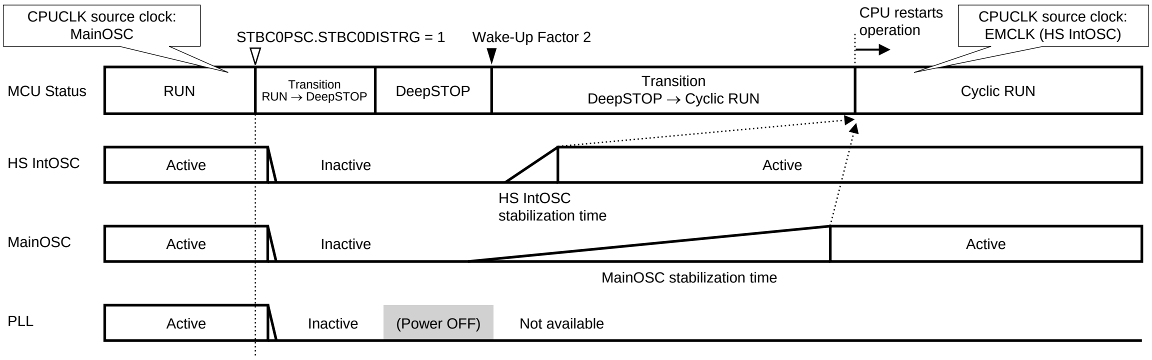
Figure 14.10 Clock Oscillators Behavior in Stand-By Mode Transition (RUN → DeepSTOP → RUN)



(a) PLL and MainOSC are disabled before entering DeepSTOP mode.
Restart MainOSC by software after wake-up. PLL is not available in Cyclic RUN mode.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

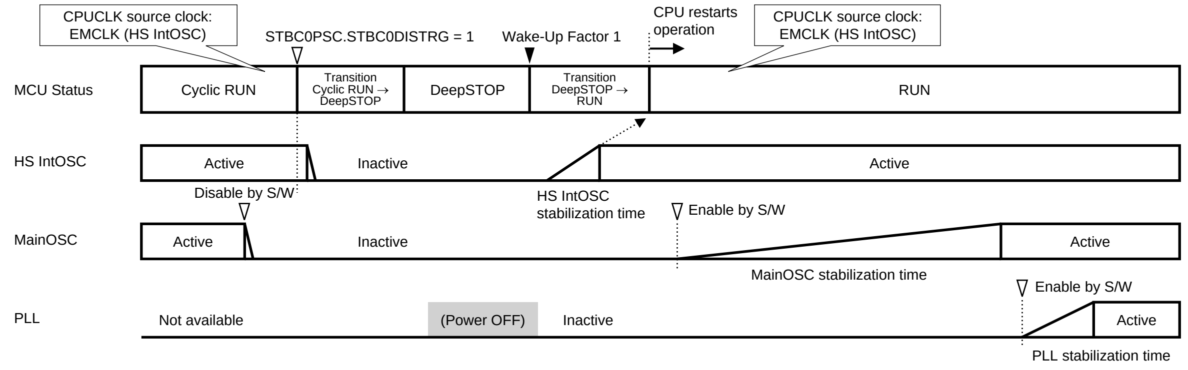


(b) PLL is disabled, and MainOSC is enabled before entering DeepSTOP mode.
MainOSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation. PLL is not available in Cyclic RUN mode.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

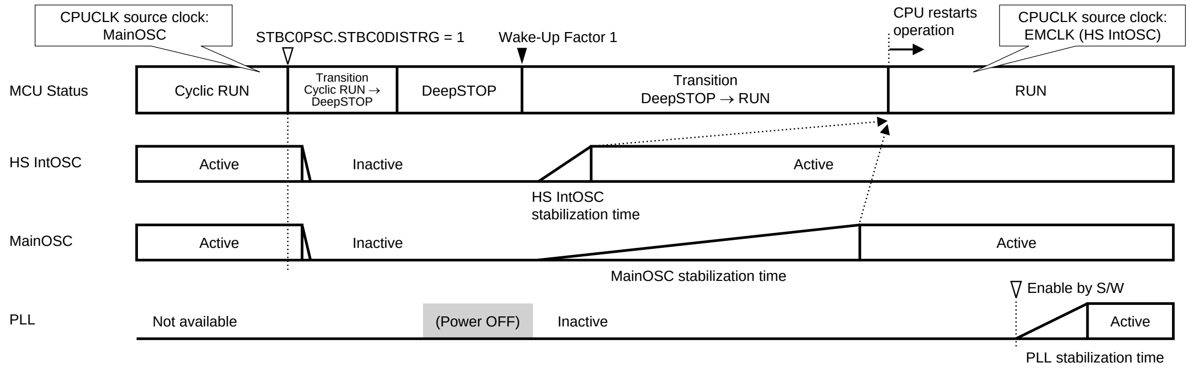


(c) PLL and MainOSC are enabled before entering DeepSTOP mode.
MainOSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation. PLL is not available in Cyclic RUN mode.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

Figure 14.11 Clock Oscillators Behavior in Stand-By Mode Transition (RUN → DeepSTOP → Cyclic RUN)

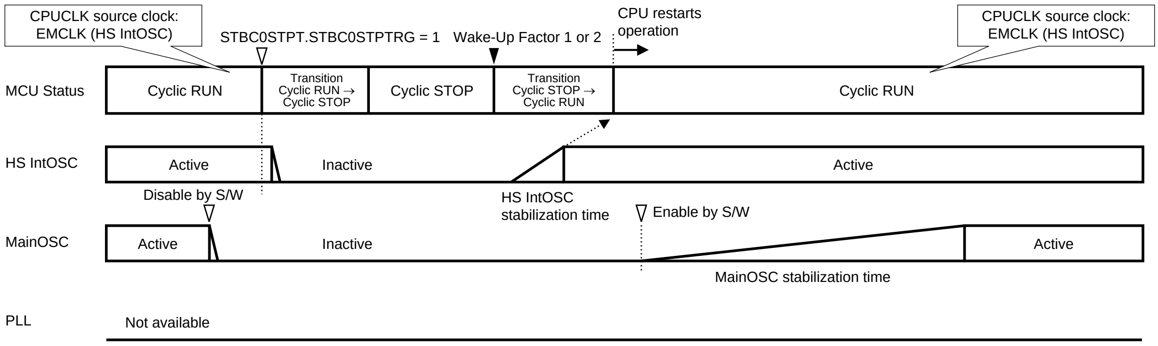


(a) MainOSC is disabled before entering DeepSTOP mode.
Restart MainOSC and PLL by software after wake-up.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

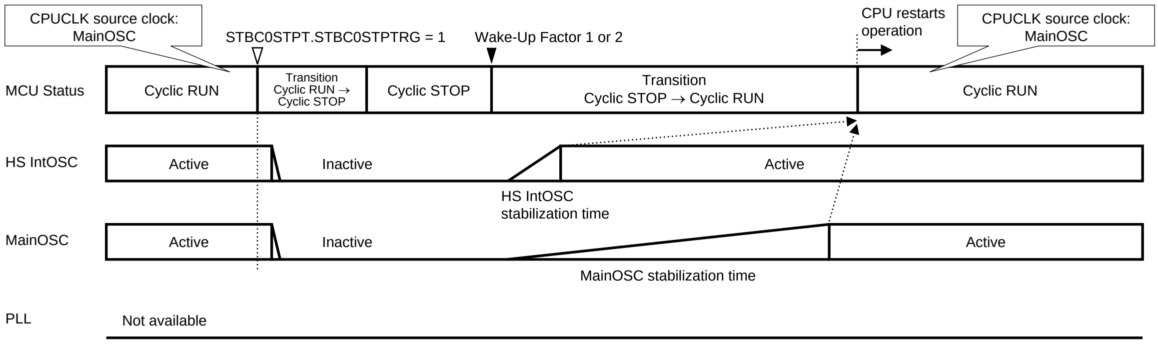


(b) MainOSC is enabled before entering DeepSTOP mode.
MainOSC is restarted automatically during transition to RUN mode, and becomes stable before CPU starts operation.
Restart PLL by software after wake-up.
CPUCLK source clock after wake-up from DeepSTOP mode is EMCLK (HS IntOSC).

Figure 14.12 Clock Oscillators Behavior in Stand-By Mode Transition (Cyclic RUN → DeepSTOP → RUN)



(a) MainOSC is disabled before entering Cyclic STOP mode.
Restart MainOSC by software after wake-up.
CPUCLK source clock after wake-up from Cyclic STOP mode is the same source clock before entering Cyclic STOP mode.



(b) MainOSC is enabled before entering Cyclic STOP mode.
MainOSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation.
CPUCLK source clock after wake-up from Cyclic STOP mode is the same source clock before entering Cyclic STOP mode.

Figure 14.13 Clock Oscillators Behavior in Stand-By Mode Transition
(Cyclic RUN → Cyclic STOP → Cyclic RUN)

14.6 Cautions when Using Stand-By Modes

14.6.1 Cautions Concerning Transitioning to DeepSTOP Mode When Using a Debugger

When using a debugger, executing a program that causes the mode to transition to DeepSTOP mode immediately after the program is started may cause improper communication between the OCD emulator and microcontroller because the microcontroller will enter DeepSTOP mode before the preparations for communication between the OCD emulator and microcontroller are completed.

The communication preparation period depends on the OCD emulator's host PC environment and the operating frequency of the microcontroller, so when performing debugging that causes the program to enter DeepSTOP mode immediately after the program starts, insert a wait between reset release and the DeepSTOP execution instruction so that the debugger starts normally.

In DeepSTOP mode, the debugging controller stops. For return from DeepSTOP mode by the debugger, see **Section 14.1.3, On-Chip Debug Wake-Up**.

Section 15 Low-Power Sampler (LPS)

This section contains a generic description of the low-power sampler (LPS).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of LPS.

15.1 Features of RH850/F1KH, RH850/F1KM LPS

15.1.1 Number of Units

This microcontroller has the following number of LPS units.

Table 15.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)

Table 15.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)

Table 15.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)

Table 15.4 Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name LPSn	Number of Channels per Unit	Function	Channel Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
LPS0	1	Digital port input m for port polling	DPINm	24 ch	24 ch	24 ch
		Analog input m for A/D converter	ADCA0Im	16 ch	16 ch	16 ch

Table 15.5 Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name LPSn	Number of Channels per Unit	Function	Channel Name	RH850/ F1KM-S4 100 Pins	RH850/ F1KM-S4 144 Pins	RH850/ F1KM-S4 176 Pins	RH850/ F1KM-S4 233 Pins	RH850/ F1KM-S4 272 Pins
LPS0	1	Digital port input m for port polling	DPINm	16 ch	24 ch	24 ch	24 ch	24 ch
		Analog input m for A/D converter	ADCA0Im	16 ch	16 ch	16 ch	16 ch	16 ch

Table 15.6 Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name LPSn	Number of Channels per Unit	Function	Channel Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
LPS0	1	Digital port input m for port polling	DPINm	3 ch	8 ch	12 ch	17 ch
		Analog input m for A/D converter	ADCA0Im	8 ch	10 ch	11 ch	16 ch

Table 15.7 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual LPS units are identified by the index “n” (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index “m” (m = 0 to 23) and the number of analog input channels for A/D converter is indicated by the index “m” (m = 0 to 15)
k	The external multiplexer select output signal is indicated by the index “k”.
x	LPS sequence start trigger input signal is indicated by the index “x”.
y	Throughout this section, the individual TAUJ units are identified by the index “y”.

Table 15.8 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual LPS units are identified by the index “n” (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index “m” (m = 0 to 23) and the number of analog input channels for A/D converter is indicated by the index “m” (m = 0 to 15)
k	The external multiplexer select output signal is indicated by the index “k”.
x	LPS sequence start trigger input signal is indicated by the index “x”.
y	Throughout this section, the individual TAUJ units are identified by the index “y”.

Table 15.9 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual LPS units are identified by the index “n” (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index “m” (m = 0 to 16) and the number of analog input channels for A/D converter is indicated by the index “m” (m = 0 to 15)
k	The external multiplexer select output signal for digital port is indicated by the index “k”.
x	LPS sequence start trigger input signal is indicated by the index “x”.
y	Throughout this section, the individual TAUJ units are identified by the index “y”.

NOTE

Descriptions of functions and registers in this section are based on the maximum configurations. Adjust the indices in the text to the proper value for each product. When writing a value to a register that will result in writing to bits outside the range of the index for the product you are using, write the value after reset to these bits.

The following table shows values indicated by the indices of each product.

Table 15.10 Indices of Products (RH850/F1KH-D8)

Indices of Each Product		
176 Pins	233 Pins	324 Pins
m = 0 to 23*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]
k = 0 to 2	k = 0 to 2	k = 0 to 2
x = 0 to 3	x = 0 to 3	x = 0 to 3
y = 0, 2	y = 0, 2	y = 0, 2

Note 1. Digital port input m for port polling

Note 2. Analog input m for A/D converter

Table 15.11 Indices of Products (RH850/F1KM-S4)

Indices of Each Product				
100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
m = 1 to 16*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]	m = 0 to 23*[1] m = 0 to 15*[2]
k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2
x = 0 to 3	x = 0 to 3	x = 0 to 3	x = 0 to 3	x = 0 to 3
y = 0, 2	y = 0, 2	y = 0, 2	y = 0, 2	y = 0, 2

Note 1. Digital port input m for port polling

Note 2. Analog input m for A/D converter

Table 15.12 Indices of Products (RH850/F1KM-S1)

Indices of Each Product			
48 Pins	64 Pins	80 Pins	100 Pins
m = 0 to 2*[1] m = 0 to 7*[2]	m = 0 to 4, 8 to 10*[1] m = 0 to 9*[2]	m = 0 to 11*[1] m = 0 to 10*[2]	m = 0 to 16*[1] m = 0 to 15*[2]
—*[3]	k = 0 to 2	k = 0 to 2	k = 0 to 2
x = 0 to 3	x = 0 to 3	x = 0 to 3	x = 0 to 3
y = 0, 2	y = 0, 2	y = 0, 2	y = 0, 2

Note 1. Digital port input m for port polling

Note 2. Analog input m for A/D converter

Note 3. ADCA0SELk is k = 0 to 2.
For details, see **Section 38.1.6, External Input/Output Signals**.

15.1.2 Register Base Address

The LPS base address is listed in the following table.

LPS register addresses are given as an offset from the base address.

Table 15.13 Register Base Address (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<LPS0_base>	FFF8 3000_H

15.1.3 Clock Supply

The LPS clock supply is shown in the following table.

If the operation request signal for the low-power sampler (LPS) is at the active level, the clock for C_AWO_ADCA for which the HS IntOSC is selected also operates.

Table 15.14 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name
LPSn	Register access clock	CPUCLK_L, EMCLK
	Operating clock	EMCLK

15.1.4 Interrupt Requests

The LPS interrupt requests are listed in the following table.

Table 15.15 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
LPS0			
INTCWEND	Port polling end interrupt (LPS)	112	—
INTDPE	Digital port error interrupt (LPS)	356	—
INTAPE	Analog port error interrupt (LPS)	357	—
INTADCA0I0*[1]	ADCA0 SG1 end interrupt	18	4
INTADCA0I1*[1]	ADCA0 SG2 end interrupt	19	5
INTADCA0I2*[1]	ADCA0 SG3 end interrupt	20, 32	6

Note 1. These signals are output from ADCA0.

15.1.5 Reset Sources

The LPS reset sources are shown in the following table. The LPS is initialized by these reset sources.

Table 15.16 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
LPSn	All reset sources except transition to DeepSTOP mode (AWORES)

15.1.6 External Input/Output Signals

External input/output signals of LPS are listed below.

Table 15.17 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
LPS0		
DPO	Port output signal for digital input	DPO
DPSELk	External multiplexer select output signal for digital port	SELDPk
DPINm	Digital port input signal	DPINm
APO	Port output signal for analog input	APO
ADCA0SELk*[1]	External analog multiplexer (MPX) output pin	ADCA0SELk
ADCA0Im*[1]	ADCA input channel signal	ADCA0Im

Note 1. These signals are input/output of ADCA0. For details, see **Section 38.1.6, External Input/Output Signals**.

15.1.7 Internal Input/Output Signals

Internal input/output signals for connecting the LPS and the STBC or the LPS and the TAUJ are listed below.

Table 15.18 Internal Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
WUTRG0	LPS wake-up source trigger 0 output signal	STBC
WUTRG1	LPS wake-up source trigger 1 output signal	STBC
INTTAUJyIx	LPS sequence start trigger x input signal	TAUJy

15.2 Overview

15.2.1 Functional Overview

To monitor the external input without consuming CPU resources, the low-power sampler (LPS) can check the digital input ports and analog input ports without using the CPU. The figure below shows a connection example between the main components of the LPS and the external circuit.

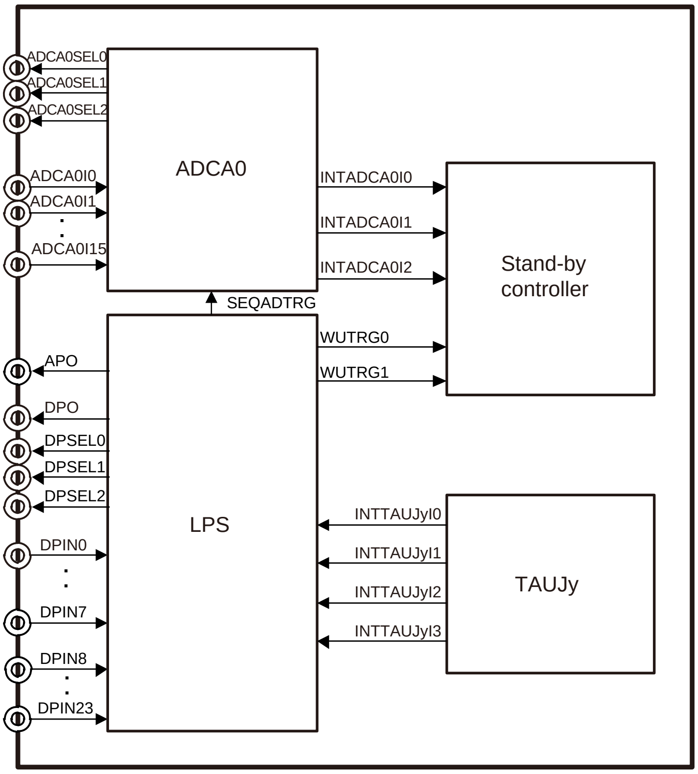


Figure 15.1 Block Diagram of the LPS

CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative function.
They cannot be used simultaneously.

15.3 Registers

15.3.1 List of Registers

LPS registers are listed in the following table.

For details about <LPS0_base>, see **Section 15.1.2, Register Base Address**.

Table 15.19 List of Registers

Module Name	Register Name	Symbol	Address
LPS0	LPS control register	SCTLR	<LPS0_base > + 00_H
	Event flag register	EVFR	<LPS0_base > + 04_H
	DPIN select register 0	DPSELR0	<LPS0_base > + 08_H
	DPIN select register M	DPSELRM	<LPS0_base > + $0C_H$
	DPIN select register H	DPSELRH	<LPS0_base > + 10_H
	DPIN data set register 0	DPDSR0	<LPS0_base > + 14_H
	DPIN data set register M	DPDSRM	<LPS0_base > + 18_H
	DPIN data set register H	DPDSRH	<LPS0_base > + $1C_H$
	DPIN data input monitor register 0	DPDIMR0	<LPS0_base > + 20_H
	DPIN data input monitor register 1	DPDIMR1	<LPS0_base > + 24_H
	DPIN data input monitor register 2	DPDIMR2	<LPS0_base > + 28_H
	DPIN data input monitor register 3	DPDIMR3	<LPS0_base > + $2C_H$
	DPIN data input monitor register 4	DPDIMR4	<LPS0_base > + 30_H
	DPIN data input monitor register 5	DPDIMR5	<LPS0_base > + 34_H
	DPIN data input monitor register 6	DPDIMR6	<LPS0_base > + 38_H
	DPIN data input monitor register 7	DPDIMR7	<LPS0_base > + $3C_H$
	Count value register	CNTVAL	<LPS0_base > + 40_H
	LPS operation status register	SOSTR	<LPS0_base > + 44_H

15.3.2 SCTLR — LPS Control Register

This register is used to configure the LPS.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TJIS2	NUMDP2	NUMDP1	NUMDP0	TJIS1	TJIS0	ADEN	DPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.20 SCTLR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 8</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>6 to 4</td><td>NUMDP[2:0]</td><td>These bits specify the number of times the port is read in digital input mode. If two or more times are specified, the external multiplexer is controlled by the DPSEL[2:0] pins.
The bits for which comparison is enabled in the DPSELR0, DPSELRM, and DPSELRH registers are compared regardless of the repeat number setting, and WUTRG will be generated according to the results.
<table>
<tr><th>NUMDP[2:0]</th><th>Number of Times the Port Is Read</th></tr>
<tr><td>000_B</td><td>One time</td></tr>
<tr><td>001_B</td><td>Two times</td></tr>
<tr><td>010_B</td><td>Three times</td></tr>
<tr><td>011_B</td><td>Four times</td></tr>
<tr><td>100_B</td><td>Five times</td></tr>
<tr><td>101_B</td><td>Six times</td></tr>
<tr><td>110_B</td><td>Seven times</td></tr>
<tr><td>111_B</td><td>Eight times</td></tr>
</table>
These bits should be set before the TAUJy and sequence operations are started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0).
(When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)</td></tr>
<tr><td>7, 3, 2</td><td>TJIS[2:0]</td><td>Sequence Start Trigger Select
000: INTTAUJ0I0 100: INTTAUJ2I0
001: INTTAUJ0I1 101: INTTAUJ2I1
010: INTTAUJ0I2 110: INTTAUJ2I2
011: INTTAUJ0I3 111: INTTAUJ2I3
These bits should be set before the sequence operation is started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0).
(When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)</td></tr>
<tr><td>1</td><td>ADEN</td><td>0: Analog input mode is disabled
1: Analog input mode is enabled</td></tr>
<tr><td>0</td><td>DPEN</td><td>0: Digital input mode is disabled
1: Digital input mode is enabled</td></tr>
</table>

15.3.3 EVFR — Event Flag Register

This register indicates the result of comparing the data sequentially captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 04$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DINEVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.21 EVFR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DINEVF	This bit indicates the result of comparing the data captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers. Read: 0: The result of comparison is a match. 1: The result of comparison is a mismatch. Write: 0: Clear the bit. 1: Prohibited. This bit is set to 1 when a mismatch is detected even in one bit. Only 0 can be written to clear this bit.

15.3.4 DPSELR0 — DPIN Select Register 0

This register specifies the compare target bits in the DPDSR0 and DPDIMR0 registers.

Write to the DPSELR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0EN_23	D0EN_22	D0EN_21	D0EN_20	D0EN_19	D0EN_18	D0EN_17	D0EN_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0EN_15	D0EN_14	D0EN_13	D0EN_12	D0EN_11	D0EN_10	D0EN_9	D0EN_8	D0EN_7	D0EN_6	D0EN_5	D0EN_4	D0EN_3	D0EN_2	D0EN_1	D0EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.22 DPSELR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0EN_n (n = 23 to 0)	These bits enable or disable comparing each bit of the first data captured at the digital input pins and stored in the DPDIMR0 register with the comparison target data in the DPDSR0 register. 0: Disables comparison. 1: Enables comparison.

15.3.5 DPSELRM — DPIN Select Register M

This register specifies the compare target bits in the DPDSRM and DPDIMRm (m = 4 to 1) registers.

Write to the DPSELRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access:	DPSELRM can be read or written in 32-bit units. DPSELRML and DPSELRMH can be read or written in 16-bit units. DPSELR1, DPSELR2, DPSELR3, and DPSELR4 can be read or written in 8-bit units.
Address:	DPSELRM: <LPS0_base> + 0C$_H$ DPSELRML: <LPS0_base> + 0C$_H$, DPSELRMH: <LPS0_base> + 0E$_H$ DPSELR1: <LPS0_base> + 0C$_H$, DPSELR2: <LPS0_base> + 0D$_H$, DPSELR3: <LPS0_base> + 0E$_H$, DPSELR4: <LPS0_base> + 0F$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4EN_7	D4EN_6	D4EN_5	D4EN_4	D4EN_3	D4EN_2	D4EN_1	D4EN_0	D3EN_7	D3EN_6	D3EN_5	D3EN_4	D3EN_3	D3EN_2	D3EN_1	D3EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2EN_7	D2EN_6	D2EN_5	D2EN_4	D2EN_3	D2EN_2	D2EN_1	D2EN_0	D1EN_7	D1EN_6	D1EN_5	D1EN_4	D1EN_3	D1EN_2	D1EN_1	D1EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.23 DPSELRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fifth data captured at the digital input pins and stored in the DPDIMR4 register with the comparison target data in the DPDSR4 register. 0: Disables comparison. 1: Enables comparison.
23 to 16	D3EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fourth data captured at the digital input pins and stored in the DPDIMR3 register with the comparison target data in the DPDSR3 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D2EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the third data captured at the digital input pins and stored in the DPDIMR2 register with the comparison target data in the DPDSR2 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D1EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the second data captured at the digital input pins and stored in the DPDIMR1 register with the comparison target data in the DPDSR1 register. 0: Disables comparison. 1: Enables comparison.

15.3.6 DPSELRH — DPIN Select Register H

This register specifies the compare target bits in the DPDSRH and DPDIMRm (m = 7 to 5) registers.

Write to the DPSELRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access:	DPSELRH can be read or written in 32-bit units. DPSELRHL and DPSELRHH can be read or written in 16-bit units. DPSELR5, DPSELR6, and DPSELR7 can be read or written in 8-bit units.
Address:	DPSELRH: <LPS0_base> + 10_H DPSELRHL: <LPS0_base> + 10_H, DPSELRHH: <LPS0_base> + 12_H DPSELR5: <LPS0_base> + 10_H, DPSELR6: <LPS0_base> + 11_H, DPSELR7: <LPS0_base> + 12_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7EN_7	D7EN_6	D7EN_5	D7EN_4	D7EN_3	D7EN_2	D7EN_1	D7EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6EN_7	D6EN_6	D6EN_5	D6EN_4	D6EN_3	D6EN_2	D6EN_1	D6EN_0	D5EN_7	D5EN_6	D5EN_5	D5EN_4	D5EN_3	D5EN_2	D5EN_1	D5EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.24 DPSELRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the eighth data captured at the digital input pins and stored in the DPDIMR7 register with the compare target data in the DPDSR7 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D6EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the seventh data captured at the digital input pins and stored in the DPDIMR6 register with the compare target data in the DPDSR6 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D5EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the sixth data captured at the digital input pins and stored in the DPDIMR5 register with the compare target data in the DPDSR5 register. 0: Disables comparison. 1: Enables comparison.

15.3.7 DPDSR0 — DPIN Data Set Register 0

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR0 register.

Write to the DPDSR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0_23	D0_22	D0_21	D0_20	D0_19	D0_18	D0_17	D0_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0_15	D0_14	D0_13	D0_12	D0_11	D0_10	D0_9	D0_8	D0_7	D0_6	D0_5	D0_4	D0_3	D0_2	D0_1	D0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.25 DPDSR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0_n (n = 23 to 0)	Data to be compared with the first digital port input (DPINm)

15.3.8 DPDSRM — DPIN Data Set Register M

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR4 to DPDIMR1 registers.

Write to the DPDSRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access:	DPDSRM can be read or written in 32-bit units. DPDSRML and DPDSRMH can be read or written in 16-bit units. DPDSR1, DPDSR2, DPDSR3, and DPDSR4 can be read or written in 8-bit units.
Address:	DPDSRM: <LPS0_base> + 18_H DPDSRML: <LPS0_base> + 18_H, DPDSRMH: <LPS0_base> + $1A_H$ DPDSR1: <LPS0_base> + 18_H, DPDSR2: <LPS0_base> + 19_H, DPDSR3: <LPS0_base> + $1A_H$, DPDSR4: <LPS0_base> + $1B_H$
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4_7	D4_6	D4_5	D4_4	D4_3	D4_2	D4_1	D4_0	D3_7	D3_6	D3_5	D3_4	D3_3	D3_2	D3_1	D3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2_7	D2_6	D2_5	D2_4	D2_3	D2_2	D2_1	D2_0	D1_7	D1_6	D1_5	D1_4	D1_3	D1_2	D1_1	D1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.26 DPDSRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4_n (n = 7 to 0)	Data to be compared with the fifth digital port input (DPINm)
23 to 16	D3_n (n = 7 to 0)	Data to be compared with the fourth digital port input (DPINm)
15 to 8	D2_n (n = 7 to 0)	Data to be compared with the third digital port input (DPINm)
7 to 0	D1_n (n = 7 to 0)	Data to be compared with the second digital port input (DPINm)

15.3.9 DPDSRH — DPIN Data Set Register H

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR7 to DPDIMR5 registers.

Write to the DPDSRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRH can be read or written in 32-bit units.
DPDSRHL and DPDSRHH can be read or written in 16-bit units.
DPDSR5, DPDSR6, and DPDSR7 can be read or written in 8-bit units.

Address: DPDSRH: <LPS0_base> + $1C_H$
DPDSRHL: <LPS0_base> + $1C_H$,
DPDSRHH: <LPS0_base> + $1E_H$
DPDSR5: <LPS0_base> + $1C_H$,
DPDSR6: <LPS0_base> + $1D_H$,
DPDSR7: <LPS0_base> + $1E_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7_7	D7_6	D7_5	D7_4	D7_3	D7_2	D7_1	D7_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6_7	D6_6	D6_5	D6_4	D6_3	D6_2	D6_1	D6_0	D5_7	D5_6	D5_5	D5_4	D5_3	D5_2	D5_1	D5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.27 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7_n (n = 7 to 0)	Data to be compared with the eighth digital port input (DPINm)
15 to 8	D6_n (n = 7 to 0)	Data to be compared with the seventh digital port input (DPINm)
7 to 0	D5_n (n = 7 to 0)	Data to be compared with the sixth digital port input (DPINm)

15.3.10 DPDIMR0 — DPIN Data Input Monitor Register 0

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 23)) in digital input mode. DPDIMR0 stores the data acquired for the first time.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <LPS0_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0M_23	D0M_22	D0M_21	D0M_20	D0M_19	D0M_18	D0M_17	D0M_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0M_15	D0M_14	D0M_13	D0M_12	D0M_11	D0M_10	D0M_9	D0M_8	D0M_7	D0M_6	D0M_5	D0M_4	D0M_3	D0M_2	D0M_1	D0M_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.28 DPDIMR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	D0M_n (n = 23 to 0)	The first digital port input (DPINm) data

15.3.11 DPDIMR1 — DPIN Data Input Monitor Register 1

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR1 stores the data acquired for the second time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D1M_7	D1M_6	D1M_5	D1M_4	D1M_3	D1M_2	D1M_1	D1M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.29 DPDIMR1 Register Contents

Bit Position	Bit Name	Function
7 to 0	D1M_n (n = 7 to 0)	The second digital port input (DPINm) data

15.3.12 DPDIMR2 — DPIN Data Input Monitor Register 2

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR2 stores the data acquired for the third time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 28_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D2M_7	D2M_6	D2M_5	D2M_4	D2M_3	D2M_2	D2M_1	D2M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.30 DPDIMR2 Register Contents

Bit Position	Bit Name	Function
7 to 0	D2M_n (n = 7 to 0)	The third digital port input (DPINm) data

15.3.13 DPDIMR3 — DPIN Data Input Monitor Register 3

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR3 stores the data acquired for the fourth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D3M_7	D3M_6	D3M_5	D3M_4	D3M_3	D3M_2	D3M_1	D3M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.31 DPDIMR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	D3M_n (n = 7 to 0)	The fourth digital port input (DPINm) data

15.3.14 DPDIMR4 — DPIN Data Input Monitor Register 4

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR4 stores the data acquired for the fifth time.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	<LPS0_base> + 30_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	D4M_7	D4M_6	D4M_5	D4M_4	D4M_3	D4M_2	D4M_1	D4M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.32 DPDIMR4 Register Contents

Bit Position	Bit Name	Function
7 to 0	D4M_n (n = 7 to 0)	The fifth digital port input (DPINm) data

15.3.15 DPDIMR5 — DPIN Data Input Monitor Register 5

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR5 stores the data acquired for the sixth time.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	<LPS0_base> + 34_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	D5M_7	D5M_6	D5M_5	D5M_4	D5M_3	D5M_2	D5M_1	D5M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.33 DPDIMR5 Register Contents

Bit Position	Bit Name	Function
7 to 0	D5M_n (n = 7 to 0)	The sixth digital port input (DPINm) data

15.3.16 DPDIMR6 — DPIN Data Input Monitor Register 6

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR6 stores the data acquired for the seventh time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D6M_7	D6M_6	D6M_5	D6M_4	D6M_3	D6M_2	D6M_1	D6M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.34 DPDIMR6 Register Contents

Bit Position	Bit Name	Function
7 to 0	D6M_n (n = 7 to 0)	The seventh digital port input (DPINm) data

15.3.17 DPDIMR7 — DPIN Data Input Monitor Register 7

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode. DPDIMR7 stores the data acquired for the eighth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $3C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D7M_7	D7M_6	D7M_5	D7M_4	D7M_3	D7M_2	D7M_1	D7M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.35 DPDIMR7 Register Contents

Bit Position	Bit Name	Function
7 to 0	D7M_n (n = 7 to 0)	The eighth digital port input (DPINm) data

15.3.18 CNTVAL — Count Value Register

This register specifies the stabilization time of the external circuits (digital signal source and analog signal source).

- In digital mode
  The time from when the DPO output is set to 1 to the time when the port input is acquired for the first time
- In analog mode
  The time from when the APO output is set to 1 to the time when the LPS outputs the A/D conversion trigger to the ADCA0

Write to the CNTVAL register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 16-bit units.

Address: <LPS0_base> + 40_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT17	CNT16	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT07	CNT06	CNT05	CNT04	CNT03	CNT02	CNT01	CNT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.36 CNTVAL Register Contents

Bit Position	Bit Name	Function
15 to 8	CNT1n (n = 7 to 0)	These bits set the stabilization time of the external circuit (analog signal source). Stabilization time = $(1/f_{RH}) \times 16 \times$ CNT1n (set value)
7 to 0	CNT0n (n = 7 to 0)	These bits set the stabilization time of the external circuit (digital signal source). Stabilization time = $(1/f_{RH}) \times 16 \times$ CNT0n (set value)

15.3.19 SOSTR — LPS Operation Status Register

This register indicates the operating state of the LPS.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 44_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SOF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.37 SOSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	SOF	LPS Operation Status Flag 0: Initial state before the occurrence of the start trigger 1: LPS operation is in progress (after the start trigger occurs) If the start trigger occurs while the SOF bit is set to 1 (during the LPS operation), the start trigger is canceled.

NOTE

When DPEN or ADEN is set to 0 for stop of the LPS, SOF bit is set to 0 after the bit becomes 1 once by hardware.

15.4 Digital Input Mode

With the digital input port DPINm and the externally connected multiplexer, up to 64 input ports can be monitored as shown in **Table 15.38, Combination of Monitored Ports**.

Port DPSELk is used to switch the external multiplexer. The DPSELk output is switched for the number of times specified in the SCTLR register.

TAUJy is used to set the timing to check the value input to the port.

Table 15.38 Combination of Monitored Ports

Combination (Number of Ports × Number of Checks)	Ports Used	Total Number
Direct mode When input ports are checked simultaneously without using the external multiplexer Up to 24 ports × 1	DPIN23 to DPIN0	Up to 24
Multiplexer mode When input ports are checked by using a small number of pins and the external multiplexer Up to 8 ports × 8	DPIN7 to DPIN0 DPSEL2 to DPSEL0	Up to 64
MIX mode When input ports are checked using a combination of the above two modes Up to 14 ports × 1 + Up to 7 ports × 7	DPIN7 to DPIN0 DPIN16 to DPIN11 DPSEL2 to DPSEL0*[1]	Up to 63

Note 1. DPIN16 to DPIN11 and DPIN7 are checked only for the first time. DPIN10 to DPIN8 cannot be used because they are shared with DPSEL2 to DPSEL0.

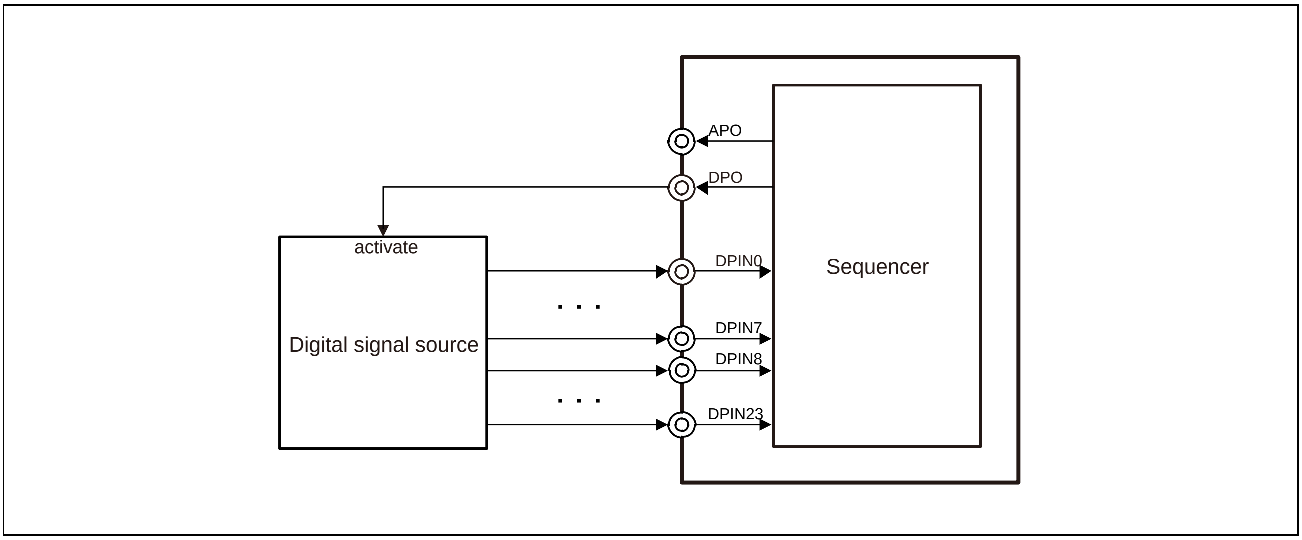


Figure 15.2 Direct Mode Connection Example

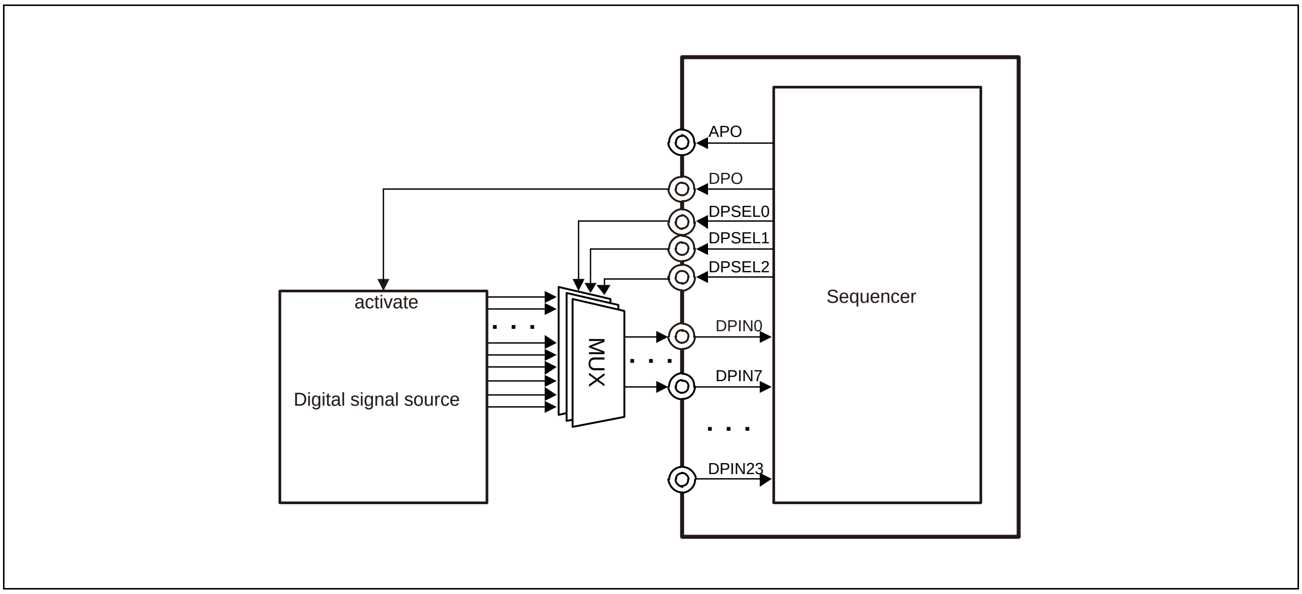


Figure 15.3 Multiplexer Mode Connection Example

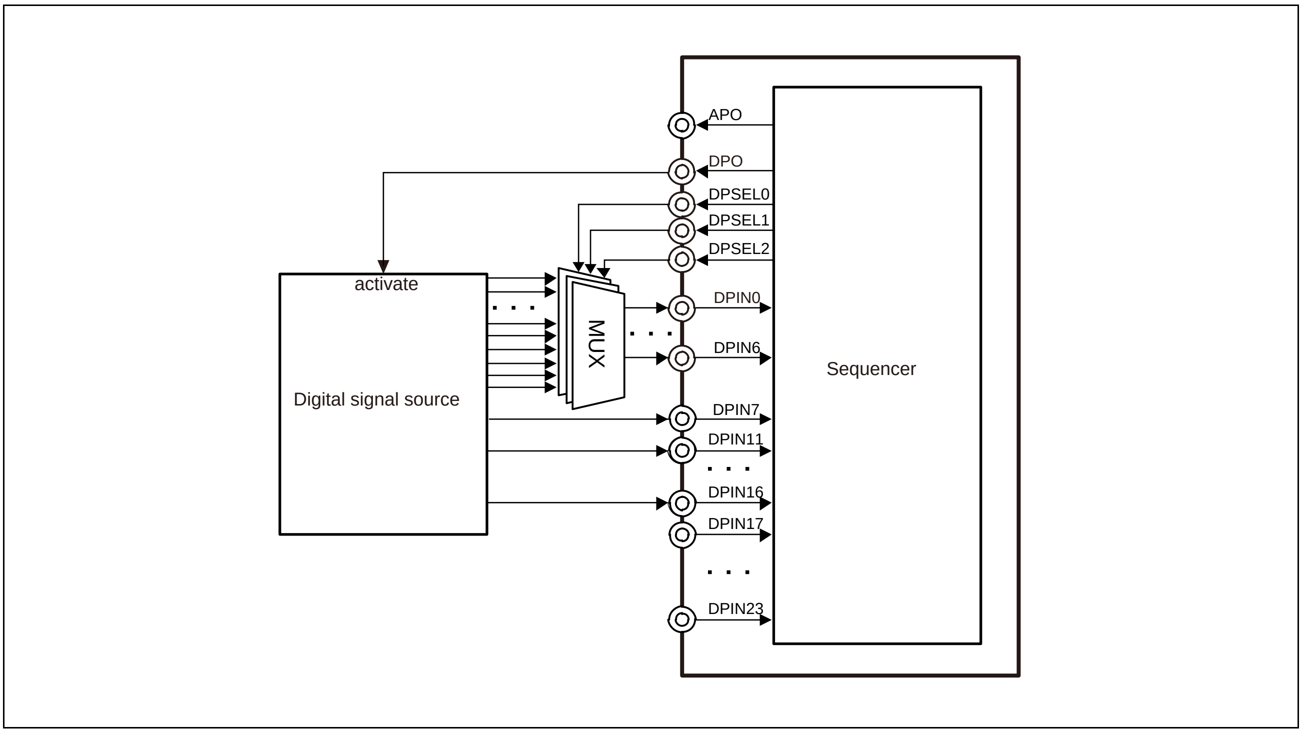


Figure 15.4 MIX Mode Connection Example

CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative function.

They cannot be used simultaneously.

Preparation

- Set NUMDP[2:0] and TJIS[2:0] bits in the SCTLR register to specify the number of times the port is to be read, and the TAUJy interrupt to be used as sequence start trigger.
- Set TAUJy to interval timer mode.
- Set the wait time of the digital signal source by using the lower 8 bits in the CNTVAL register.
- Set expected values in the DPDSR0, DPDSRM and DPDSRH registers.
- Set the ports to be checked in the DPSELR0, DPSELRM, and DPSELRH registers.

Start

- Start the TAUJy.
- Set the SCTLR.DPEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJy. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

Upon completion of checking all ports that have been set, the INTCWEND interrupt occurs. In addition, if the input value of the port is different from the expected value set by the DPDSR0, DPDSRM, or DPDSRH register, the wake-up factor WUTRG0 occurs. The following figures show an example of the operation in digital input mode.

Stop

To stop the LPS operation in Digital Input Mode (by changing the SCTLR.DPEN bit setting from 1 to 0), follow the procedure shown below. In this example, the P0_0 pin is used as DPO.

1. Set the port register to specify low level output on the pin (P0.P0_0 = 0).
2. Change the setting for the P0_0 pin from the alternative port mode to the port mode (PMC0.PMC0_0 = 0).
3. Set SCTLR.DPEN = 0.

NOTE

The above procedure applies when the P0_0 pin is used as DPO. If the P0_2 pin is used as DPO, specify the P0_2 pin settings in the same way.

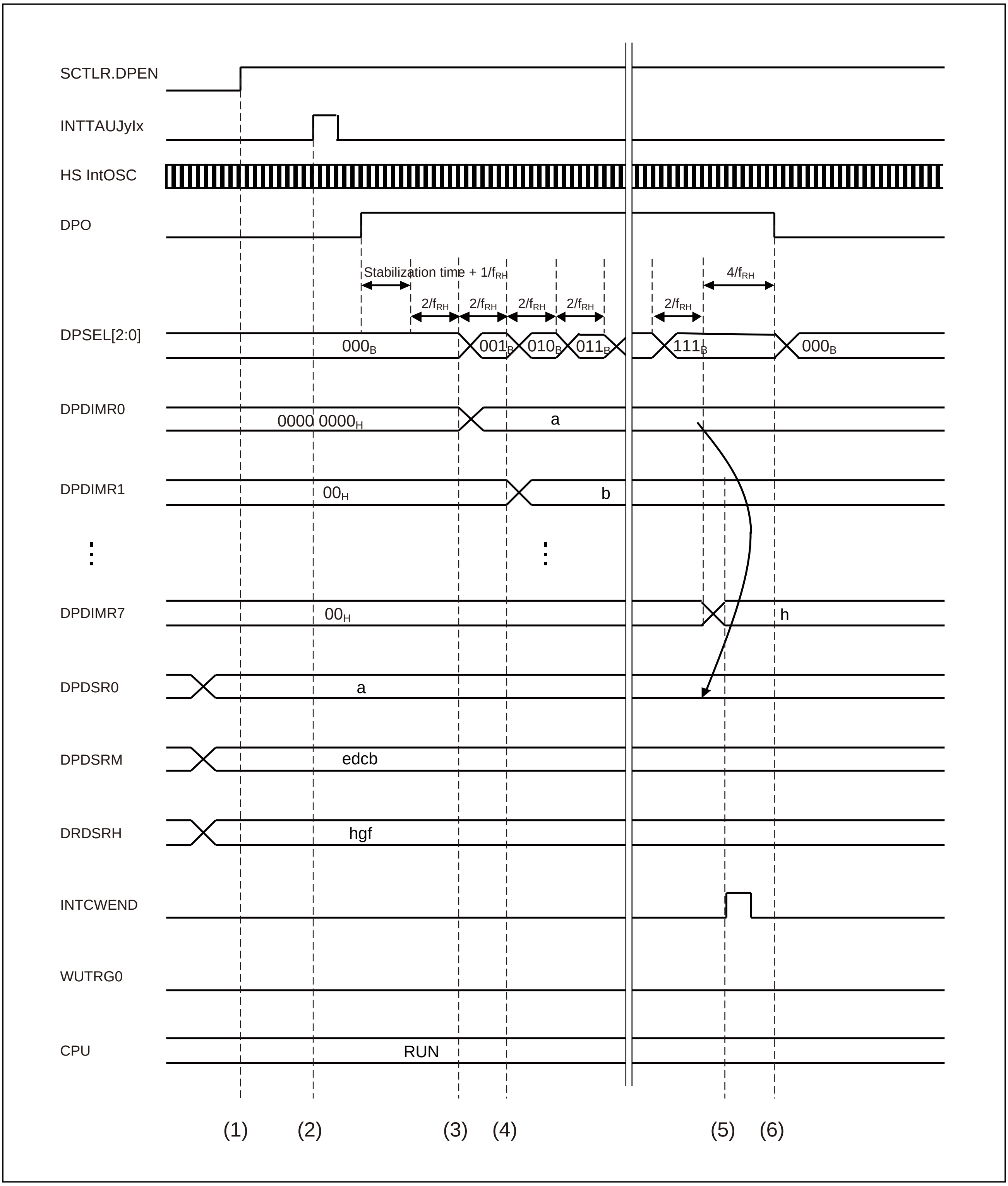


Figure 15.5 Operation of Digital Input Mode (8 Ports × 8) when the Input Value is not Changed (RUN Mode)

(1) Set the SCTLR.DPEN bit to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the sequencer outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(3) After the completion of the signal source stabilization, the LPS stores the DPIN[7:0] input value to the DPDIMR0 register and increments the DPSEL[2:0] pins to switch the external multiplexer.

(4) After the switching of the DPSEL[2:0] pins, the LPS stores the values in the DPDIMRn registers in order from DPDIMR1 and continues to increment the DPSEL[2:0] pins.

(5) After the value is stored up to the DPDIMR7 register, the INTCWEND interrupt is generated and the value is compared with the expected value set in the DPDSR0, DPDSRM, and DPDSRH registers.

(6) When the value is not different from the expected value, the wake-up factor WUTRG0 is not generated. The LPS stops the DPO output and returns to the waiting state for the trigger.

High level width of the DPO pin is calculated by the following formula. For the Stabilization time, see **Section 15.3.18, CNTVAL — Count Value Register.**

High level width of the DPO pin

$$= \text{Stabilization time} + (1/f_{RH}) \times 1 + (1/f_{RH}) \times 2 \times (\text{SCTLR.NUMDP (set value)} + 1) + (1/f_{RH}) \times 4$$

$$= \text{Stabilization time} + (1/f_{RH}) \times (2 \times \text{SCTLR.NUMDP (set value)} + 7)$$

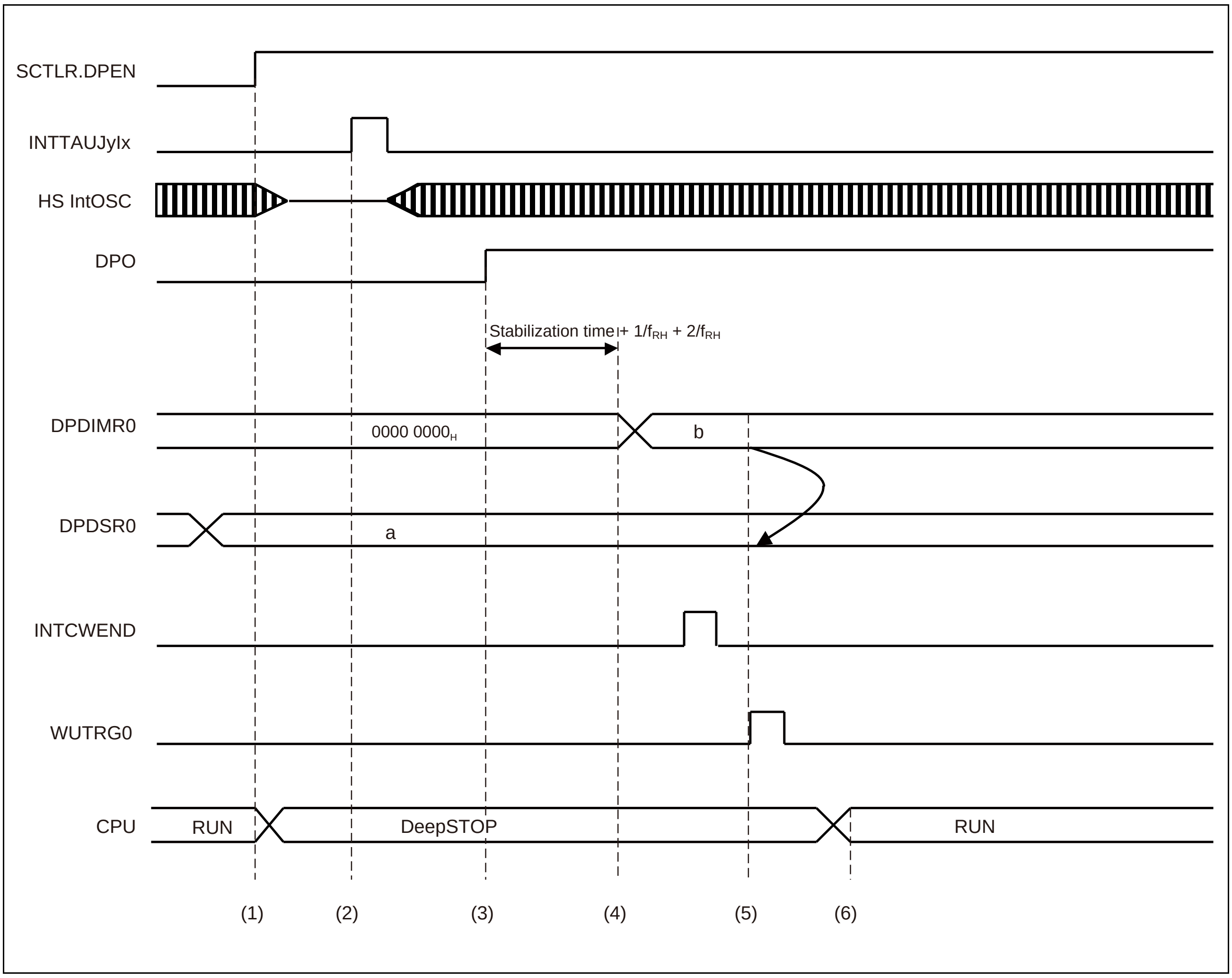


Figure 15.6 Operation of Digital Input Mode (24 Ports × 1) when the Input Value is Changed (DeepSTOP Mode)

(1) Set the STBC0PSC.STBC0DISTRG bit to 1 to transition to the DeepSTOP mode, while the SCTLR.DPEN bit is set to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(3) After the completion of the HS IntOSC stabilization time, the LPS outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(4) After the completion of the signal source stabilization, the LPS stores the DPIN[23:0] input value to the DPDIMR0 register and the INTCWEND interrupt is generated.

(5) The value stored in the DPDIMR0 register is compared with the expected value set in the DPDSR0 register. When the value is different from the expected value, the wake-up factor WUTRG0 is generated.

(6) The CPU returns to RUN mode at the generation of WUTRG0. The DPO pin is driven high until the EVFR.DINEVF bit is cleared to 0 by software.

15.4.1 Digital Port Error Interrupt

A level sensitive interrupt indicating a data comparison mismatch is generated. This interrupt is generated not only in stand-by mode but also in RUN mode. The set and clear conditions are shown below.

Table 15.39 Digital Port Error Interrupt

Unit Interrupt Signal	Set Condition	Clear Condition
INTDPE	When EVFR.DINEVF is set to 1 by hardware	When EVFR.DINEVF is cleared to 0 by software

15.5 Analog Input Mode

The analog input port ADCA0Im (m = 0 to 15) can be monitored.

TAUJy is used to set the timing to check the value input to the port.

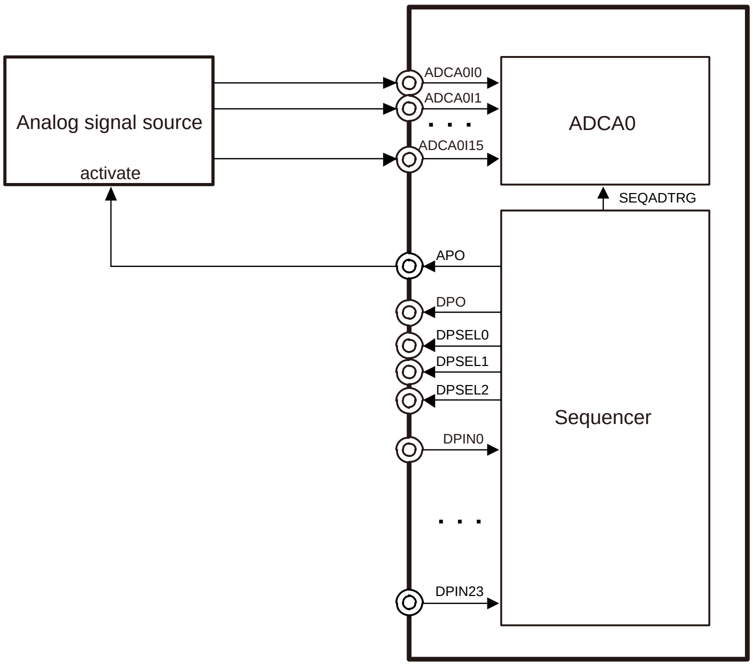


Figure 15.7 Analog Input Mode Connection Example 1

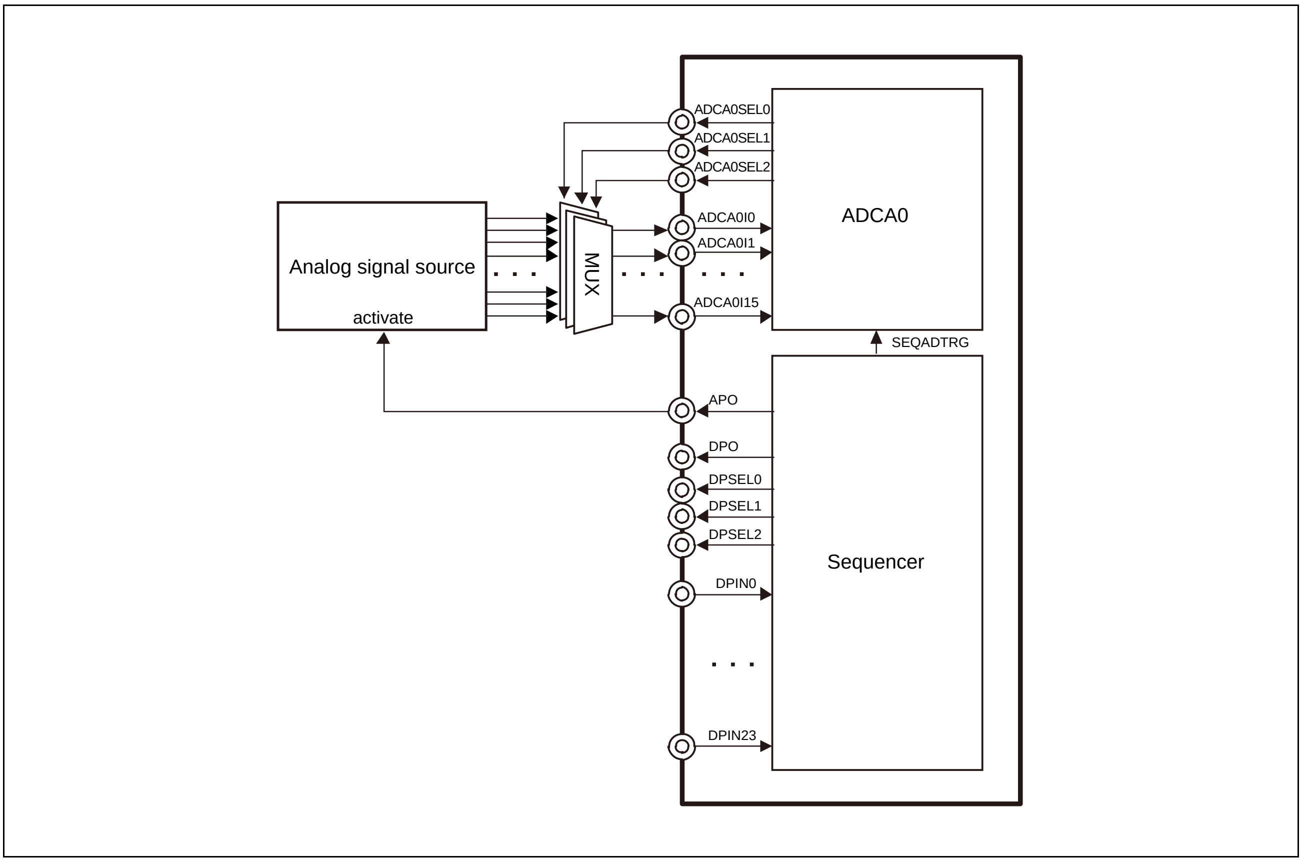


Figure 15.8 Analog Input Mode Connection Example 2

Preparation

- Set TJIS[2:0] bits in the SCTLR register to specify the TAUJy interrupt to be used as sequence start trigger.
- Set TAUJy to interval timer mode.
- Set the wait time of the analog signal source by using the upper 8 bits in the CNTVAL register.
- Set the ADCA0.

CAUTIONS

1. When the LPS is in use, the A/D conversion completion interrupt (INTADCA0Ix_{-1}) should be output after the conversion of all channels of the LPS has been completed. The setting is as follows.
    - Set the ADIE bit in virtual channel register j (ADCA0VCRj) to 0 (a scan group x end interrupt (INTADCA0Ix_{-1}) is not generated when A/D conversion for virtual channel j ends in SGx.).
    - Set the ADIE bit in the scan group x control register (ADCA0SGCRx) to 1 (INTADCA0Ix_{-1} is output when the scan for SGx ends).
2. Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[2:0] bits to the completion of A/D conversion for all channels of the LPS, only proceed with A/D conversion for PWM-Diag. LPS acknowledges the A/D conversion completion interrupts of the scan group SG1, SG2 or SG3, but does not recognize the kind of SG (three interrupts are ORed to one interrupt). During LPS operation using one SG, an A/D conversion completion interrupt and A/D error interrupt cannot be set for another SG.
3. Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[2:0] bits to the completion of A/D conversion for all channels of LPS, do not forcibly end A/D conversion by using the ADCA0ADHALTR.HALT bit.

4. When the LPS is in use, do not use the following modes.
   - Continuous scan mode
     (the setting ADCA0SGCRx.SCANMD = 1 is prohibited)
   - Multicycle scan mode with 2 or more cycles
     (the settings ADCA0SGMCYCRx.MCYC = 01_B and 11_B are prohibited)
   - Channel repeat mode with 2 or more cycles
     (the settings ADCA0SGCRx.SCT = 01_B and 10_B are prohibited)

Start

- Start the TAUJy.
- Set the SCTLR.ADEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJy. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

To detect whether the analog input value differs from the expected value, use the A/D error interrupt request (INTADCA0ERR) of the A/D converter.

In addition, if the analog input value is different from the expected value, the wake-up factor WUTRG1 occurs.

For details on the A/D error interrupt request (INTADCA0ERR), see **Section 38.4.13, A/D Error Interrupt Request***[1]. The following figures show an example of the operation in analog input mode.

Note 1. In **Section 38, A/D Converter (ADCA)**, the name of the A/D error interrupt request is described as "INT_ADE".

Stop

To stop the LPS operation in Analog Input Mode (by changing the SCTLR.ADEN bit setting from 1 to 0), follow the procedure shown below. Note that the P0_1 pin is used as APO.

1. Set the port register to specify low level output on the pin (P0.P0_1 = 0).
2. Change the setting for the P0_1 pin from the alternative port mode to the port mode (PMC0.PMC0_1 = 0).
3. Set SCTLR.ADEN = 0.

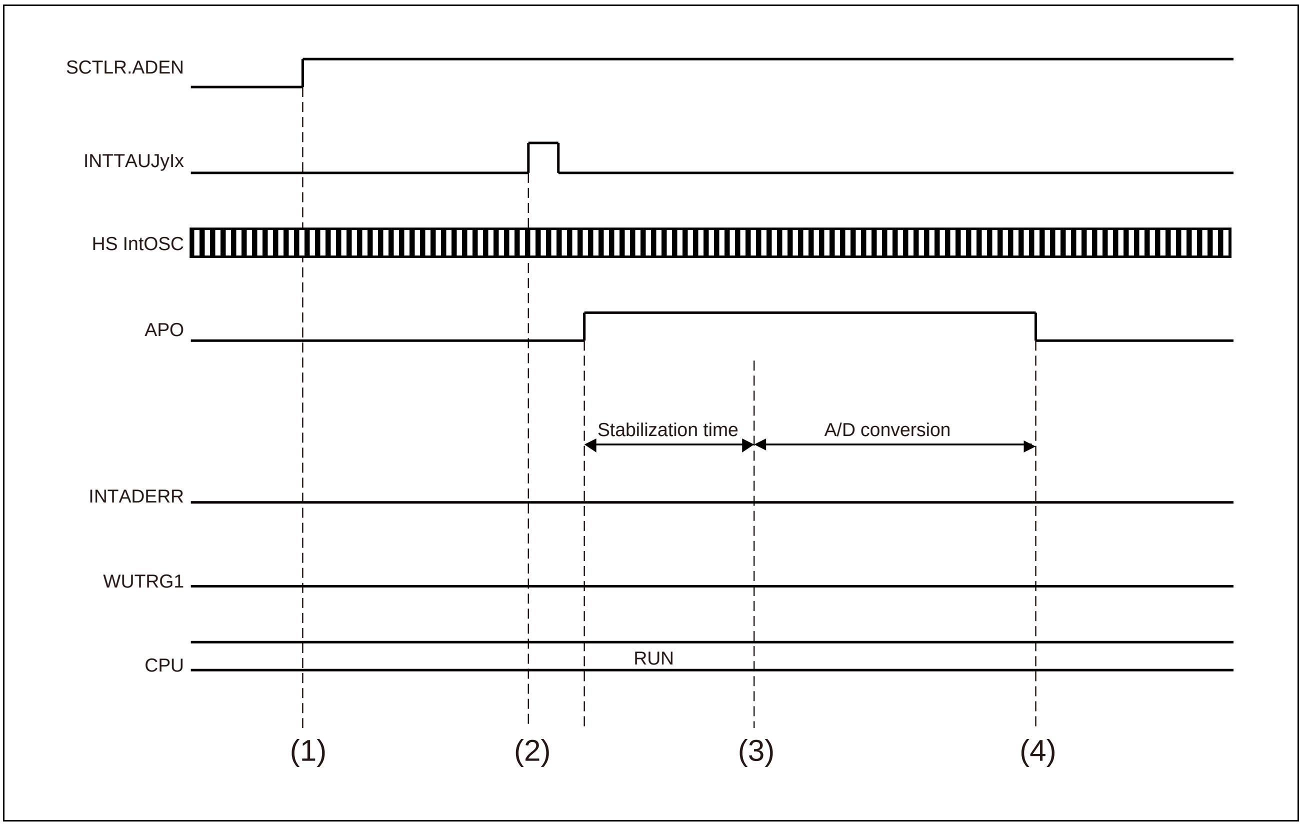


Figure 15.9 Operation of Analog Input Mode when the Conversion Result is within the Expected Range (RUN Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.

(3) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im (m = 0 to 15), set in the A/D converter scan group, is started.

(4) When the INTADCA0ERR interrupt is not generated as a result of A/D conversion, the LPS halts the A/D converter and resets the APO pin.

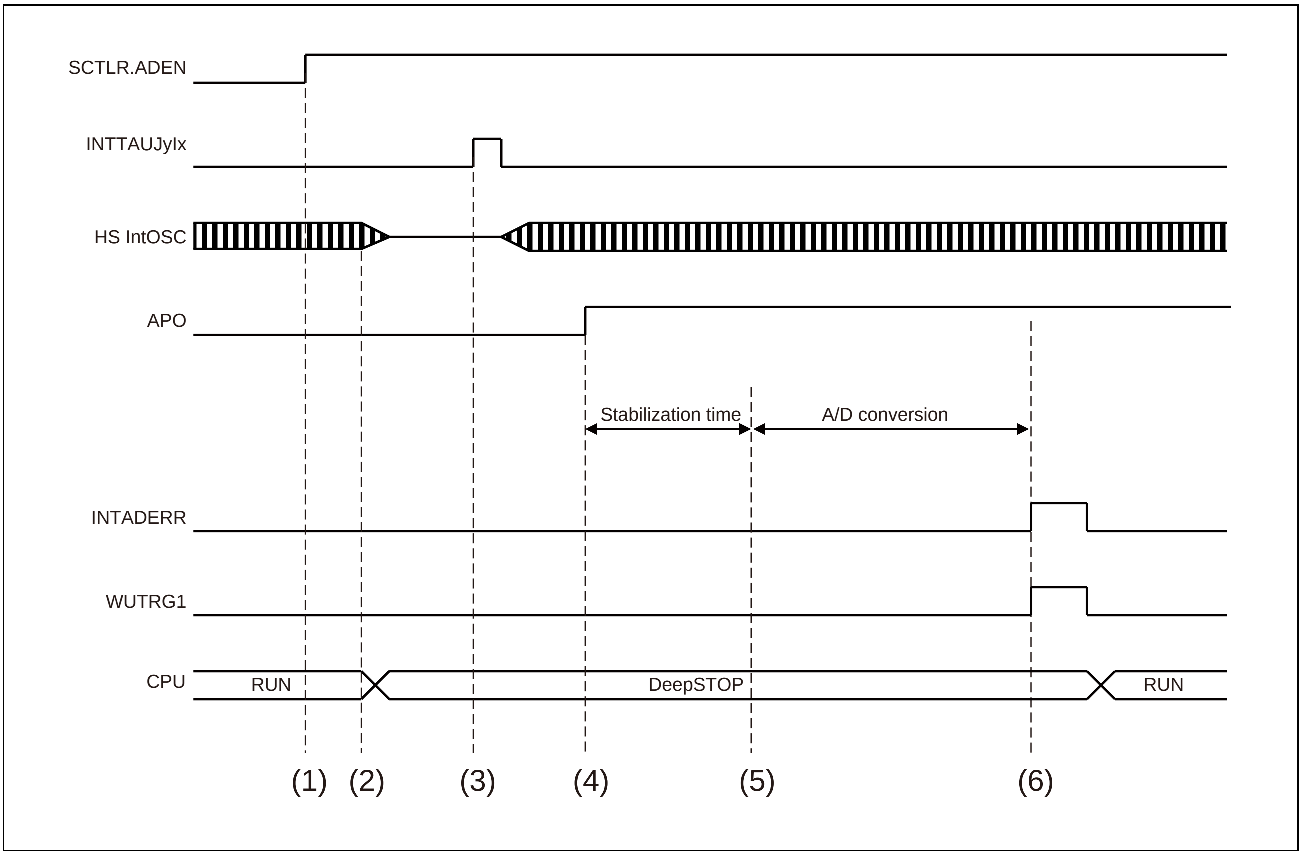


Figure 15.10 Operation of Analog Input Mode when the Conversion Result is not within the Expected Range (DeepSTOP Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) Set the STBC0PSC.STBC0DISTRG bit to 1 by software to transition to the DeepSTOP mode.

(3) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(4) After the completion of the HS IntOSC stabilization, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.

(5) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im (m = 0 to 15), set in the A/D converter scan group, is started.

(6) When the INTADCA0ERR interrupt is generated as a result of A/D conversion, the wake-up factor WUTRG1 is generated and the CPU returns to RUN mode. The APO pin is driven high until the upper limit/lower limit error flag of the A/D converter is cleared to 0 by software.

NOTE

WUTRG1 is generated when the first ADCA0 SG interrupt occurred after ADCAnULER.UE or ADCAnULER.LE is set.

15.5.1 Analog Port Error Interrupt

A level sensitive interrupt indicating a data comparison mismatch is generated. This interrupt is generated not only in the stand-by mode but also in RUN mode. The set and clear conditions are shown below.

Table 15.40 Analog Port Error Interrupt

Unit Interrupt Signal	Set Condition	Clear Condition
INTAPE	When WUTRG1 is set to 1 by hardware	When both ADCAnULER.UE and ADCAnULER.LE are cleared to 0 by software

Section 16 External Memory Access Controller (MEMC)

This section contains a generic description of the external memory access controller (MEMC).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of MEMC.

16.1 Features of RH850/F1KH, RH850/F1KM MEMC

16.1.1 Number of Units

This microcontroller has the following number of MEMC units.

Table 16.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	MEMCn (n = 0)	MEMCn (n = 0)	MEMCn (n = 0)

Table 16.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	—	—	1	1	1
Name	—	—	MEMCn (n = 0)	MEMCn (n = 0)	MEMCn (n = 0)

Table 16.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	—	—
Name	—	—	—	—

Table 16.4 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual MEMC units are identified by the index “n” (n = 0).
x	Throughout this section, chip select areas are identified by “x”. For example, the external memory area of CSx is described as the CSx (x = 0 to 3) area.

Table 16.5 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual MEMC units are identified by the index “n” (n = 0).
x	Throughout this section, chip select areas are identified by “x”. For example, the external memory area of CSx is described as the CSx (x = 0 to 3) area.

Table 16.6 External Memory Access Functions (RH850/F1KH-D8)

Control Signal Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Address output	23 bits	23 bits	24 bits
Chip select output	4	4	4
Data bus	16 bits	16 bits	16 bits

Table 16.7 External Memory Access Functions (RH850/F1KM-S4)

Control Signal Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Address output	—	—	23 bits	23 bits	24 bits
Chip select output	—	—	4	4	4
Data bus	—	—	16 bits	16 bits	16 bits

16.1.2 Register Base Address

The MEMC base address is listed in the following table.

The MEMC register addresses are given as an offset from the base address.

Table 16.8 Register Base Address (RH850/F1KH-D8)

Base Address Name	Base Address
<MEMC0_base>	1003 0000_H

Table 16.9 Register Base Address (RH850/F1KM-S4)

Base Address Name	Base Address
<MEMC0_base>	1003 0000_H

16.1.3 Clock Supply

The MEMC clock supply is shown in the following table.

Table 16.10 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name	Description
MEMCn	MEMC0CLK	CPUCLK_L*1	Communication clock
	Register access clock	CPUCLK_L*1	Bus clock

Note 1. This supply clock runs at half the frequency of the CPUCLK_L signal.

Table 16.11 Clock Supply (RH850/F1KM-S4)

Unit Name	Unit Clock Name	Supply Clock Name	Description
MEMCn	MEMC0CLK	CPUCLK_L*1	Communication clock
	Register access clock	CPUCLK_L*1	Bus clock

Note 1. This supply clock runs at half the frequency of the CPUCLK_L signal.

16.1.4 Reset Sources

The MEMC reset sources are shown below. MEMC is initialized by the following reset sources.

Table 16.12 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
MEMCn	All reset sources (ISORES)

Table 16.13 Reset Sources (RH850/F1KM-S4)

Unit Name	Reset Source
MEMCn	All reset sources (ISORES)

16.1.5 External Input/Output Signals

External input/output signals of MEMC are listed below.

Table 16.14 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
MEMC0		
MEMC0A[23:16]	Address Bus Output Signal	MEMC0A[23:16]
MEMC0AD[15:0]	Address/Data Bus I/O Signal	MEMC0AD[15:0]
MEMC0ASTB	Address Strobe Output Signal	$\overline{\text{MEMC0ASTB}}$
MEMC0BEN[1:0]	Byte Enable Output Signal	$\overline{\text{MEMC0BEN[1:0]}}$
MEMC0CLK	Bus Clock Output Signal	MEMC0CLK
MEMC0CS[3:0]	Chip Select Output Signal	$\overline{\text{MEMC0CS[3:0]}}$
MEMC0RD	Read Strobe Output Signal	$\overline{\text{MEMC0RD}}$
MEMC0WAIT	External Wait Request Input Signal	$\overline{\text{MEMC0WAIT}}$
MEMC0WR	Write Strobe Output Signal	$\overline{\text{MEMC0WR}}$

Note: For the port to be used as MEMC, set the output driver strength to high (PDSCn_m = 1).

Table 16.15 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
MEMC0		
MEMC0A[23:16]	Address Bus Output Signal	MEMC0A[23:16]
MEMC0AD[15:0]	Address/Data Bus I/O Signal	MEMC0AD[15:0]
MEMC0ASTB	Address Strobe Output Signal	$\overline{\text{MEMC0ASTB}}$
MEMC0BEN[1:0]	Byte Enable Output Signal	$\overline{\text{MEMC0BEN[1:0]}}$
MEMC0CLK	Bus Clock Output Signal	MEMC0CLK
MEMC0CS[3:0]	Chip Select Output Signal	$\overline{\text{MEMC0CS[3:0]}}$
MEMC0RD	Read Strobe Output Signal	$\overline{\text{MEMC0RD}}$
MEMC0WAIT	External Wait Request Input Signal	$\overline{\text{MEMC0WAIT}}$
MEMC0WR	Write Strobe Output Signal	$\overline{\text{MEMC0WR}}$

Note: For the port to be used as MEMC, set the output driver strength to high (PDSCn_m = 1).

16.2 Overview

The external memory access controller provides four chip select areas, and wait time is selectable in each chip select area.

The bus clock runs at half the frequency of the MEMC supply clock.

16.2.1 Functional Overview

The main features of the external memory access controller:

- Multiplexed bus mode
- Four chip select areas and 16-bit bus width
- The data endian format can be selected for each chip select area individually.
- Various wait functions can be set individually for each chip select area.
- External wait on SRAM access cycles
- External wait error detection

16.2.1.1 Multiplexed Bus

This is an operation mode that connects address output and data input/output to external memory using the same signal line, making it possible to reduce the number of pins required for external memory connection.

16.2.1.2 Chip Select Output Function

The external bus area of the memory space is divided into four chip select areas, and a chip select signal can be output for each chip select area. The allocation of these chip select areas is fixed by the system and cannot be changed through programming.

In addition, the bus width is 16-bit fixed.

16.2.1.3 Data Endian Setting Function

The data endian (little endian/big endian) can be specified for each chip select area.

16.2.1.4 Programmable Wait Setting Functions

This microcontroller has the following wait functions, which can be set for each chip select area.

- Programmable data wait
- Data hold wait
- Data setup wait
- Address setup wait
- Address hold wait
- Idle cycle

16.2.1.5 External Wait Function

Waits of any width can be inserted externally from the $\overline{\text{MEMC0WAIT}}$ pin. The $\overline{\text{MEMC0WAIT}}$ pin is sampled just before the data output cycle, and the data latch timing can be delayed by any amount.

16.2.1.6 External Wait Error Detection Function

This microcontroller has a function to forcibly cancel a wait state if an external wait is continuously input for 128 clock cycles, to prevent the system hanging up if an external wait is continuously input due to a defect of the $\overline{\text{MEMC0WAIT}}$ pin.

16.3 Registers

16.3.1 List of Registers

The MEMC registers are listed in the following table.

For details on <MEMCn_base>, see **Section 16.1.2, Register Base Address**.

Table 16.16 List of Registers

Module Name	Register Name	Symbol	Address
MEMCn	Data endian configuration register	DEC	<MEMCn_base> + 02_H
	Data wait configuration register	DWC	<MEMCn_base> + 08_H
	Data hold wait configuration register	DHC	<MEMCn_base> + $0C_H$
	Data setup wait configuration register	DSC	<MEMCn_base> + $0E_H$
	Address wait configuration register	AWC	<MEMCn_base> + 10_H
	Idle cycle configuration register	ICC	<MEMCn_base> + 14_H
	External wait error configuration register	EWC	<MEMCn_base> + $1A_H$

16.3.2 DEC — Data Endian Configuration Register

The DEC register is used to set the endianness for the external bus.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 02$_{\text{H}}$

Value after reset: 0000$_{\text{H}}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE3	—	DE2	—	DE1	—	DE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 16.17 DEC Register Contents

Bit Position	Bit Name	Function
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DE3	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	DE2	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	DE1	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DE0	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian

The relationship between each chip select area and the control bits is shown below.

Table 16.18 Relationship between DEx Bit and Chip Select Area

Chip Select Area	DEx Bit
CS3 area	DE3
CS2 area	DE2
CS1 area	DE1
CS0 area	DE0

16.3.3 DWC — Data Wait Configuration Register

The DWC register is used to set the number of data wait states for the external bus.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 08$_H$

Value after reset: FFFF$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DW33	DW32	DW31	DW30	DW23	DW22	DW21	DW20	DW13	DW12	DW11	DW10	DW03	DW02	DW01	DW00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.19 DWC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>DWx3,
DWx2,
DWx1,
DWx0</td><td>Data Wait Setting
These bits set the number of data wait states for each chip select area.
<table>
<tr><th>DWx3</th><th>DWx2</th><th>DWx1</th><th>DWx0</th><th>Number of Data Wait States</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15 clock cycles</td></tr>
</table>
</td></tr>
</table>

The relationship between each chip select area and the control bits is shown below.

Table 16.20 Relationship between DWx3 to DWx0 Bits and Chip Select Area

Chip Select Area	DWx3 to DWx0 Bits
CS3 area	DW33, DW32, DW31, DW30
CS2 area	DW23, DW22, DW21, DW20
CS1 area	DW13, DW12, DW11, DW10
CS0 area	DW03, DW02, DW01, DW00

16.3.4 DHC — Data Hold Wait Configuration Register

The DHC register is used to set the number of data hold waits for the external bus.

By setting this register, a data hold wait of “DHC register setting + one cycle” is inserted in a write cycle.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $0C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DH31	DH30	DH21	DH20	DH11	DH10	DH01	DH00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.21 DHC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 8</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>7 to 0</td><td>DHx1, DHx0</td><td>Data Hold Wait Setting
These bits set the number of data wait states for each chip select area.
<table>
<tr><th>DHx1</th><th>DHx0</th><th>Number of Data Hold Wait</th></tr>
<tr><td>0</td><td>0</td><td>No data hold wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table>
</td></tr>
</table>

The relationship between each chip select area and the control bits is shown below.

Table 16.22 Relationship between DHx1, DHx0 Bits and Chip Select Area

Chip Select Area	DHx1, DHx0 Bits
CS3 area	DH31, DH30
CS2 area	DH21, DH20
CS1 area	DH11, DH10
CS0 area	DH01, DH00

16.3.5 DSC — Data Setup Wait Configuration Register

The DSC register is used to set the number of data setup wait states for the external bus.

This wait is inserted in a write cycle.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $0E_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DS31	DS30	DS21	DS20	DS11	DS10	DS01	DS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.23 DSC Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	DSx1, DSx0	Data Setup Wait Setting These bits set the number of data setup wait states for each chip select area. <table><tr><th>DSx1</th><th>DSx0</th><th>Number of Data Setup Wait States</th></tr><tr><td>0</td><td>0</td><td>No data setup wait</td></tr><tr><td>0</td><td>1</td><td>1 clock cycle</td></tr><tr><td>1</td><td>0</td><td>2 clock cycles</td></tr><tr><td>1</td><td>1</td><td>3 clock cycles</td></tr></table>

The relationship between each chip select area and the control bits is shown below.

Table 16.24 Relationship between DSx1, DSx0 Bits and Chip Select Area

Chip Select Area	DSx1, DSx0 Bits
CS3 area	DS31, DS30
CS2 area	DS21, DS20
CS1 area	DS11, DS10
CS0 area	DS01, DS00

16.3.6 AWC — Address Wait Configuration Register

The AWC register is used to set the address wait period of the external bus for each chip select area.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 10_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AHW31	AHW30	ASW31	ASW30	AHW21	AHW20	ASW21	ASW20	AHW11	AHW10	ASW11	ASW10	AHW01	AHW00	ASW01	ASW00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.25 AWC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>AHW31, AHW30</td><td>Address Hold Wait Setting
These bits set the number of address hold wait states for each chip select area.
<table>
<tr><th>AHW31</th><th>AHW30</th><th>Number of Address Hold Waits</th></tr>
<tr><td>0</td><td>0</td><td>No address hold wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>13, 12</td><td>ASW31, ASW30</td><td>Address Setup Wait Setting
These bits set the number of address setup wait states for each chip select area.
<table>
<tr><th>ASW31</th><th>ASW30</th><th>Number of Address Setup Wait</th></tr>
<tr><td>0</td><td>0</td><td>No address setup wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>11, 10</td><td>AHW21, AHW20</td><td>Address Hold Wait Setting
These bits set the number of address hold wait states for each chip select area.
<table>
<tr><th>AHW21</th><th>AHW20</th><th>Number of Address Hold Waits</th></tr>
<tr><td>0</td><td>0</td><td>No address hold wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>9, 8</td><td>ASW21, ASW20</td><td>Address Setup Wait Setting
These bits set the number of address setup wait states for each chip select area.
<table>
<tr><th>ASW21</th><th>ASW20</th><th>Number of Address Setup Wait</th></tr>
<tr><td>0</td><td>0</td><td>No address setup wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
</table>

Table 16.25 AWC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>AHW11, AHW10</td><td>Address Hold Wait Setting
These bits set the number of address hold wait states for each chip select area.
<table>
<tr><th>AHW11</th><th>AHW10</th><th>Number of Address Hold Waits</th></tr>
<tr><td>0</td><td>0</td><td>No address hold wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>ASW11, ASW10</td><td>Address Setup Wait Setting
These bits set the number of address setup wait states for each chip select area.
<table>
<tr><th>ASW11</th><th>ASW10</th><th>Number of Address Setup Wait</th></tr>
<tr><td>0</td><td>0</td><td>No address setup wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>AHW01, AHW00</td><td>Address Hold Wait Setting
These bits set the number of address hold wait states for each chip select area.
<table>
<tr><th>AHW01</th><th>AHW00</th><th>Number of Address Hold Waits</th></tr>
<tr><td>0</td><td>0</td><td>No address hold wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>ASW01, ASW00</td><td>Address Setup Wait Setting
These bits set the number of address setup wait states for each chip select area.
<table>
<tr><th>ASW01</th><th>ASW00</th><th>Number of Address Setup Wait</th></tr>
<tr><td>0</td><td>0</td><td>No address setup wait</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
</table>

The relationship between each chip select area and the control bits is shown below.

Table 16.26 Relationship between ASWx1, ASWx0, AHWx1, AHWx0 Bits, and Chip Select Area

Chip Select Area	ASWx1, ASWx0 and AHWx1, AHWx0 Bits
CS3 area	ASW31, ASW30, AHW31, AHW30
CS2 area	ASW21, ASW20, AHW21, AHW20
CS1 area	ASW11, ASW10, AHW11, AHW10
CS0 area	ASW01, ASW00, AHW01, AHW00

16.3.7 ICC — Idle Cycle Configuration Register

The ICC register is used to set the number of idle cycles of the external bus. The number of idle cycles can be set for each chip select area.

Access:	This register can be read or written in 16-bit units.
Address:	<MEMCn_base> + 14_H
Value after reset:	3333_H

CAUTION

The Idle cycles set by the ICC register is invalid during bus sizing cycle. However, even in bus sizing cycle, idle cycles are valid during read cycle.
In this case, the number of idle cycles to be inserted depends on the ICC register setting. At least 1 clock cycle is inserted even when “no idle cycle” is selected.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WIC31	WIC30	RIC31	RIC30	WIC21	WIC20	RIC21	RIC20	WIC11	WIC10	RIC11	RIC10	WIC01	WIC00	RIC01	RIC00
Value after reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.27 ICC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>WIC31, WIC30</td><td>Idle Cycle Setting Bits after Write Cycle
These bits set the number of idle cycles for each chip select area after a write cycle.
<table>
<tr><th>WIC31</th><th>WIC30</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>13, 12</td><td>RIC31, RIC30</td><td>Idle Cycle Setting Bits after Read Cycle
These bits set the number of idle cycles for each chip select area after a read cycle.
<table>
<tr><th>RIC31</th><th>RIC30</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>11, 10</td><td>WIC21, WIC20</td><td>Idle Cycle Setting Bits after Write Cycle
These bits set the number of idle cycles for each chip select area after a write cycle.
<table>
<tr><th>WIC21</th><th>WIC20</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
</table>

Table 16.27 ICC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>RIC21, RIC20</td><td>Idle Cycle Setting Bits after Read Cycle
These bits set the number of idle cycles for each chip select area after a read cycle.
<table>
<tr><th>RIC21</th><th>RIC20</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>WIC11, WIC10</td><td>Idle Cycle Setting Bits after Write Cycle
These bits set the number of idle cycles for each chip select area after a write cycle.
<table>
<tr><th>WIC11</th><th>WIC10</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>RIC11, RIC10</td><td>Idle Cycle Setting Bits after Read Cycle
These bits set the number of idle cycles for each chip select area after a read cycle.
<table>
<tr><th>RIC11</th><th>RIC10</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>WIC01, WIC00</td><td>Idle Cycle Setting Bits after Write Cycle
These bits set the number of idle cycles for each chip select area after a write cycle.
<table>
<tr><th>WIC01</th><th>WIC00</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>RIC01, RIC00</td><td>Idle Cycle Setting Bits after Read Cycle
These bits set the number of idle cycles for each chip select area after a read cycle.
<table>
<tr><th>RIC01</th><th>RIC00</th><th>Number of Idle Cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table></td></tr>
</table>

The relationship between each chip select area and the control bits is shown below.

Table 16.28 Relationship between RICx1/RICx0/WICx1/WICx0 Bits and Chip Select Area

Chip Select Area	RICx1, RICx0, WICx1, WICx0 Bits
CS3 area	RIC31, RIC30, WIC31, WIC30
CS2 area	RIC21, RIC20, WIC21, WIC20
CS1 area	RIC11, RIC10, WIC11, WIC10
CS0 area	RIC01, RIC00, WIC01, WIC00

16.3.8 EWC — External Wait Error Configuration Register

The EWC register is used to enable and disable the external wait error function.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $1A_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	EW3	—	EW2	—	EW1	—	EW0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 16.29 EWC Register Contents

Bit Position	Bit Name	Function
15 to 7	Reserved	When read, the value after reset is read. When writing, write the value after reset.
6	EW3	External wait error setting These bits are used to enable and disable the external wait error in each chip select area. 0: External wait error disabled 1: External wait error enabled When this function is enabled, a wait state is forcibly released and SYSERR exception is occurred if an external wait request is detected continuously for 128 clock cycles.
5	Reserved	When read, the value after reset is read. When writing, write the value after reset.
4	EW2	External wait error setting These bits are used to enable and disable the external wait error in each chip select area. 0: External wait error disabled 1: External wait error enabled When this function is enabled, a wait state is forcibly released and SYSERR exception is occurred if an external wait request is detected continuously for 128 clock cycles.
3	Reserved	When read, the value after reset is read. When writing, write the value after reset.
2	EW1	External wait error setting These bits are used to enable and disable the external wait error in each chip select area. 0: External wait error disabled 1: External wait error enabled When this function is enabled, a wait state is forcibly released and SYSERR exception is occurred if an external wait request is detected continuously for 128 clock cycles.
1	Reserved	When read, the value after reset is read. When writing, write the value after reset.
0	EW0	External wait error setting These bits are used to enable and disable the external wait error in each chip select area. 0: External wait error disabled 1: External wait error enabled When this function is enabled, a wait state is forcibly released and SYSERR exception is occurred if an external wait request is detected continuously for 128 clock cycles.

The relationship between each chip select area and the control bits is shown below.

Table 16.30 Relationship between EWx Bits and Chip Select Area

Chip Select Area	EWx
CS3 area	EW3
CS2 area	EW2
CS1 area	EW1
CS0 area	EW0

16.4 Functions

16.4.1 Bus Control Functions

16.4.1.1 Chip Select Output Function

The connected external memory area is divided into and managed in 4 chip select areas, as shown in **Figure 16.1, External Memory Map (RH850/F1KH-D8)**, **Figure 16.2, External Memory Map (RH850/F1KM-S4)**.

When a bus cycle is generated for the external bus, this microcontroller makes the $\overline{\text{MEMC0CS[3:0]}}$ output pin corresponding to the access target address active (low level), along with outputting the access target address from the MEMC0A[23:16] and MEMC0AD[15:0].

The various settings for the external bus, such as the number of wait/idle states, can all be made for each chip select area.

By using these functions, different types of memory can be connected to each chip select area.

The allocation of the chip select areas is fixed by the system and cannot be changed through programming.

The memory map is shown below.

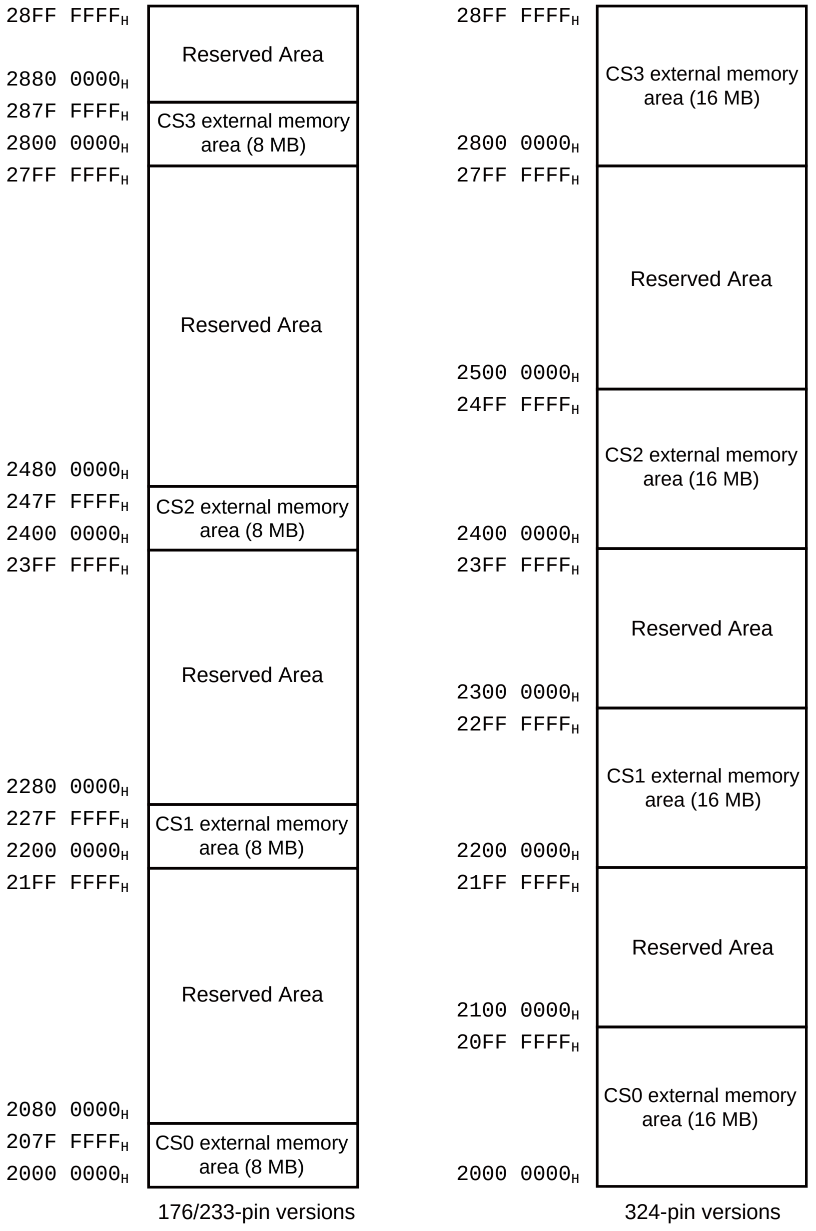


Figure 16.1 External Memory Map (RH850/F1KH-D8)

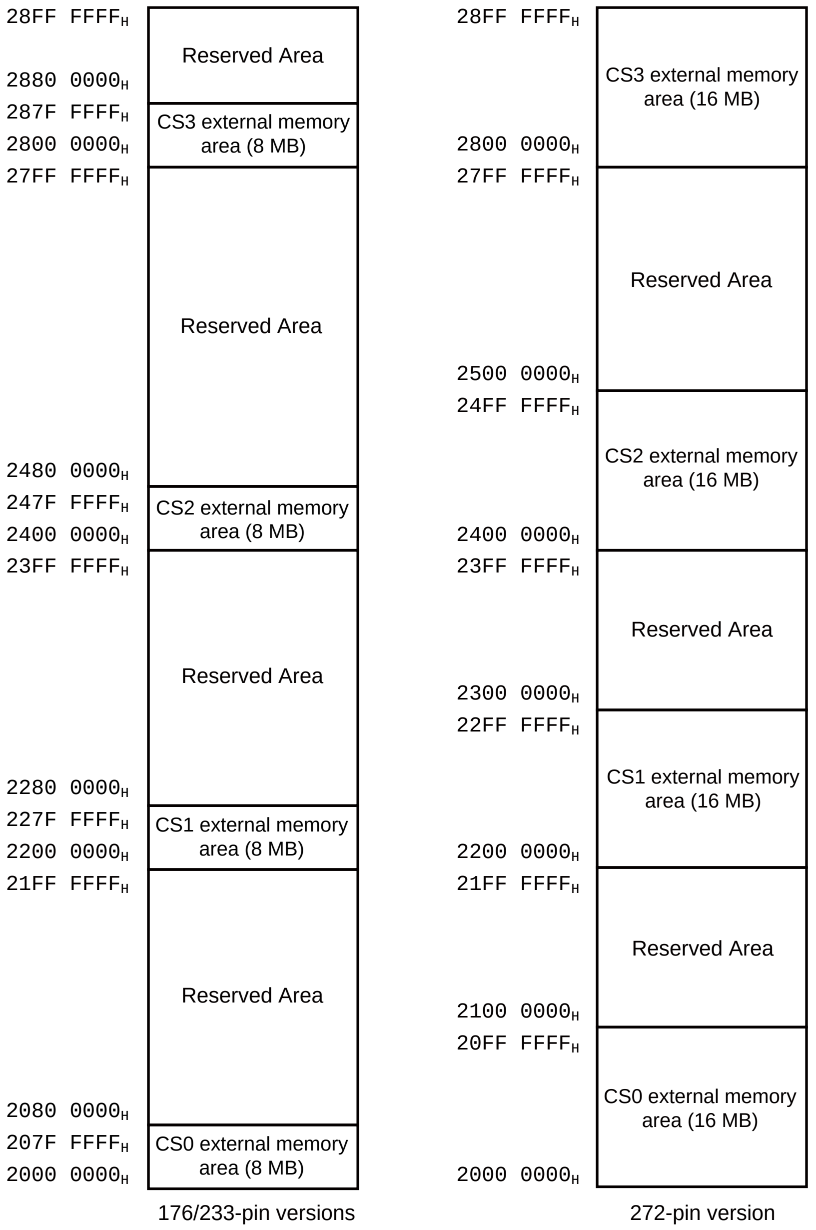
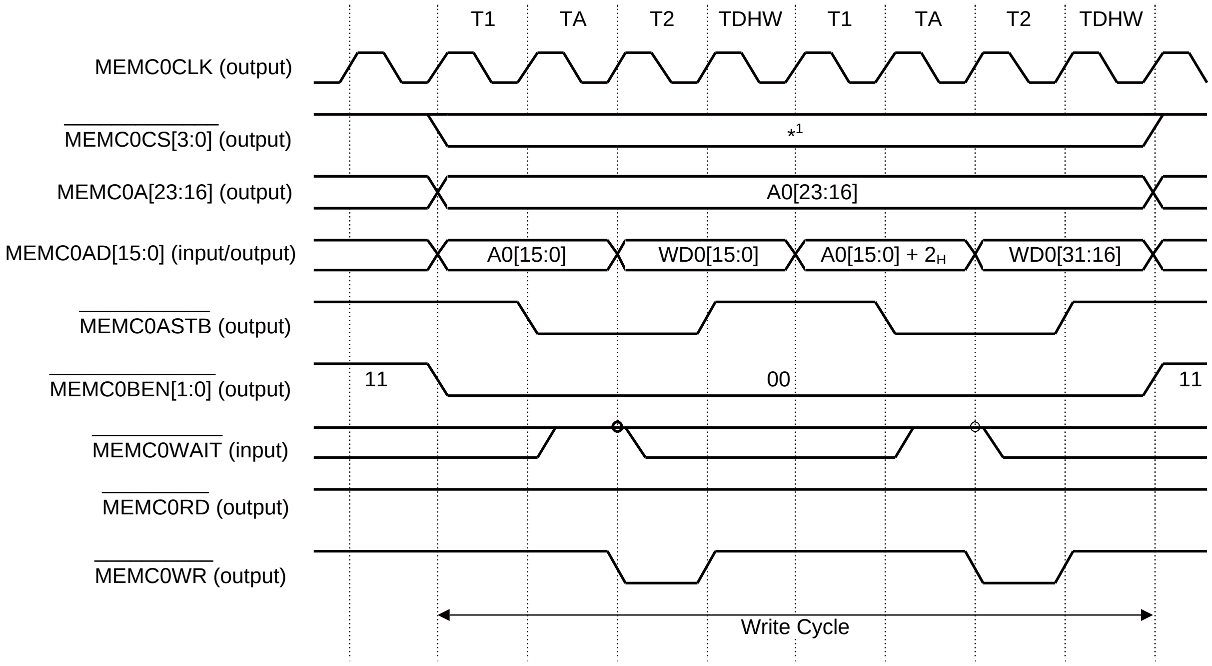


Figure 16.2 External Memory Map (RH850/F1KM-S4)

16.4.1.2 Bus Sizing Function

Access requests from the CPU (or DMA) are executed after being divided in accordance with the bit width of the external bus of the access destination.



Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Note 1. Indicates the chip select area to be accessed.

Figure 16.3 Bus Cycles when Making a 32-Bit Write Access to the External Bus

16.4.1.3 Data Endian Setting Function

Either little endian or big endian can be selected as the data endian of the external bus interface. This setting can be made for each chip select area with the DEC register.

CAUTIONS

1. In this microcontroller, instruction fetch operations in big endian are not supported. The assembler and debugger were designed for little endian operations.
2. Misaligned access in big endian format is not supported.

16.4.2 Wait Functions

Wait functions are listed below.

Table 16.31 Wait Functions

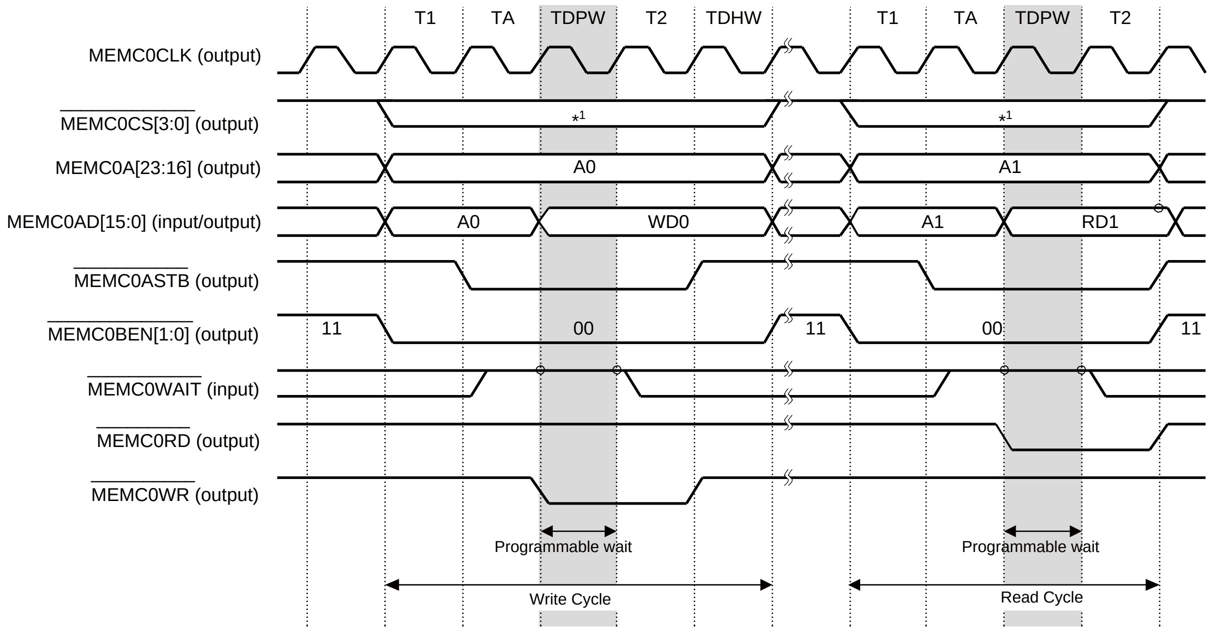
Wait Function	Data Wait		Data Hold Wait	Data Setup Wait	Address Wait	Idle State
	Programmable	External Pin				
Read access	✓	✓	—	—	✓	✓
Write access	✓	✓	✓	✓	✓	✓
Setting register	DWC	—	DHC	DSC	AWC	ICC
Max. number of waits	15	—	3	3	3	3

16.4.2.1 Programmable Data Wait Function

This wait function is used to delay the data latch timing by extending the read strobe and write strobe periods.

This function is enabled during data transfer. Up to 15 cycles can be inserted.

Data waits can be set for each chip select area using the DWC register.



Note: T_1: Address setup period

T_A: Address strobe period

T_2: Read (write) strobe period

T_{DPW}: Programmable data wait period

T_{DHW}: Data hold wait period

Note 1. Indicates the chip select area to be accessed.

Figure 16.4 Bus Cycles when Inserting Programmable Data Waits

16.4.2.2 External Wait Function

Data waits of any length can be inserted from the $\overline{\text{MEMC0WAIT}}$ pin.

The $\overline{\text{MEMC0WAIT}}$ pin input level is sampled immediately after completion of the TA cycle and the TDPW, TDEW cycles.

Data wait cycles obtained by ORing the programmable data wait set by the data wait control register (DWC register) and the external wait set by the $\overline{\text{MEMC0WAIT}}$ pin input, are inserted.

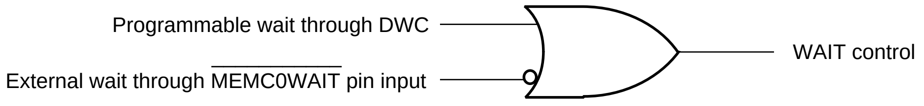
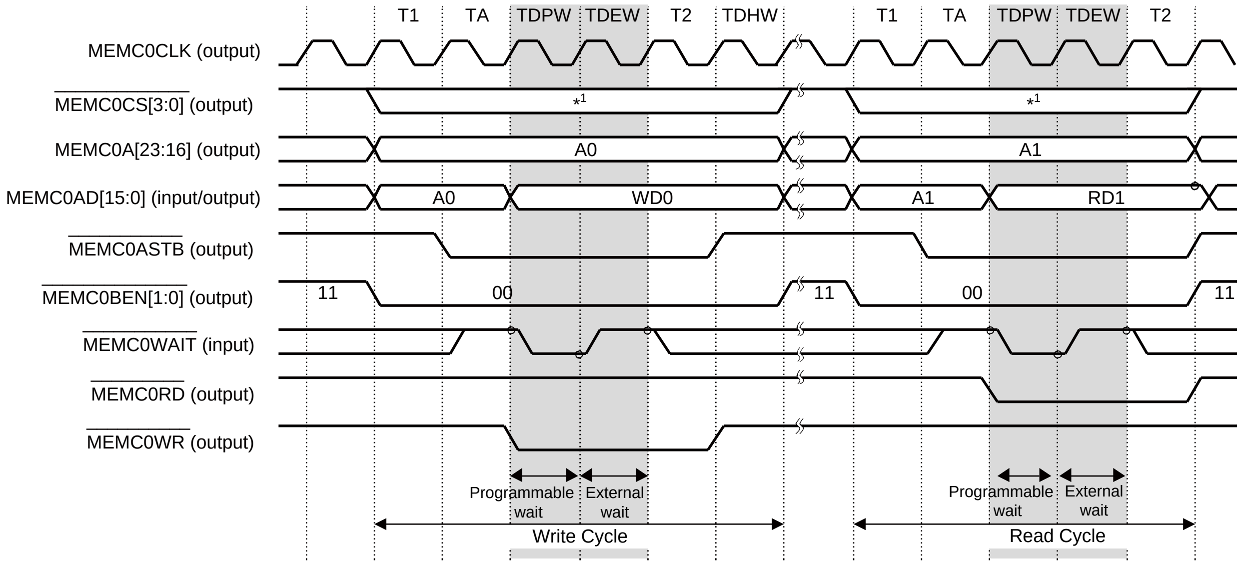


Figure 16.5 Internal Data Wait Generator



Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DPW}: Programmable data wait period
T_{DEW}: External data wait period
T_{DHW}: Data hold wait period

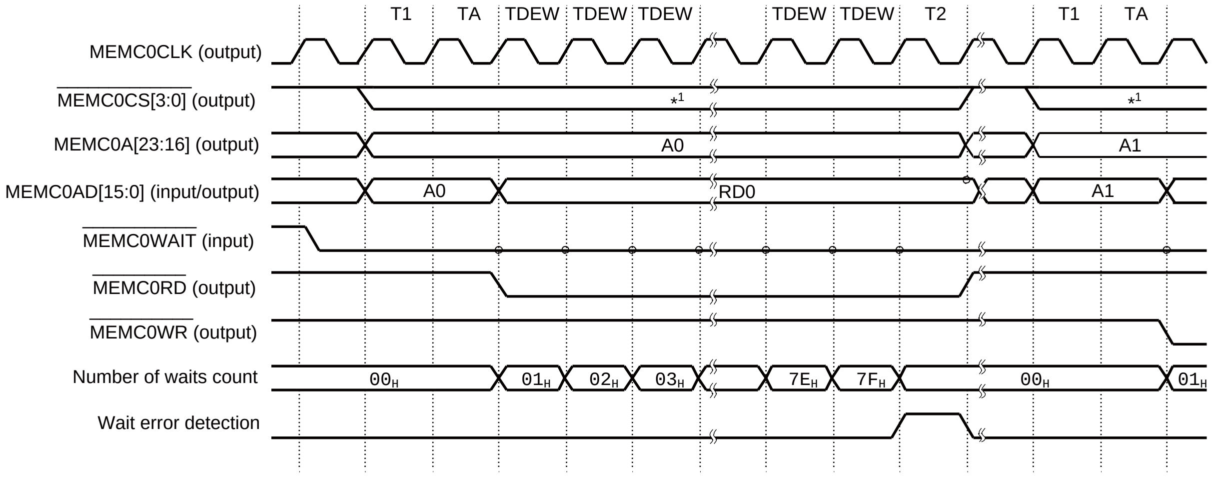
Note 1. Indicates the chip select area to be accessed.

Figure 16.6 Bus Cycles when Inserting Programmable Data Waits and External Waits (while DWC = 1)

16.4.2.3 External Wait Error Detection Function

By setting the EWx bit in the external wait error configuration register (EWC register) to 1, a wait state is forcibly canceled and SYSERR exception is generated if an external wait request is input continuously for 128 clock cycles.

This function prevents the system hanging up due to an unexpected failure of the $\overline{\text{MEMC0WAIT}}$ input pin.



Note: T_1: Address setup period
T_A: Address strobe period
T_{DEW}: External data wait period
T_2: Read (write) strobe period

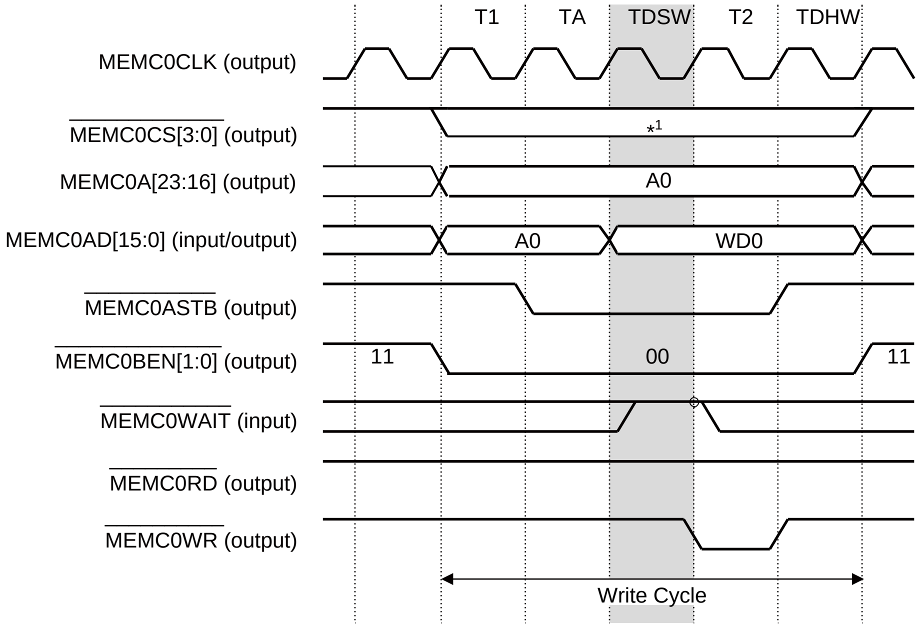
Note 1. Indicates the chip select area to be accessed.

Figure 16.7 Operation Timing during External Wait Error Detection

16.4.2.4 Data Setup Wait Function

This function inserts waits before a transfer state to secure setup time for data write strobe, and is only valid during a write cycle.

Waits of up to 3 cycles per chip select area can be inserted by setting the DSC register.



Note: T_1: Address setup period
T_A: Address strobe period
T_{DSW}: Data setup wait period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Note 1. Indicates the chip select area to be accessed.

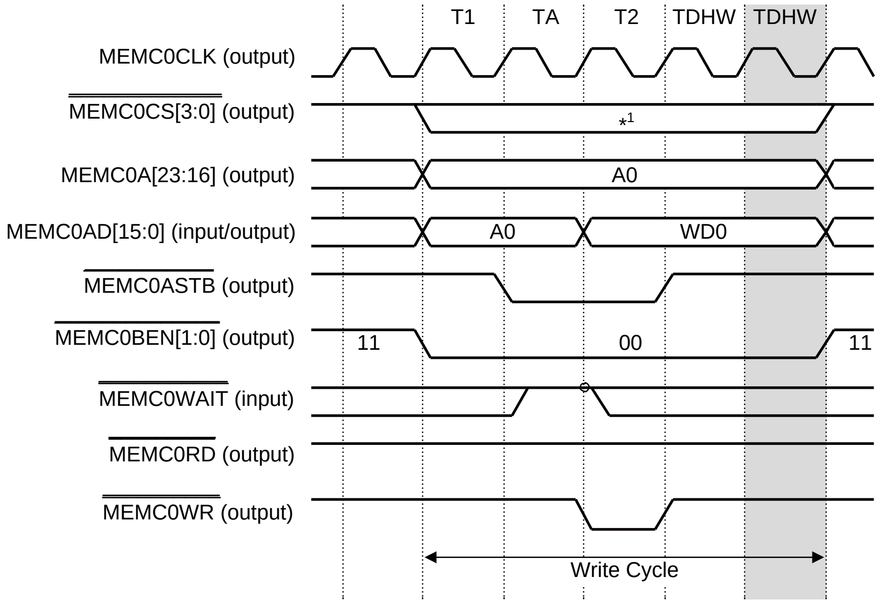
Figure 16.8 Bus Cycles when Inserting Data Setup Waits

16.4.2.5 Data Hold Wait Function

This function inserts a wait in the state following the rising edge of the write strobe signal in order to secure the hold time for the data write strobe.

This microcontroller always inserts 1 data hold wait state upon occurrence of a write cycle. This data hold wait can be extended by up to 3 cycles by setting the DHC register, allowing insertion of 4 cycles.

The number of cycles to extend the data hold wait can be set for each chip select area with the DHC register.



Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

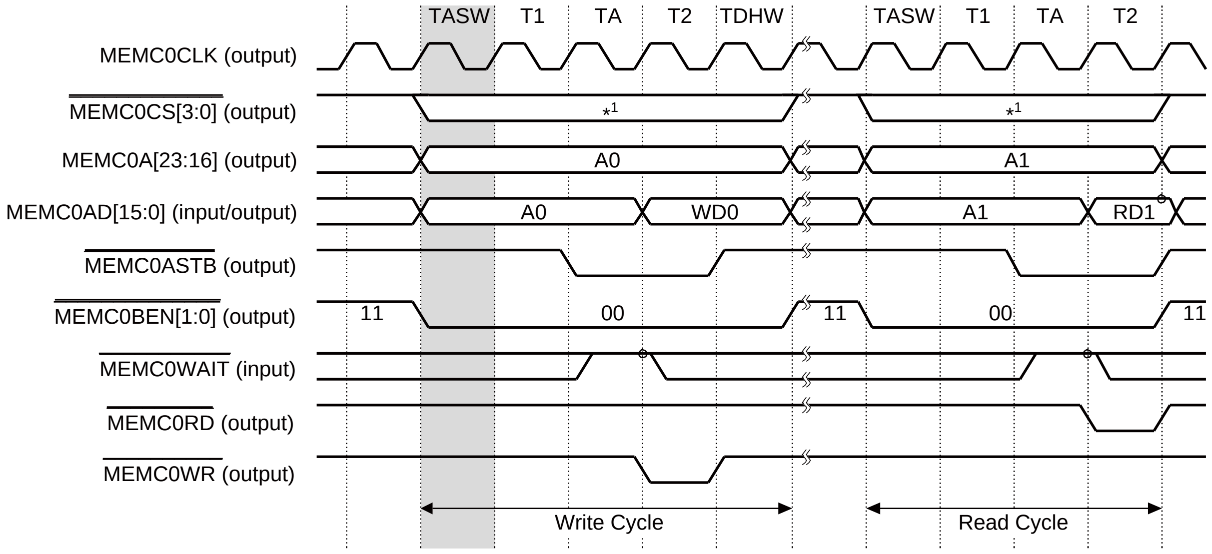
Note 1. Indicates the chip select area to be accessed.

Figure 16.9 Bus Cycles when Inserting Data Hold Waits

16.4.2.6 Address Setup Wait Function

The address setup wait function inserts a wait before the address transfer state in order to secure the setup time for the address strobe.

Up to 3 wait cycles can be inserted for each chip select area by setting the AWC register.



Note: T1: Address setup period
TASW: Address setup wait period
TA: Address strobe period
T2: Read (write) strobe period
TDHW: Data hold wait period

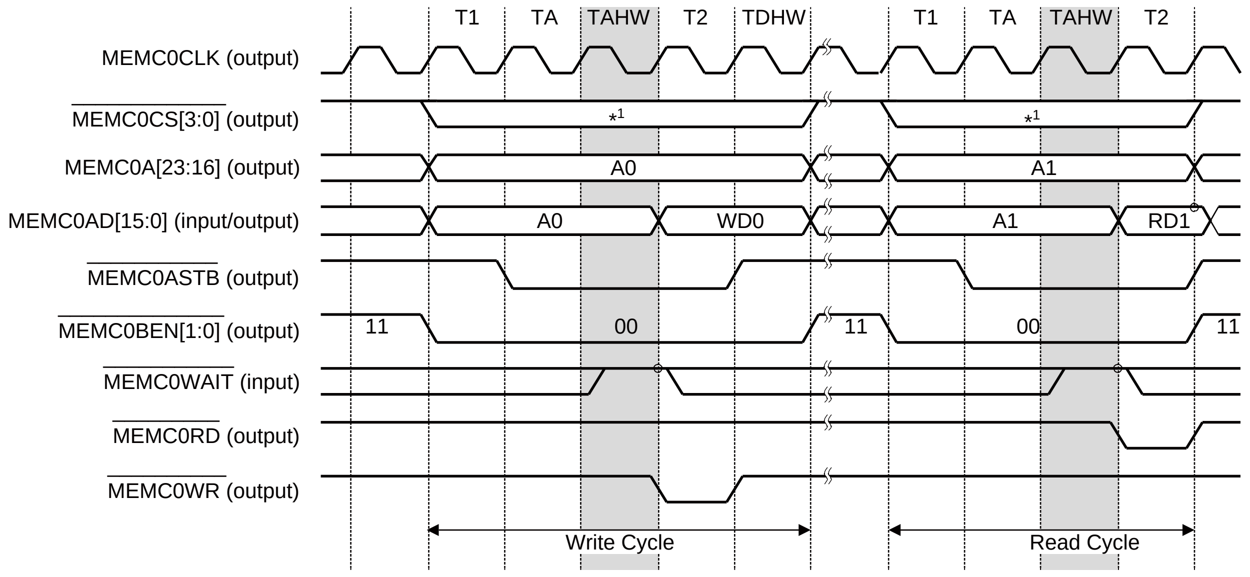
Note 1. Indicates the chip select area to be accessed.

Figure 16.10 Bus Cycles when Inserting Address Setup Waits

16.4.2.7 Address Hold Wait Function

The address hold wait function inserts waits after the address transfer state to secure hold time for address strobe.

Up to 3 wait cycles can be inserted for each chip select area by setting the AWC register.



Note: T_1: Address setup period
T_A: Address strobe period
T_{AHW}: Address hold wait period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

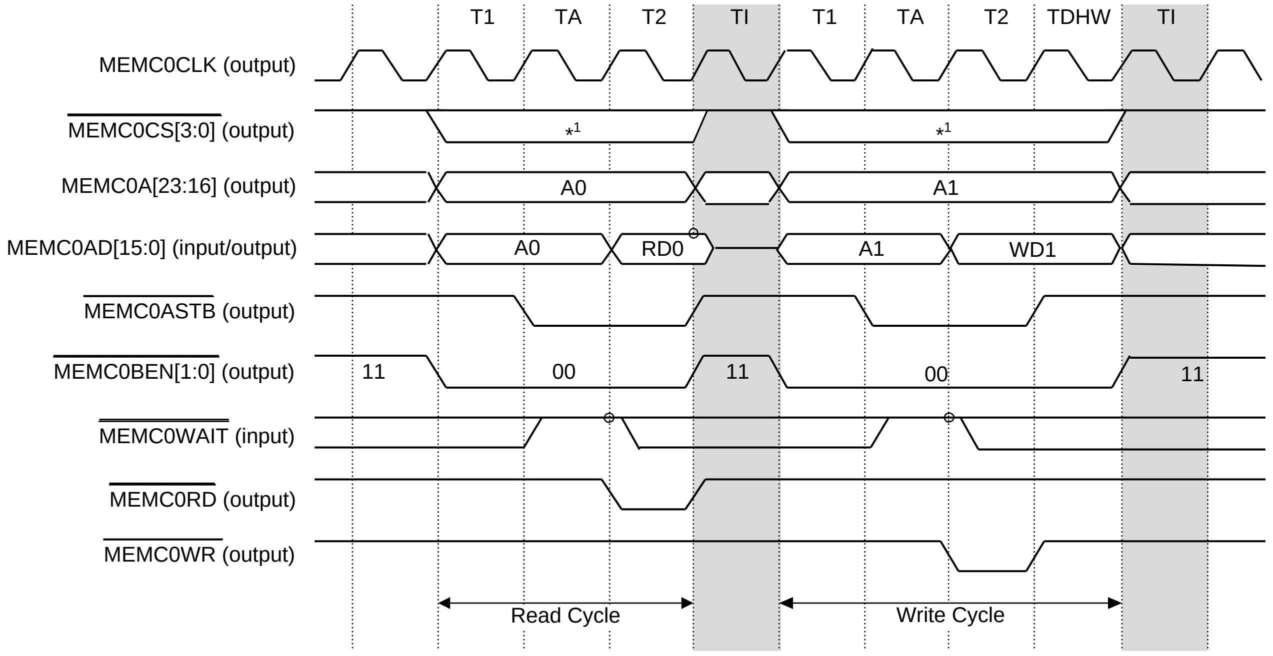
Note 1. Indicates the chip select area to be accessed.

Figure 16.11 Bus Cycles when Inserting Address Hold Waits

16.4.2.8 Idle Insertion Function

This function inserts an idle state after the last state of each cycle in order to prevent bus conflicts between cycles.

This function can be set independently after a read cycle/write cycle for each chip select area by setting the ICC register.



Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_I: Idle state period
T_{DHW}: Data hold wait period

Note 1. Indicates the chip select area to be accessed.

Figure 16.12 Bus Cycles when Inserting Idle States

16.4.3 Memory Connection Example

16.4.3.1 SRAM Connection Example

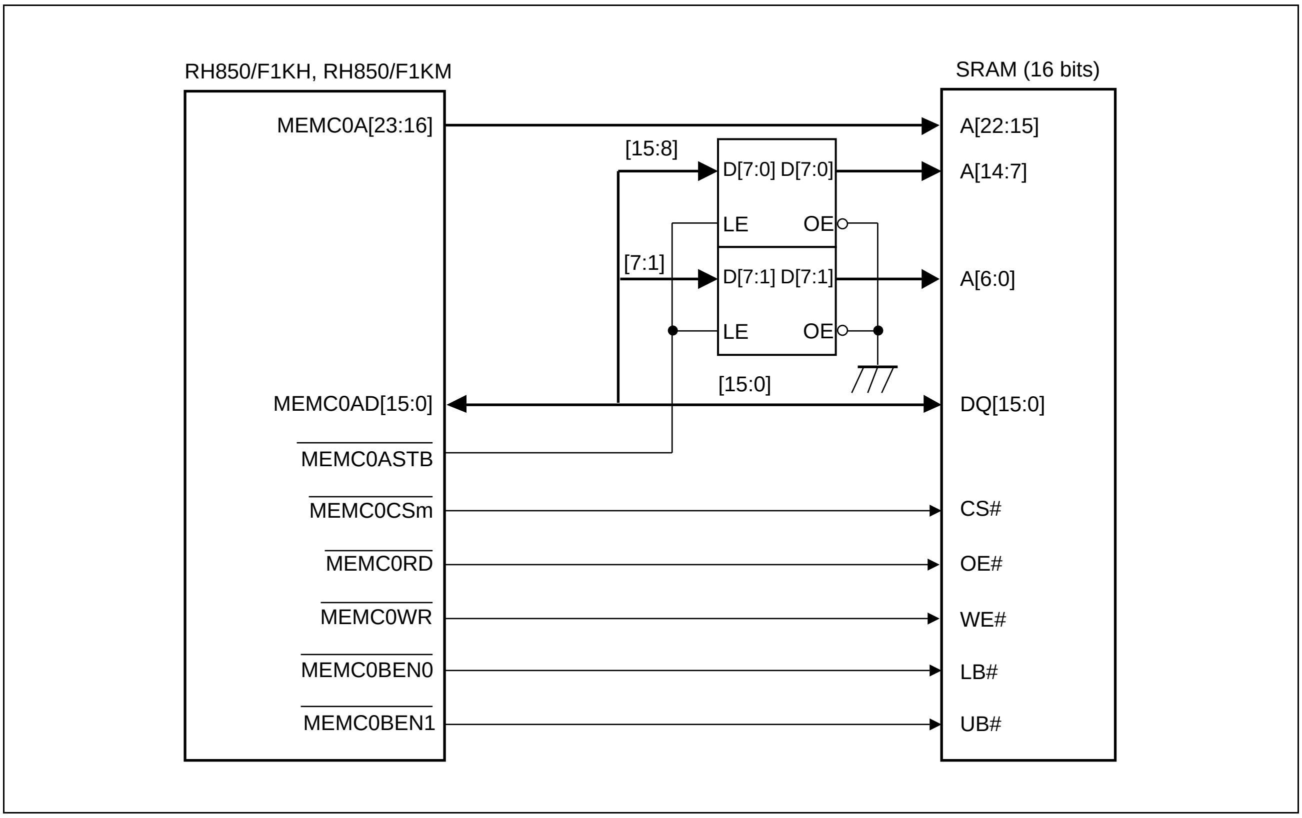


Figure 16.13 SRAM (16 Bits) Connection Example

16.4.4 Data Access Flow

The data transfer flow to the external memory differs according to the data width, specified endian, external bus width, and start address.

For misaligned access, the CPU performs the division and coupling of data. Whether the access is misaligned or not depends on the data width and start address.

Misaligned

As the accesses listed below are misaligned, the CPU divides the cycles.

- When half-word (16-bit) data is read from/written to an odd address
- When word (32-bit) data is read from/written to an address that is not a multiple of 4

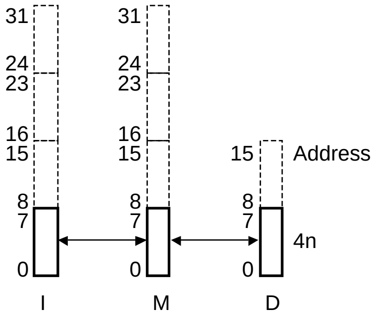
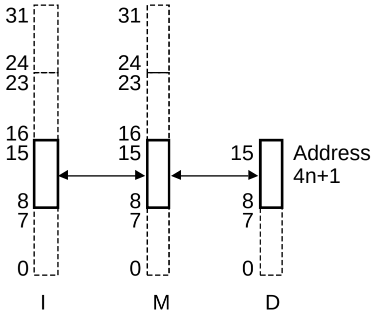
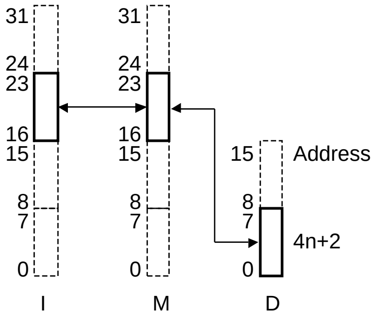
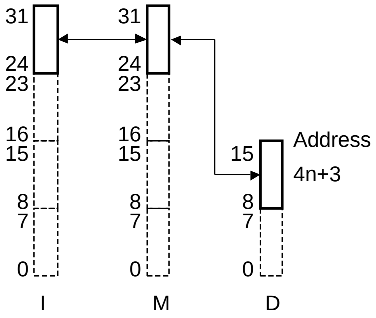
Table 16.32 Misaligned Access Conditions

Access Conditions		Cycles Divided by the CPU		
Data Width	Address	1st	2nd	3rd
16 bits	2n + 1	8-bit access to 2n + 1	8-bit access to 2n + 2	—
32 bits	4n + 1	8-bit access to 4n + 1	16-bit access to 4n + 2	8-bit access to 4n + 4
32 bits	4n + 2	16-bit access to 4n + 2	16-bit access to 4n + 4	—
32 bits	4n + 3	8-bit access to 4n + 3	16-bit access to 4n + 4	8-bit access to 4n + 6

The data flows for each condition are shown on the following pages.

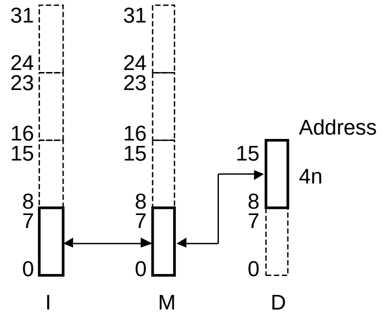
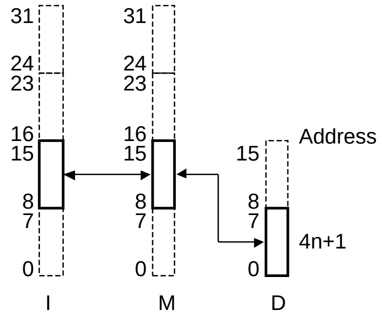
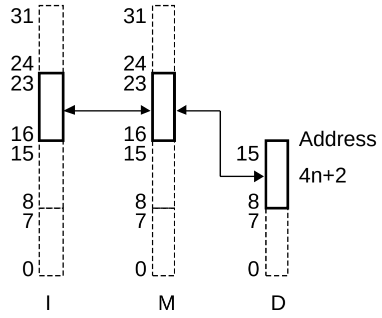
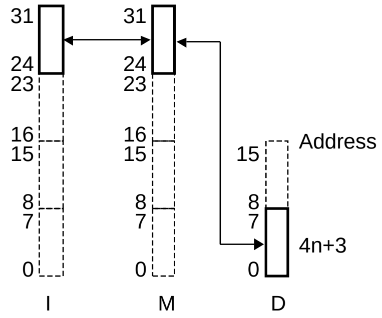
16.4.4.1 Data Flow for Byte Access (for Reading and Writing)

Table 16.33 Data Flow for Byte Access (Little Endian)

Address to be Accessed	Data Transfer Flow
4n	
4n + 1	
4n + 2	
4n + 3	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

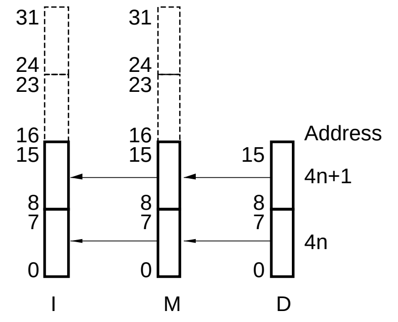
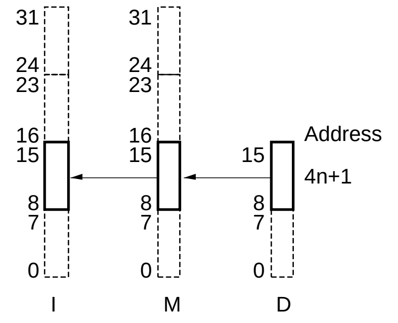
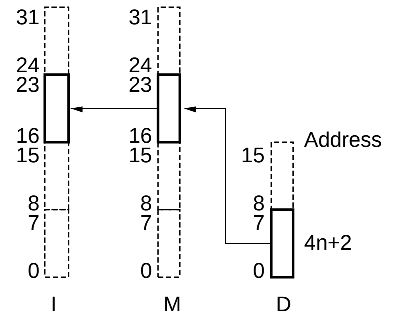
Table 16.34 Data Flow for Byte Access (Big Endian)

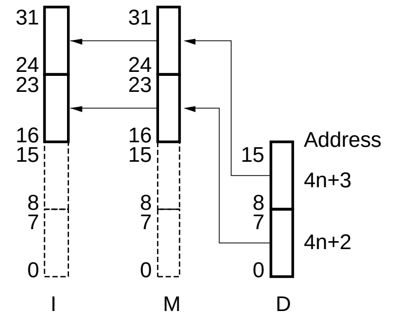
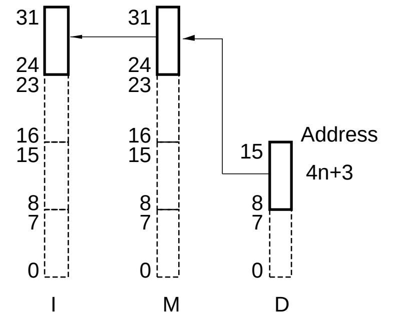
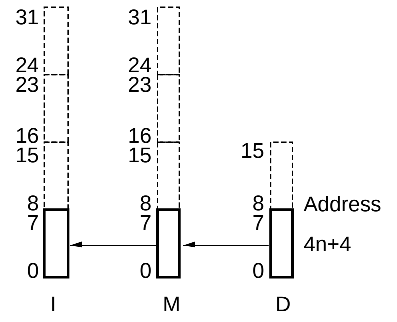
Address to be Accessed	Data Transfer Flow
4n	
4n + 1	
4n + 2	
4n + 3	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

16.4.4.2 Data Flow for Half-Word Read Access

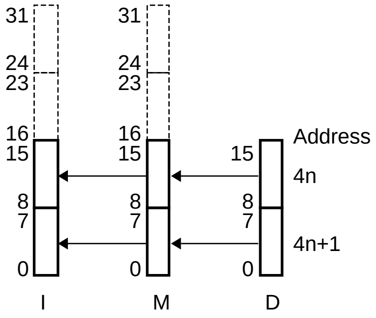
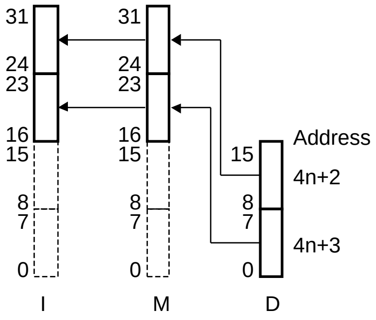
Table 16.35 Data Flow for Half-Word Read Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
4n + 1	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n + 2	1st	
4n + 3	1st	
	2nd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 16.36 Data Flow for Half-Word Read Access (Big Endian)

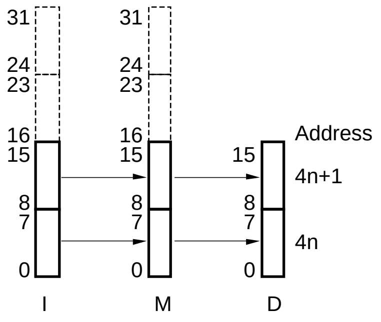
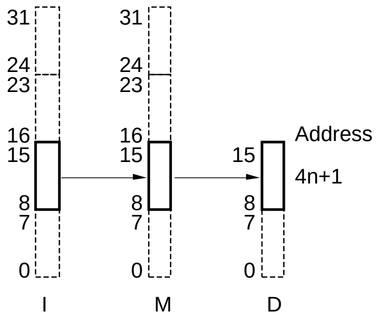
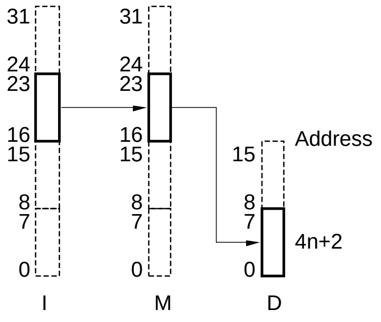
Address to be Accessed	Data Transfer Flow
4n	
4n + 2	

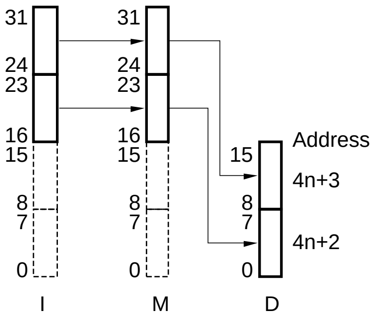
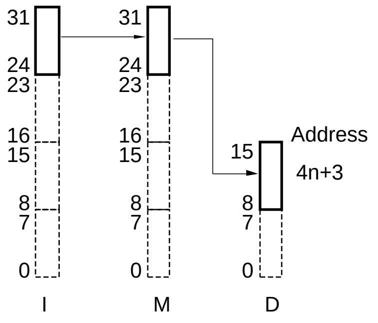
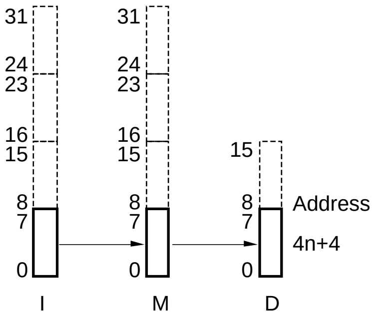
Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1 or 4n + 3 is prohibited.

16.4.4.3 Data Flow for Half-Word Write Access

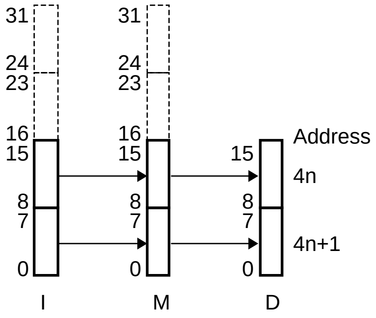
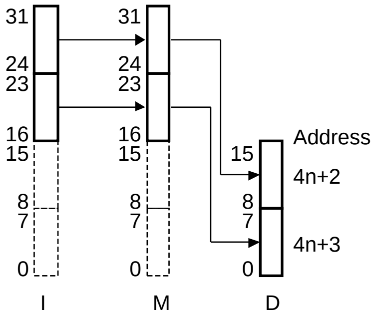
Table 16.37 Data Flow for Half-Word Write Access (Little Endian)

<table>
<tr><th colspan="2">Address to be Accessed</th><th>Data Transfer Flow</th></tr>
<tr><td>4n</td><td>1st</td><td>
</td></tr>
<tr><td>4n + 1</td><td>1st</td><td>
</td></tr>
<tr><td></td><td>2nd</td><td>
</td></tr>
</table>

<table>
<tr><th colspan="2">Address to be Accessed</th><th>Data Transfer Flow</th></tr>
<tr><td>4n + 2</td><td>1st</td><td>
</td></tr>
<tr><td>4n + 3</td><td>1st</td><td>
</td></tr>
<tr><td></td><td>2nd</td><td>
</td></tr>
</table>

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 16.38 Data Flow for Half-Word Write Access (Big Endian)

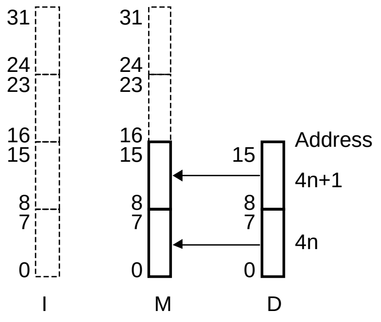
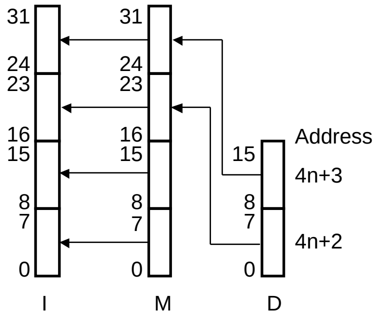
Address to be Accessed	Data Transfer Flow
4n	
4n + 2	

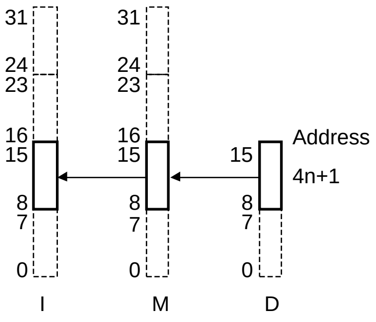
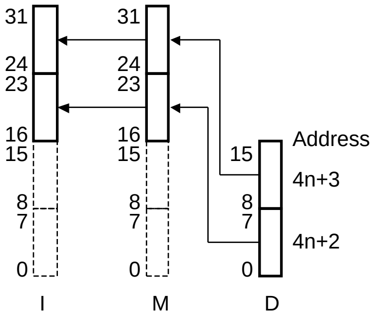
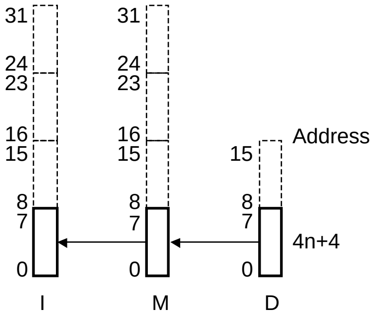
Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1 or 4n + 3 is prohibited.

16.4.4.4 Data Flow for Word Read Access

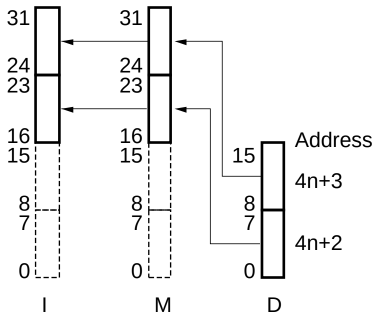
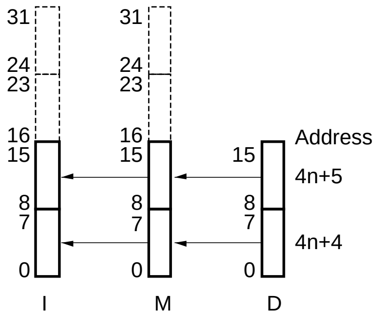
Table 16.39 Data Flow for Word Read Access (Little Endian)

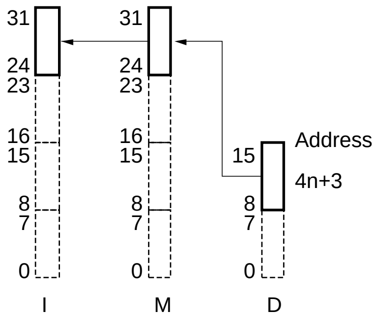
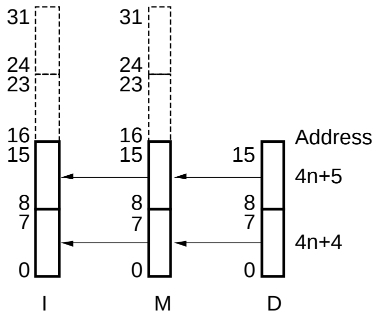
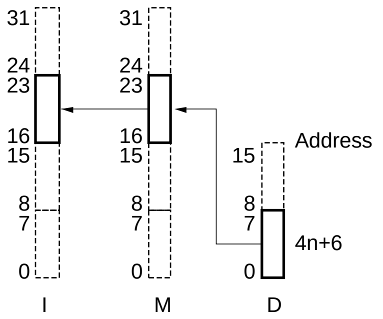
Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n + 1	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

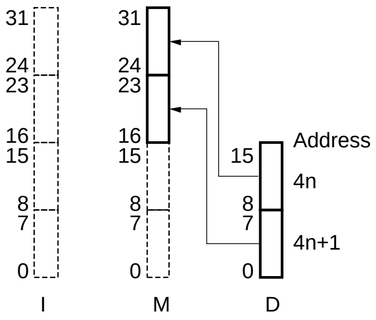
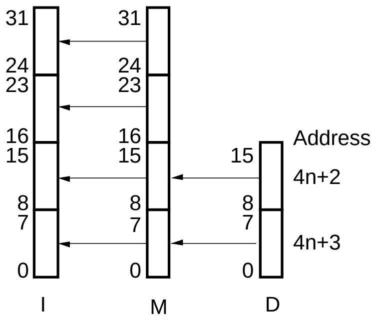
Table 16.39 Data Flow for Word Read Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n + 2	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n + 3	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 16.40 Data Flow for Word Read Access (Big Endian)

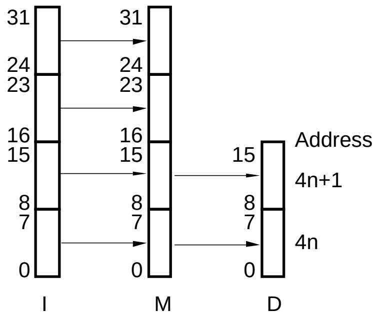
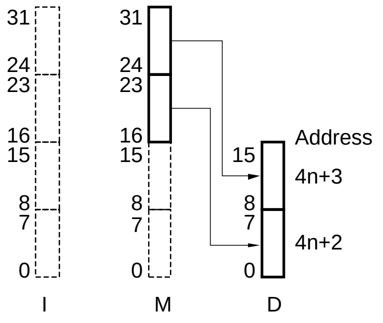
Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

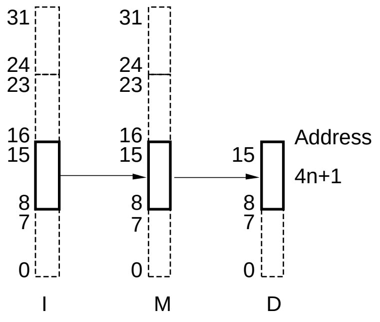
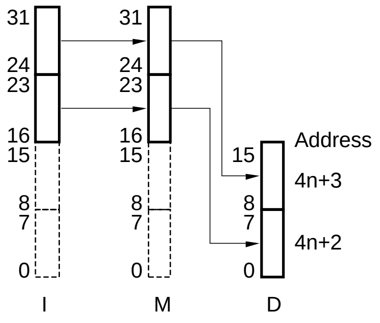
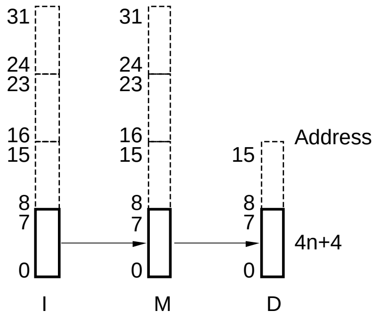
Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

16.4.4.5 Data Flow for Word Write Access

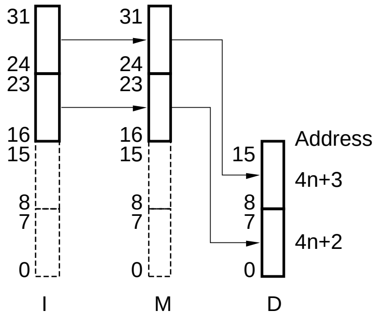
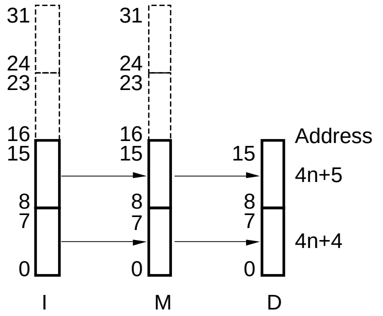
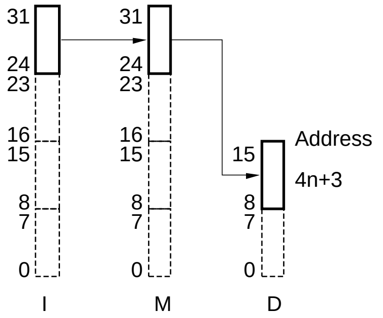
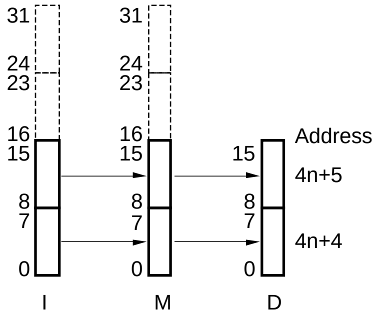
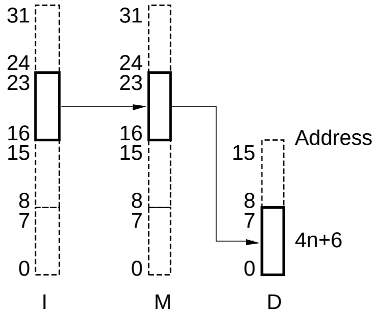
Table 16.41 Data Flow for Word Write Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n + 1	1st	
	2nd	
	3rd	

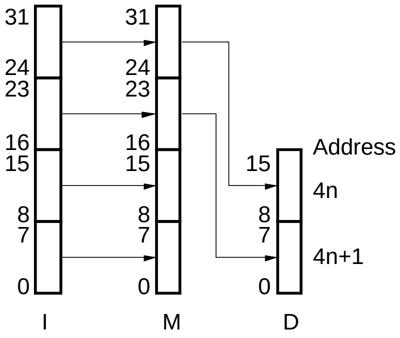
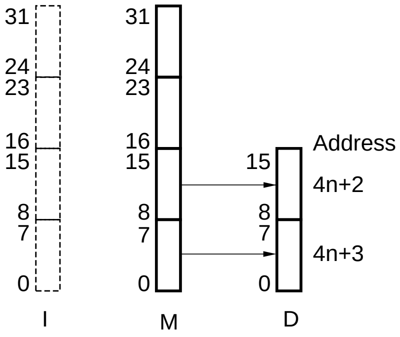
Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 16.41 Data Flow for Word Write Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n + 2	1st	
	2nd	
4n + 3	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 16.42 Data Flow for Word Write Access (Big Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

16.5 Notes on Use of MEMC

Stop MEMC before transitioning to standby mode.

Section 17 Serial Flash Memory Interface A (SFMA)

This section contains a generic description of the Serial Flash Memory Interface A (SFMA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of SFMA.

17.1 Features of RH850/F1KH, RH850/F1KM SFMA

17.1.1 Number of Units

This microcontroller has the following number of SFMA units.

Table 17.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	SFMAn (n = 0)	SFMAn (n = 0)	SFMAn (n = 0)

Table 17.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	—	1	1	1	1
Name	—	SFMAn (n = 0)	SFMAn (n = 0)	SFMAn (n = 0)	SFMAn (n = 0)

Table 17.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	—	—
Name	—	—	—	—

Table 17.4 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual SFMA units are identified by the index “n”; for example, SFMAnCMNCR (n = 0) is the SFMAn common control register.

Table 17.5 Index (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual SFMA units are identified by the index “n”; for example, SFMAnCMNCR (n = 0) is the SFMAn common control register.

17.1.2 Register Base Address

SFMAn base address is listed in the following table.

SFMAn register addresses are given as an offset from the base address in general.

Table 17.6 Register Base Address (RH850/F1KH-D8)

Base Address Name	Base Address
<SFMA0_base>	1004 0000_H

Table 17.7 Register Base Address (RH850/F1KM-S4)

Base Address Name	Base Address
<SFMA0_base>	1004 0000_H

17.1.3 Clock Supply

The SFMAn clock supply is shown in the following table.

Table 17.8 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name	Description
SFMAn	Bϕ	CKSCLK_IPERI1	SFMA clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	SFMA clock

Table 17.9 Clock Supply (RH850/F1KM-S4)

Unit Name	Unit Clock Name	Supply Clock Name	Description
SFMAn	Bϕ	CKSCLK_IPERI1	SFMA clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	SFMA clock

17.1.4 Reset Sources

SFMAn reset sources are listed in the following table. SFMAn is initialized by these reset sources.

Table 17.10 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
SFMAn	All reset sources (ISORES)

Table 17.11 Reset Sources (RH850/F1KM-S4)

Unit Name	Reset Source
SFMAn	All reset sources (ISORES)

17.1.5 External Input/Output Signals

External input/output signals of SFMAn are listed below.

Table 17.12 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
SFMA0		
SPBCLK	Clock output	SFMA0CLK
SPBSSL	Slave select signal output	SFMA0SSL
SPBMO/SPBIO0	Master transmit data/data 0	SFMA0IO0
SPBMI/SPBIO1	Master input data/data 1	SFMA0IO1
SPBIO2	Data 2	SFMA0IO2
SPBIO3	Data 3	SFMA0IO3

Table 17.13 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
SFMA0		
SPBCLK	Clock output	SFMA0CLK
SPBSSL	Slave select signal output	SFMA0SSL
SPBMO/SPBIO0	Master transmit data/data 0	SFMA0IO0
SPBMI/SPBIO1	Master input data/data 1	SFMA0IO1
SPBIO2	Data 2	SFMA0IO2
SPBIO3	Data 3	SFMA0IO3

17.2 Overview

17.2.1 Functional Overview

The Serial Flash Memory Interface outputs control signals to the serial flash memory connected to the SPI multi I/O bus space, thus enabling direct connection of the serial flash memory.

This module allows the connected serial flash memory to be accessed by directly reading the SPI multi I/O bus space, or using SPI operating mode to transmit and receive data.

- Serial Flash Memory Interface

  One serial flash memory device can be connected.

  A data bus size of 1 bit, 2 bits, or 4 bits can be selected.

- External Address Space Read Mode

  A maximum of 4-Gbyte address space is supported

  The SPBSSL pin can be automatically controlled through access address monitoring

  Efficient data reception due to built-in read cache (64-bit line × 16 entries)

- SPI Operating Mode

  Desired read/write access to serial flash memory possible

- Bit rate

  SPBCLK is generated by frequency division of Bϕ by internal baud rate generator

  SPBCLK frequency division ratio can be set from 2 to 4080

- SPBSSL Pin Control

  Delay from SPBSSL signal assertion to SPBCLK operation (clock delay) can be set
    Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)

  Delay from SPBCLK stop to SPBSSL output negation (SPBSSL negation delay) can be set
    Range: 1.5 to 8.5 SPBCLK cycles (set in SPBCLK-cycle units)

  SPBSSL output assertion wait before next access (next access delay) can be set
    Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)

  SPBSSL polarity can be changed

17.2.2 Block Diagram

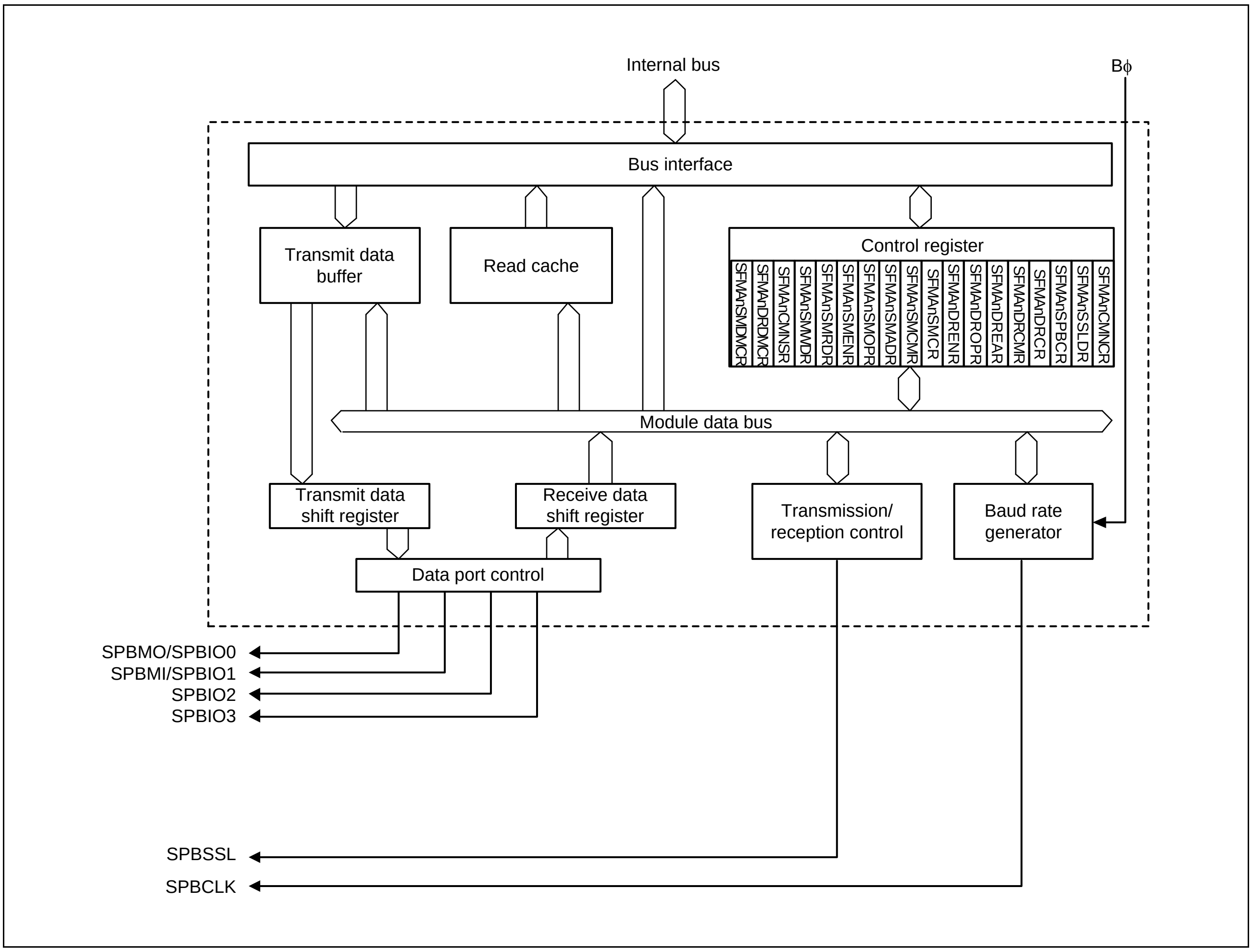


Figure 17.1 SFMA Block Diagram

17.3 Registers

17.3.1 List of Registers

SFMA registers are listed in the following table.

For details about <SFMAn_base>, see **Section 17.1.2, Register Base Address**.

Table 17.14 List of Registers

Module Name	Register Name	Symbol	Address
SFMAn	SFMAn common control register	SFMAnCMNCR	<SFMAn_base> + 00_H
	SFMAn SSL delay register	SFMAnSSLDR	<SFMAn_base> + 04_H
	SFMAn bit rate register	SFMAnSPBCR	<SFMAn_base> + 08_H
	SFMAn data read control register	SFMAnDRCR	<SFMAn_base> + $0C_H$
	SFMAn data read command setting register	SFMAnDRCMR	<SFMAn_base> + 10_H
	SFMAn data read extended address setting register	SFMAnDREAR	<SFMAn_base> + 14_H
	SFMAn data read option setting register	SFMAnDROPR	<SFMAn_base> + 18_H
	SFMAn data read enable setting register	SFMAnDRENR	<SFMAn_base> + $1C_H$
	SFMAn SPI mode control register	SFMAnSMCR	<SFMAn_base> + 20_H
	SFMAn SPI mode command setting register	SFMAnSMCMR	<SFMAn_base> + 24_H
	SFMAn SPI mode address setting register	SFMAnSMADR	<SFMAn_base> + 28_H
	SFMAn SPI mode option setting register	SFMAnSMOPR	<SFMAn_base> + $2C_H$
	SFMAn SPI mode enable setting register	SFMAnSMENR	<SFMAn_base> + 30_H
	SFMAn SPI mode read data register	SFMAnSMRDR	<SFMAn_base> + 38_H
	SFMAn SPI mode write data register	SFMAnSMWDR	<SFMAn_base> + 40_H
	SFMAn common status register	SFMAnCMNSR	<SFMAn_base> + 48_H
	SFMAn data read dummy cycle setting register	SFMAnDRDMCR	<SFMAn_base> + 58_H
	SFMAn SPI mode dummy cycle setting register	SFMAnSMDMCR	<SFMAn_base> + 60_H

17.3.2 SFMAnCMNCR — SFMAn Common Control Register

SFMAnCMNCR is a 32-bit register that controls the SPI multi I/O bus controller. The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 00$_H$

Value after reset: 01AA 4000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD	—	—	—	—	—	—	—	MOIIO3[1:0]		MOIIO2[1:0]		MOIIO1[1:0]		MOIIO0[1:0]	
Value after reset	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO3FV[1:0]		IO2FV[1:0]		—	—	IO0FV[1:0]		—	CPHAT	CPHAR	SSLP	CPOL	—	—	—
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R

Table 17.15 SFMAnCMNCR Register Contents

Bit Position	Bit Name	Function
31	MD	Operating Mode Switch Switches the operating modes. 0: External address space read mode 1: SPI operating mode
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23, 22	MOIIO3[1:0]	SPBSSL Output Idle Value Fix SPBIO3 Fixes the output value of SPBIO3 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
21, 20	MOIIO2[1:0]	SPBSSL Output Idle Value Fix SPBIO2 Fixes the output value of SPBIO2 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
19, 18	MOIIO1[1:0]	SPBSSL Output Idle Value Fix SPBIO1 Fixes the output value of SPBIO1 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z

Table 17.15 SFMAnCMNCR Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17, 16</td><td>MOIIO0[1:0]</td><td>SPBSSL Output Idle Value Fix SPBIO0
Fixes the output value of SPBIO0 in SPBSSL negation period.
00: Output value 0
01: Output value 1
10: Output value is kept at a preceding value.
11: Output value Hi-Z</td></tr>
<tr><td>15, 14</td><td>IO3FV[1:0]</td><td>SPBIO3 Fixed Value for 1-bit/2-bit Size
Fixes the output value of SPBIO3 pin for 1-bit/2-bit size.
00: Output value 0
01: Output value 1
10: Output value is kept at a preceding value.
11: Output value Hi-Z</td></tr>
<tr><td>13, 12</td><td>IO2FV[1:0]</td><td>SPBIO2 Fixed Value for 1-bit/2-bit Size
Fixes the output value of SPBIO2 pin for 1-bit/2-bit size.
00: Output value 0
01: Output value 1
10: Output value is kept at a preceding value.
11: Output value Hi-Z</td></tr>
<tr><td>11, 10</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>9, 8</td><td>IO0FV[1:0]</td><td>SPBIO0 Fixed Value for 1-bit Size Input
Fixes the output value of SPBIO0 pin for 1-bit size input.
00: Output value 0
01: Output value 1
10: Output value is kept at a preceding value.
11: Output value Hi-Z</td></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>6</td><td>CPHAT</td><td>Output Shift
Sets the SPBCLK edge of the output data.
CPHAT and CPHAR should be set according to the table in the description of CPHAR.
0: Data transmission at even edge
1: Data transmission at odd edge</td></tr>
<tr><td>5</td><td>CPHAR</td><td>Input Latch
Sets the SPBCLK edge of the reception data.
CPHAT and CPHAR should be set according to the following table.
0: Data reception at odd edge
1: Data reception at even edge
<table>
<tr><th>CPHAT</th><th>CPHAR</th><th></th></tr>
<tr><td>0</td><td>0</td><td>Setting enabled</td></tr>
<tr><td>0</td><td>1</td><td>Setting enabled</td></tr>
<tr><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Setting enabled</td></tr>
</table></td></tr>
<tr><td>4</td><td>SSLP</td><td>SPBSSL Signal Polarity
Sets the polarity of SPBSSL signal.
0: Active low SPBSSL signal
1: Active high SPBSSL signal</td></tr>
<tr><td>3</td><td>CPOL</td><td>SPBSSL Negation Period SPBCLK Output Direction
Sets the SPBCLK output direction during SPBSSL negation period.
0: SPBCLK output is 0 during SPBSSL negation period.
1: SPBCLK output is 1 during SPBSSL negation period.</td></tr>
<tr><td>2 to 0</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

17.3.3 SFMAnSSLDR — SFMAn SSL Delay Register

SFMAnSSLDR is a 32-bit register that adjusts the timing between the SPBSSL signal and the SPBCLK signal.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 04_H

Value after reset: $0007\ 0707_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Value after reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 17.16 SFMAnSSLDR Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	SPNDL[2:0]	Next Access Delay Sets the period from transfer end to next transfer start (next access). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	SLNDL[2:0]	SPBSSL Negation Delay Sets the period from the time the last SPBCLK edge is sent of a transfer to SPBSSL pin negation (SPBSSL negation delay). 000: 1.5 SPBCLK cycles 001: 2.5 SPBCLK cycles 010: 3.5 SPBCLK cycles 011: 4.5 SPBCLK cycles 100: 5.5 SPBCLK cycles 101: 6.5 SPBCLK cycles 110: 7.5 SPBCLK cycles 111: 8.5 SPBCLK cycles
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.16 SFMAnSSLDR Register Contents

Bit Position	Bit Name	Function
2 to 0	SCKDL[2:0]	Clock Delay Sets the period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles

17.3.4 SFMAnSPBCR — SFMAn Bit Rate Register

SFMAnSPBCR is a 32-bit register that sets the bit rate.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 08_H

Value after reset: $0000\ 0003_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPBR[7:0]								—	—	—	—	—	—	BRDV[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 17.17 SFMAnSPBCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	SPBR[7:0]	Bit Rate Sets the bit rate. The bit rate is determined by a combination of these bits with the BRDV[1:0] bits. Setting SPBR[7:0] to 0 is prohibited. SPBR[7:0] needs to be changed to non-zero value before SPI communication starts.
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	BRDV[1:0]	Bit Rate Frequency Division Sets the bit rate. The bit rate is determined by a combination of these bits with the SPBR[7:0] bits. The SPBR value is used to set the base bit rate. The BRDV value is used to select a division ratio of the base bit rate from among no division, 2, 4, and 8. 00: Base bit rate 01: Base bit rate divided by 2 10: Base bit rate divided by 4 11: Base bit rate divided by 8

Bit Rate

SPBR[7:0] and BRDV[1:0] are used for setting the bit rate.

The following formula is used to calculate the bit rate when SPBR[7:0] $\neq 0$.

$$\text{Bit rate} = B\phi / (2 \times n \times 2^N)$$

n: SPBR[7:0] setting (1, …, 255)

N: BRDV[1:0] setting (0 to 3)

17.3.5 SFMAnDRCR — SFMAn Data Read Control Register

SFMAnDRCR is a 32-bit register that sets the operation in external address space read mode.

The bits except the SSLN bit should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 0C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SSLN	—	—	—	—	RBURST[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RCF	RBE	—	—	—	—	—	—	—	SSLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R/W	R	R	R	R	R	R	R	R/W

Table 17.18 SFMAnDRCR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	SSLN	SPBSSL Negation Asserted SPBSSL can be negated by writing 1 to this bit when both the RBE and SSLE bits are 1. This bit is always read as 0. **NOTE:** To start next access after SPBSSL negation using this bit, read SSLF in SFMAnCMNSR = 0 to confirm that the SPBSSL has been negated.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	RBURST[3:0]	Read Data Burst Length Sets the burst length (data unit count) when reading. This bit is enabled when the RBE bit is set to 1. 0000: 1 data unit 0001: 2 continuous data units : 1110: 15 continuous data units 1111: 16 continuous data units One data unit is 64 bits long.
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	RCF	Read cache Flush When 1 is written to this bit, all the entries in the read cache are cleared. This bit is always read as 0. **NOTE:** After flushing the read cache by writing 1 to the RCF bit, read the SFMAnDRCR before proceeding to read from the external address space.
8	RBE	Read Burst Turns burst ON or OFF when reading. 0: Data is read according to the access size. 1: Read cache is enabled, and as many data units as the burst count specified in RBURST[3:0] bits is read.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.18 SFMAnDRCR Register Contents

Bit Position	Bit Name	Function
0	SSLE	SPBSSL Negation Sets the conditions for SPBSSL negation during read burst. SPBSSL is negated for each access during normal read. 0: SPBSSL is negated after transfer of data set in burst length. 1: SPBSSL is negated when the accessed address is not continuous with the previously transferred address.

17.3.6 SFMAnDRCMR — SFMAn Data Read Command Setting Register

SFMAnDRCMR is a 32-bit register that sets the commands issued in external address space read mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access:	This register can be read or written in 32-bit units.
Address:	<SFMAn_base> + 10_H
Value after reset:	0003 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.19 SFMAnDRCMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

17.3.7 SFMAnDREAR — SFMAn Data Read Extended Address Setting Register

SFMAnDREAR is a 32-bit register that sets the address when the serial flash address is output in 32-bit mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 14_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	EAV[5:0]						—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.20 SFMAnDREAR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 17	EAV[5:0]	32-Bit Extended Upper Address Fixed Value Sets the serial flash address [31:26] fixed values to EAV[5:0]. This setting is valid when the ADE[3] bit in SFMAnDRENR is 1.
16 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

17.3.8 SFMAnDROPR — SFMAn Data Read Option Setting Register

SFMAnDROPR is a 32-bit register that sets the option data in external address space read mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.21 SFMAnDROPR Register Contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.3.9 SFMAnDRENR — SFMAn Data Read Enable Setting Register

SFMAnDRENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and read data in external address space read mode and enables outputting them other than read data.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 1C$_H$

Value after reset: 0000 4700$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		—	—	ADB[1:0]		—	—	OPDB[1:0]		—	—	DRDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	ADE[3:0]				OPDE[3:0]				—	—	—	—
Value after reset	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 17.22 SFMAnDRENR Register Contents

Bit Position	Bit Name	Function
31, 30	CDB[1:0]	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	ADB[1:0]	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	OPDB[1:0]	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.22 SFMAnDRENR Register Contents

Bit Position	Bit Name	Function
17, 16	DRDB[1:0]	Data Read Bit Size Sets the data read size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	DME	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. **NOTE:** A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	Address Enable Sets the address to be output. Be sure to use the following setting; otherwise, the operation is not guaranteed. 0000: Output disabled 0111: Address[23:0] 1111: Address[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

17.3.10 SFMAnSMCR — SFMAn SPI Mode Control Register

SFMAnSMCR is a 32-bit register that sets the operation in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access:	This register can be read or written in 32-bit units.
Address:	<SFMAn_base> + 20_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SSLKP	—	—	—	—	—	SPIRE	SPIWE	SPIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	W

Table 17.23 SFMAnSMCR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SSLKP	SPBSSL Signal Level Determines the SPBSSL status after the end of transfer. 0: SPBSSL signal is negated at the end of transfer. 1: SPBSSL signal level is maintained from the end of transfer to the start of next access.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	SPIRE	Data Read Enable Enables reading in SPI operating mode. 0: Data reading disabled 1: Data reading enabled **NOTE:** When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
1	SPIWE	Data Write Enable Enables writing in SPI operating mode. 0: Data writing disabled 1: Data writing enabled **NOTE:** When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
0	SPIE	SPI Data Transfer Enable Data is transferred by setting this bit to 1. This bit is enabled only when the TEND bit in SFMAnCMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0. This bit is always read as 0. **NOTE:** When the SPBSSL pin is de-asserted, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0. When the SPBSSL pin is asserted, follow the notes described in **Section 17.5.2, Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode**.

17.3.11 SFMAnSMCMR — SFMAn SPI Mode Command Setting Register

SFMAnSMCMR is a 32-bit register that sets the commands issued in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 24_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.24 SFMAnSMCMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

17.3.12 SFMAnSMADR — SFMAn SPI Mode Address Setting Register

SFMAnSMADR is a 32-bit register that sets the addresses in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 28_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:24]								ADR[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.25 SFMAnSMADR Register Contents

Bit Position	Bit Name	Function
31 to 24	ADR[31:24]	Address Sets the value of bits 31 to 24 when the serial flash address is output in 32-bit units. This setting is valid when ADE[3] in SFMAnSMENR is 1.
23 to 0	ADR[23:0]	Address Sets the address.

17.3.13 SFMAnSMOPR — SFMAn SPI Mode Option Setting Register

SFMAnSMOPR is a 32-bit register that sets the option data in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + $2C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.26 SFMAnSMOPR Register Contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.3.14 SFMAnSMENR — SFMAn SPI Mode Enable Setting Register

SFMAnSMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in SPI operating mode and enables their output. SFMAnSMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 30_H

Value after reset: $0000\ 4000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		—	—	ADB[1:0]		—	—	OPDB[1:0]		—	—	SPIDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	ADE[3:0]				OPDE[3:0]				SPIDE[3:0]			
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.27 SFMAnSMENR Register Contents

Bit Position	Bit Name	Function
31, 30	CDB[1:0]	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	ADB[1:0]	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	OPDB[1:0]	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited

Table 17.27 SFMAnSMENR Register Contents

Bit Position	Bit Name	Function
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	SPIDB[1:0]	Transfer Data Bit Size Sets the transfer data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	DME	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. **NOTE:** Dummy cycle insertion is prohibited for write in SPI operating mode including the case in which a transfer ends with a dummy cycle. **NOTE:** A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	Address Enable Sets the address to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 0100: ADR[23:16] 0110: ADR[23:8] 0111: ADR[23:0] 1111: ADR[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	SPIDE[3:0]	Transfer Data Enable Sets valid transfer data. The following settings must be used. Otherwise, the operation is not guaranteed. 0000: Not transferred 1000: 8 bits transferred (enables data at address 0 of the SPI mode read/write data register) 1100: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data register) 1111: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data register) Other than above: Setting prohibited

17.3.15 SFMAnSMRDR — SFMAn SPI Mode Read Data Register

SFMAnSMRDR is a 32-bit register that stores the read data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SFMAnSMENR). Be sure to access from address 0.

The settings of this register should be read when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: SFMAnSMRDR is a read-only register that can be read in 32-bit units.

SFMAnSMRDRL, SFMAnSMRDRH are read-only registers that can be read in 16-bit units.

SFMAnSMRDRLL, SFMAnSMRDRLH, SFMAnSMRDRHL, SFMAnSMRDRHH are read-only registers that can be read in 8-bit units.

Address: SFMAnSMRDR: <SFMAn_base> + 38_H

SFMAnSMRDRL: <SFMAn_base> + 38_H

SFMAnSMRDRLL: <SFMAn_base> + 38_H

SFMAnSMRDRLH: <SFMAn_base> + 39_H

SFMAnSMRDRH: <SFMAn_base> + $3A_H$

SFMAnSMRDRHL: <SFMAn_base> + $3A_H$

SFMAnSMRDRHH: <SFMAn_base> + $3B_H$

Value after reset: XXXX $XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDATA[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDATA[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.28 SFMAnSMRDR Register Contents

Bit Position	Bit Name	Function
31 to 0	RDATA[31:0]	Read Data Holds the data read in SPI operating mode.

Note: The contents of this register are modified upon completion of reception in SPI operating mode. Be sure to read data when reception in SPI operating mode is completed.

17.3.16 SFMAnSMWDR — SFMAn SPI Mode Write Data Register

SFMAnSMWDR is a 32-bit register that sets the write data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SFMAnSMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access:	SFMAnSMWDR can be read or written in 32-bit units. SFMAnSMWDRL, SFMAnSMWDRH can be read or written in 16-bit units. SFMAnSMWDRLL, SFMAnSMWDRLH, SFMAnSMWDRHL, SFMAnSMWDRHH can be read or written in 8-bit units.
Address:	SFMAnSMWDR: <SFMAn_base> + 40_H SFMAnSMWDRL: <SFMAn_base> + 40_H SFMAnSMWDRLL: <SFMAn_base> + 40_H SFMAnSMWDRLH: <SFMAn_base> + 41_H SFMAnSMWDRH: <SFMAn_base> + 42_H SFMAnSMWDRHL: <SFMAn_base> + 42_H SFMAnSMWDRHH: <SFMAn_base> + 43_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.29 SFMAnSMWDR Register Contents

Bit Position	Bit Name	Function
31 to 0	WDATA[31:0]	Write Data Holds the data to be written in SPI operating mode.

17.3.17 SFMAnCMNSR — SFMAn Common Status Register

SFMAnCMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<SFMAn_base> + 48_H
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSLF	TEND
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.30 SFMAnCMNSR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SSLF	SPBSSL Pin Monitor 0: SPBSSL pin is negated 1: SPBSSL pin is asserted
0	TEND	Transfer End Flag Indicates whether the data transfer has ended. 0: Indicates that data transfer is in progress 1: Indicates that data transfer has ended

17.3.18 SFMAnDRDMCR — SFMAn Data Read Dummy Cycle Setting Register

SFMAnDRDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in external address space read mode.

The settings of this register are enabled when the DME bit in the data read enable setting register (SFMAnDRENR) is 1.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access:	This register can be read or written in 32-bit units.
Address:	<SFMAn_base> + 58_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 17.31 SFMAnDRDMCR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DMDB[1:0]	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (SFMAnCMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMCYC[2:0]	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (SFMAnDRENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.3.19 SFMAnSMDMCR — SFMAn SPI Mode Dummy Cycle Setting Register

SFMAnSMDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in SPI operating mode.

The settings of this register are enabled when the DME bit in the SPI mode enable setting register (SFMAnSMENR) is 1.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access:	This register can be read or written in 32-bit units.
Address:	<SFMAn_base> + 60_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 17.32 SFMAnSMDMCR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DMDB[1:0]	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (SFMAnCMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMCYC[2:0]	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the SPI mode enable setting register (SFMAnSMENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.4 Operation

17.4.1 System Configuration

With this module, one serial flash memory can be directly connected (data size of 1, 2, and 4 bits).

The example of system configuration is shown in **Figure 17.2, System Configuration Example with 4-Bit Data Size**.

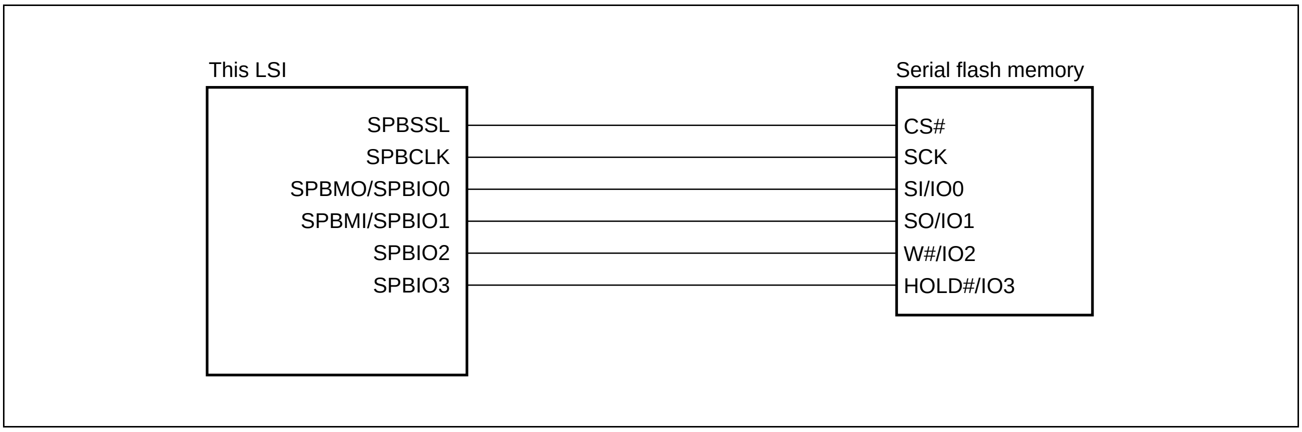


Figure 17.2 System Configuration Example with 4-Bit Data Size

17.4.2 Address Map

In external address space read mode, the serial flash connected is assigned in the SPI multi I/O bus space. In combination with DREAR, a maximum of 4 Gbytes can be accessed.

Table 17.33 Address Map

Internal Address	Max. Access Area
3000 0000$_H$ to 33FF FFFF$_H$	4 Gbytes

17.4.3 32-bit Serial Flash Addresses

Since the SPI multi I/O bus space is 64 Mbytes, only a part of the 32-bit serial flash address area can be directly accessed. Here, the fixed value set in the pertinent register is used as the upper bit value of a 32-bit address.

To output serial flash addresses in 32 bits, set the ADE[3] bit in SFMAnDRENR to 1, and set the serial flash address [31:26] fixed values to the EAV[5:0] bits in SFMAnDREAR.

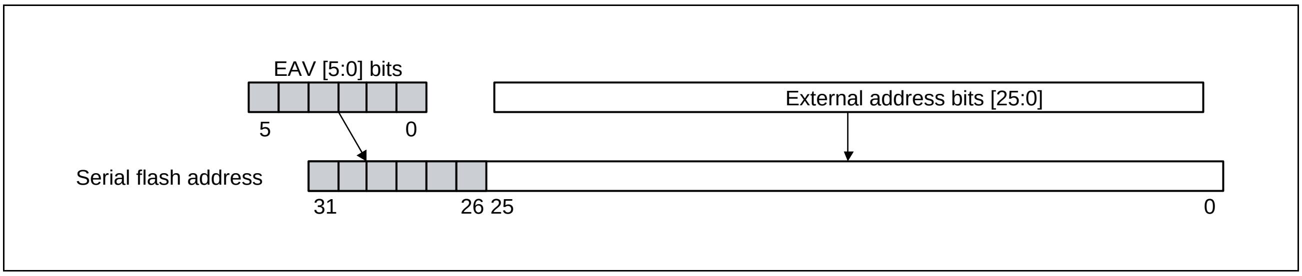


Figure 17.3 32-Bit Address Setting

17.4.4 Operating Modes

This module has two operating modes: external address space read mode and SPI operating mode.

In external address space read mode, a read access to the SPI multi I/O bus space is converted into SPI communication and data is received. After data acquisition, data is returned to the bus master that is the issuing source. For details, see **Section 17.4.5, External Address Space Read Mode**.

In SPI operating mode, arbitrary SPI communication is carried out using register settings. For details, see **Section 17.4.8, SPI Operating Mode**.

17.4.5 External Address Space Read Mode

A read access to the SPI multi I/O bus space can be converted into SPI communication in external address space read mode. Further, the commands, optional commands, option data, and dummy cycle issued for reading can be modified using registers.

In external address space read mode, either normal read operation or burst read operation can be selected. The transfer format is determined based on the common control register (SFMAnCMNCR), SSL delay register (SFMAnSSLDR), bit rate setting register (SFMAnSPBCR), data read control register (SFMAnDRCR), data read command setting register (SFMAnDRCMR), data read extended address setting register (SFMAnDREAR), data read option setting register (SFMAnDROPR), data read enable setting register (SFMAnDRENR), and data read dummy cycle setting register (SFMAnDRDMCR).

17.4.5.1 Normal Read Operation

When the RBE bit in SFMAnDRCR is set to 0, normal read operation is performed.

In the normal read operation, the data of 8 bits, 16 bits, and 32 bits are read for respectively a byte, a half-word, and a word read access. After reading, the SPBSSL pin is negated.

The normal read operation timing is shown in **Figure 17.4, Normal Read Operation Timing**.

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay), t2 is the time period from transmission of the last SPBCLK edge of a transfer to SPBSSL pin negation (SPBSSL negation delay), and t3 is the time period from one transfer end to the next transfer start (next access). For details of t1, t2, and t3, see **Section 17.4.9, Transfer Format**.

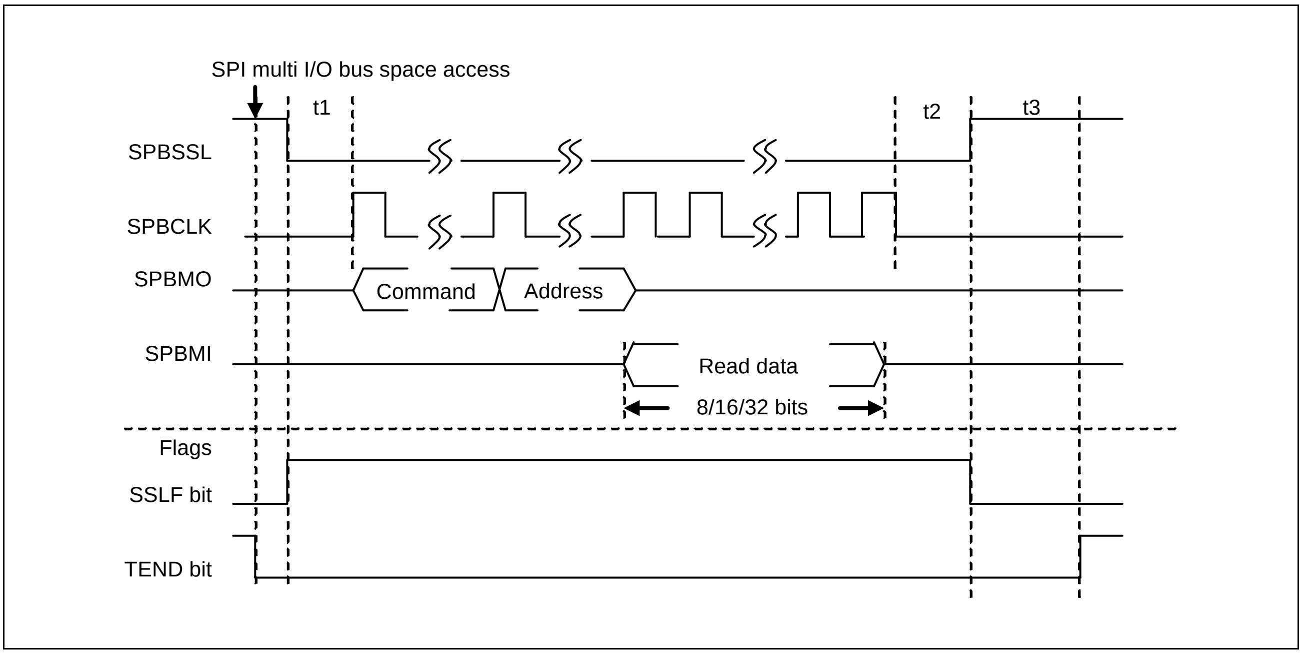


Figure 17.4 Normal Read Operation Timing

17.4.5.2 Burst Read Operation

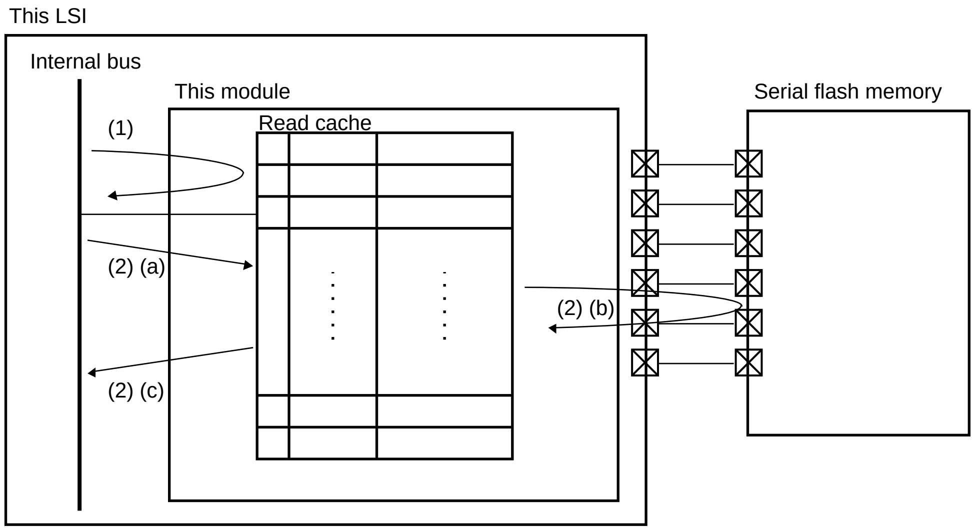
When the RBE bit in SFMAnDRCR is set to 1, burst read operation is performed.

Read cache is enabled in the burst read operation. For read cache operation, see **Section 17.4.7, Read Cache**.

For reading bytes, words, or longwords, the read cache is first referred to for the data. When the read cache contains the data, the data is read from the read cache without accessing the serial flash memory. When the read cache does not contain the data, burst read operation is performed in the serial flash memory and the read data is stored in the read cache. The data transfer length at that time is 64 bits × RBURST[3:0] bits and the data is always read from the 64-bit boundary.

The SPBSSL pin status after data transfer can be selected by using the SSLE bit in SFMAnDRCR. When the SSLE bit is set to 0, the SPBSSL pin is negated after data transfer. For an operation performed when the SSLE bit is set to 1, see **Section 17.4.5.3, Burst Read Operation with Automatic SPBSSL Negation**, just below.

A pattern diagram of this operation and a burst read operation timing diagram when SSLE bit is set to 0 are shown in **Figure 17.5, Burst Read Operation** and **Figure 17.6, Burst Read Operation Timing (SSLE Bit = 0)**.



(1) When the read cache contains the data

The data is read from the read cache without performing SPI communication.

(2) When the read cache does not contain the data

(a) The read cache is accessed to confirm that the data is not in the read cache.

(b) The data is read from the serial flash memory and the read data is stored in the read cache.

(c) The data is read from the read cache.

Figure 17.5 Burst Read Operation

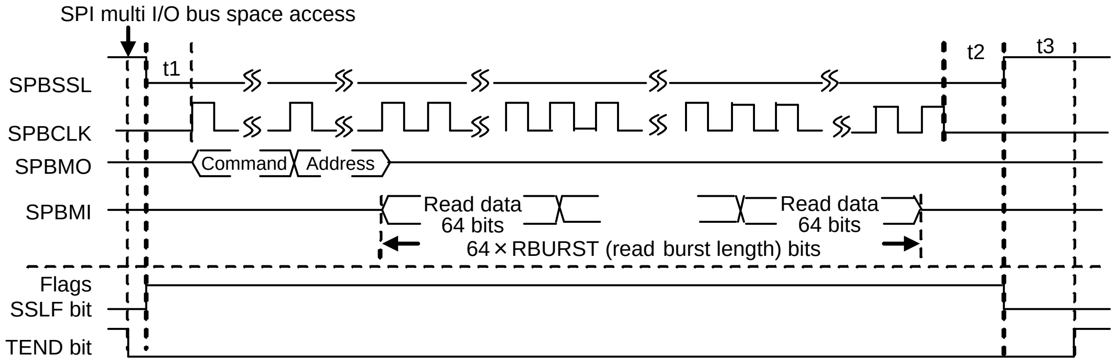


Figure 17.6 Burst Read Operation Timing (SSLE Bit = 0)

17.4.5.3 Burst Read Operation with Automatic SPBSSL Negation

When SSLE bit in SFMAnDRCR is set to 1, this module does not negate the SPBSSL pin after the burst read transfer. When accessing the next time, if the address is continuous with the previous read address, the burst read operation is performed without issuing the command, optional command, address, option data, or dummy cycle. If the address is not continuous with the previous read address, the SPBSSL pin is once negated and the burst read operation is performed after issuing the command, optional command, address, option data, or dummy cycle.

Burst read timing diagrams for continuous address and non-continuous address are shown in **Figure 17.7, Burst Read Timing for Continuous Address (SSLE Bit = 1)** and **Figure 17.8, Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)**.

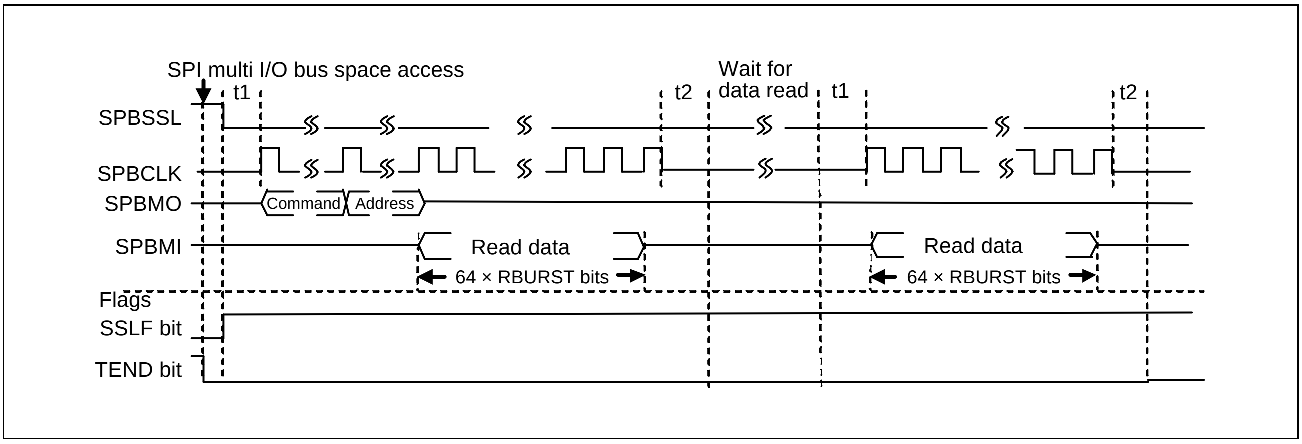


Figure 17.7 Burst Read Timing for Continuous Address (SSLE Bit = 1)

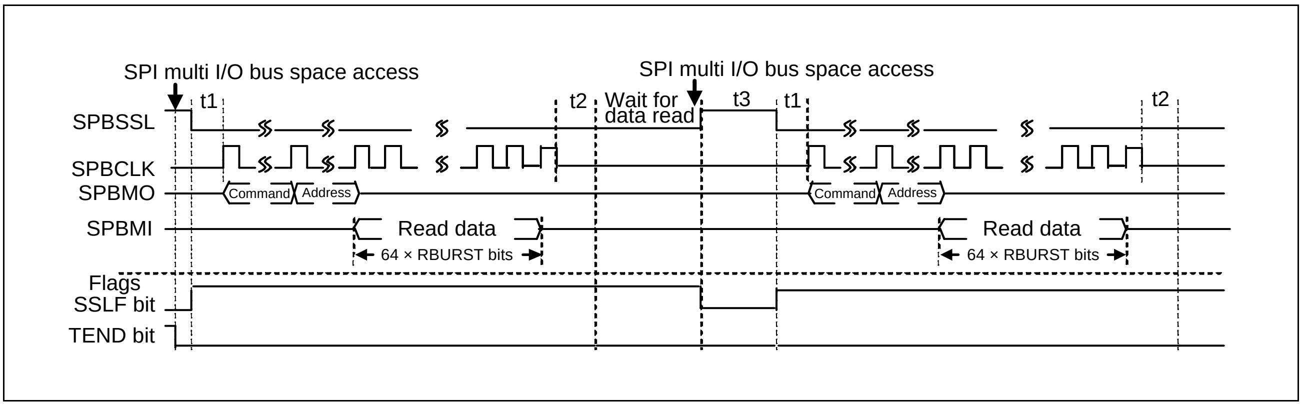


Figure 17.8 Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)

For the next access after negation of the SPBSSL with the SSLN bit in SFMAnDRCR with this operation, read SSLF = 0 in SFMAnCMNSR to confirm that the SPBSSL has been negated.

17.4.6 Initial Setting Flow

An example of an initial setting flow in external address space read mode is shown in **Figure 17.9, Example of Initial Setting Flow in External Address Space Read Mode**.

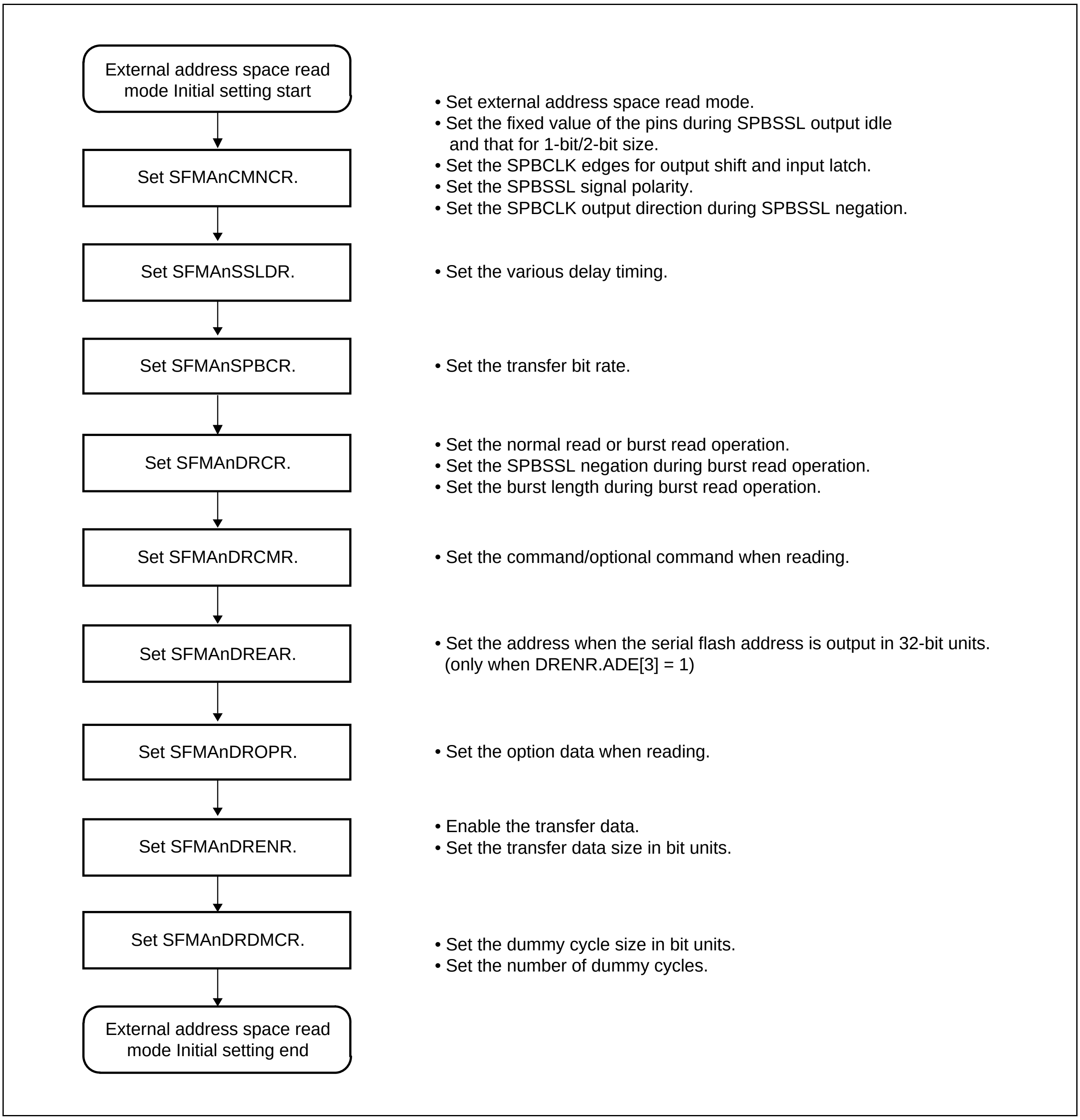


Figure 17.9 Example of Initial Setting Flow in External Address Space Read Mode

17.4.7 Read Cache

This module has a simple built-in read cache. The read cache can be used during external address space read mode and burst read operation. The read cache is configured with a line size of 64 bits and 16 entries.

Read cache configuration is shown in **Figure 17.10, Read Cache Configuration**.

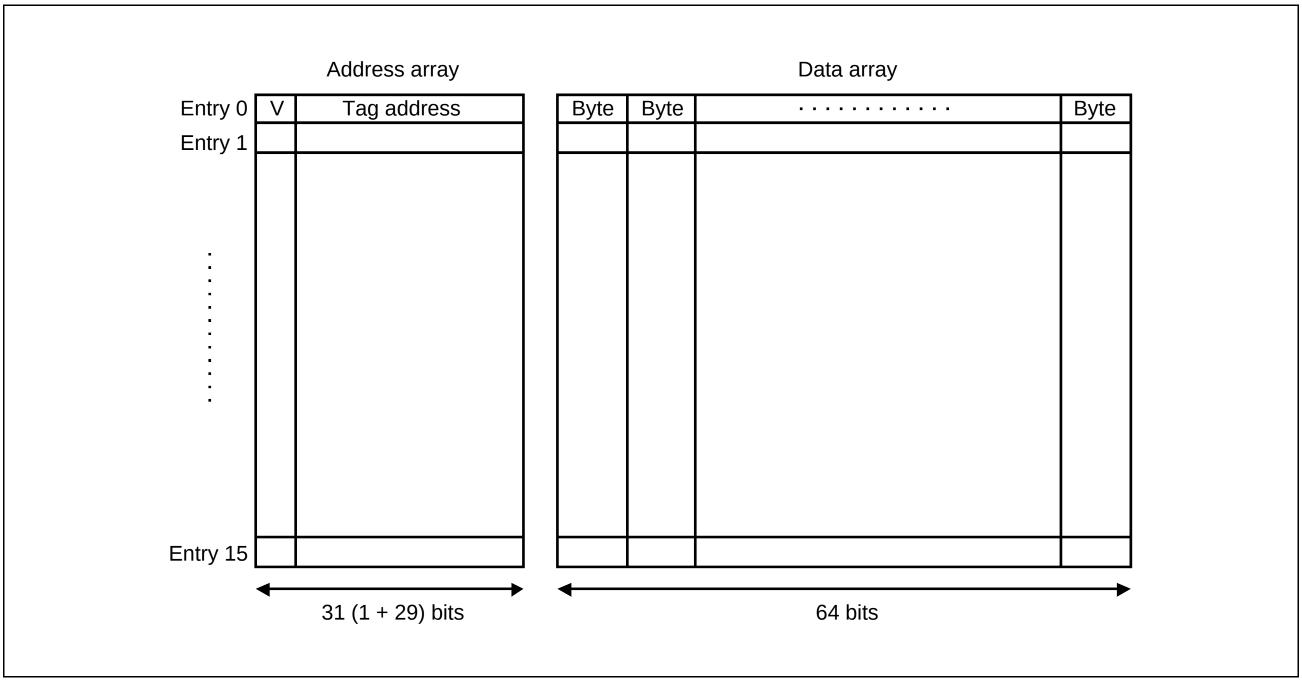


Figure 17.10 Read Cache Configuration

17.4.7.1 Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.

The tag address bits hold the address used for the serial flash memory. Address bits 31 to 3 are used for the tag address.

Address bits 23 to 3 are enabled when address output is 24 bits.
Address bits 31 to 3 are enabled when address output is 32 bits.

17.4.7.2 Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

17.4.7.3 Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 × RBURST (read burst length) data is read from the serial flash memory and the read cache is updated, the data is returned to the bus master.

17.4.7.4 Data Replacement

The write pointer is used to update data. In case of read-miss, the RBURST (read burst length) portion data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data. Whether data is referred to or not will not affect the replacement order of data.

17.4.8 SPI Operating Mode

This module can carry out an arbitrary SPI operation by using the register settings.

The transfer format is determined based on the common control register (SFMAnCMNCR), SSL delay register (SFMAnSSLDR), bit rate setting register (SFMAnSPBCR), SPI mode control register (SFMAnSMCR), SPI mode command setting register (SFMAnSMCMR), SPI mode address setting register (SFMAnSMADR), SPI mode option setting register (SFMAnSMOPR), and SPI mode enable setting register (SFMAnSMENR), SPI mode read data register (SFMAnSMRDR), SPI mode write data register (SFMAnSMWDR), and SPI mode dummy cycle setting register (SFMAnSMDMCR).

SPI operating mode can be used for reading the status of the serial flash memory and writing to the serial flash memory.

In this mode, one transfer refers to the operation from when the SPIE bit in SFMAnSMCR is set to 1 when the TEND bit in SFMAnCMNSR is set to 1.

17.4.8.1 Transfer Start

The transfer of data is started in the set transfer format by setting the SPIE bit in SFMAnSMCR to 1. When write operation is enabled, the SPI mode write data register is transmitted to the serial flash memory. When read operation is enabled, data read from the serial flash memory is stored into the SPI mode read data register.

The SPI operation timing is shown in **Figure 17.11, SPI Operation Timing**.

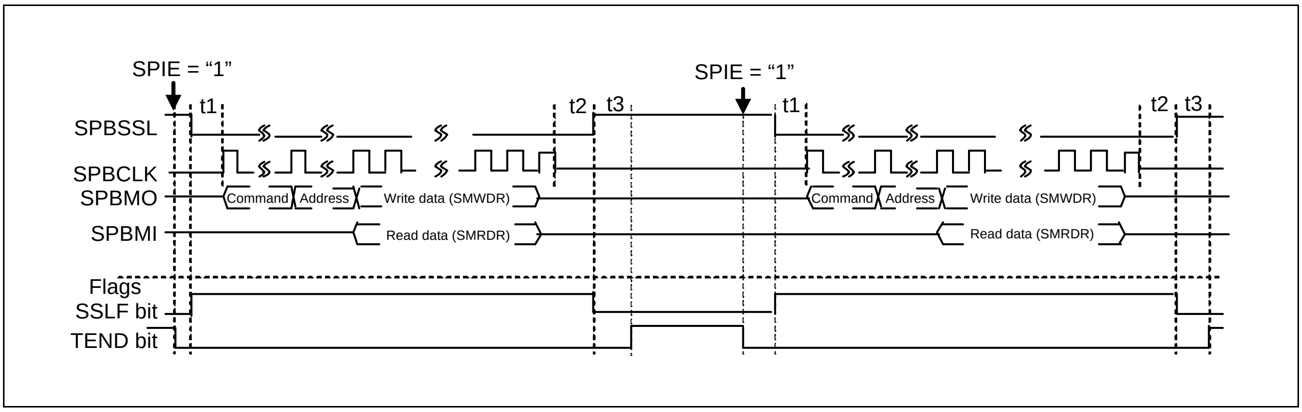


Figure 17.11 SPI Operation Timing

17.4.8.2 Read/Write Enable

- Read operation: Data can be read by setting the SPIRE bit in SFMAnSMCR to 1. The read data is stored into SFMAnSMRDR.
- Write operation: Data can be written by setting the SPIWE bit in SFMAnSMCR to 1. The data stored in SFMAnSMWDR is output.

When the data size is set to 1 bit using the SPIDB[1:0] bits in SFMAnSMENR, data can be transmitted and received by setting the SPIRE and SPIWE bits to 1. However, when the data size is set to 2 or 4 bits by using the SPIDB[1:0] bits, only one of the SPIRE and SPIWE bits should be enabled. The operation is not guaranteed if both the bits are enabled.

17.4.8.3 Retention of SPBSSL Pin Assertion

By setting the SSLKP bit in SFMAnSMCR to 1, assertion of the SPBSSL pin can be continued till the next transfer. With this function, the transfer can be carried out continuously with the SPBSSL kept in the asserted state.

The data transfer timing using the SSLKP bit is shown in **Figure 17.12, Data Transfer Timing using the SSLKP Bit**.

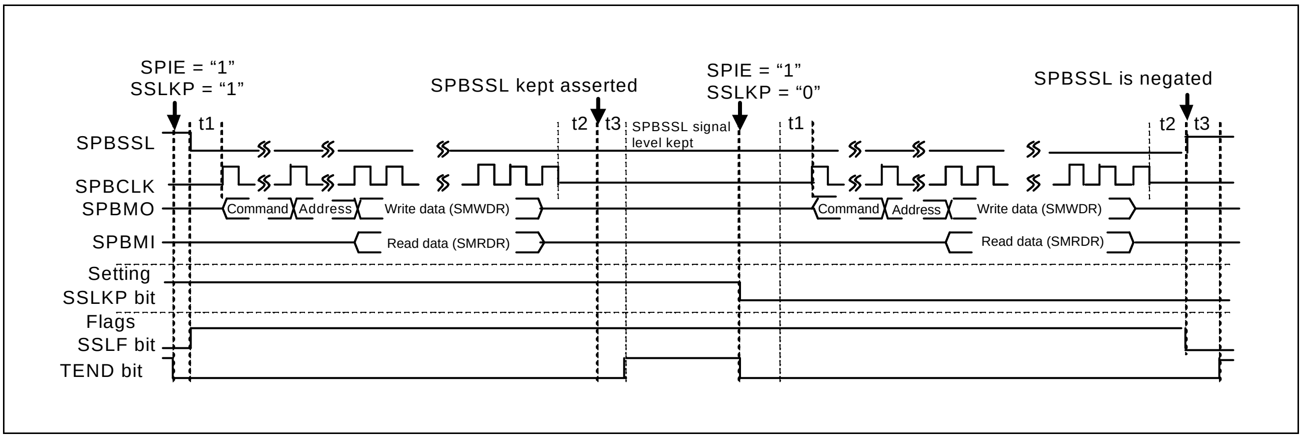
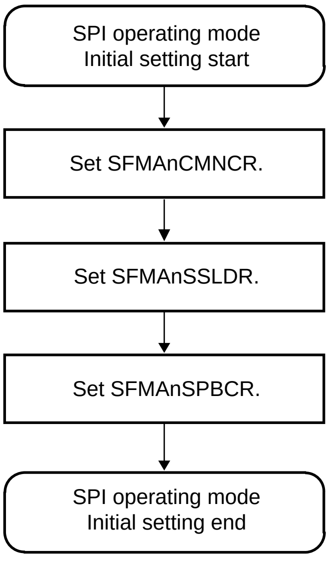


Figure 17.12 Data Transfer Timing using the SSLKP Bit

17.4.8.4 Initial Setting Flow

An example of an initial setting flow in SPI operating mode is shown in **Figure 17.13, Example of Initial Setting Flow in SPI Operating Mode**.



- Set SPI operating mode.
- Set the fixed value of the pins during SPBSSL output idle and that for 1-bit/2-bit size.
- Set the SPBCLK edges for output shift and input latch.
- Set the SPBSSL signal polarity.
- Set the SPBCLK output direction during SPBSSL negation.

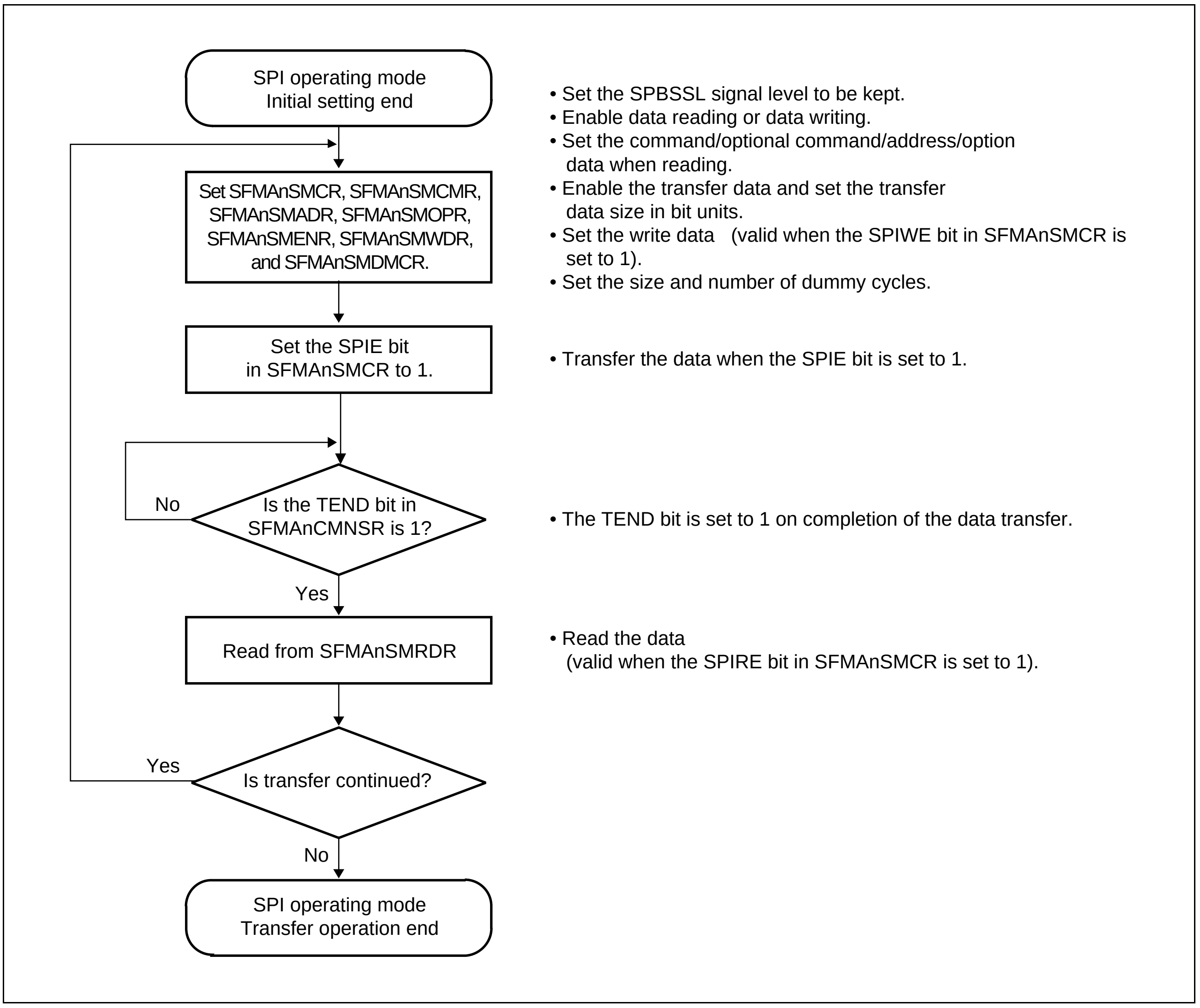
- Set the various delay timing.

- Set the transfer bit rate.

Figure 17.13 Example of Initial Setting Flow in SPI Operating Mode

17.4.8.5 Data Transfer Setting Flow

An example of a data transfer setting flow in SPI operating mode is shown in **Figure 17.14, Example of a Data Transfer Setting Flow in SPI Operating Mode**.



- Set the SPBSSL signal level to be kept.
- Enable data reading or data writing.
- Set the command/optional command/address/option data when reading.
- Enable the transfer data and set the transfer data size in bit units.
- Set the write data (valid when the SPIWE bit in SFMAnSMCR is set to 1).
- Set the size and number of dummy cycles.

- Transfer the data when the SPIE bit is set to 1.

- The TEND bit is set to 1 on completion of the data transfer.

- Read the data (valid when the SPIRE bit in SFMAnSMCR is set to 1).

Figure 17.14 Example of a Data Transfer Setting Flow in SPI Operating Mode

17.4.9 Transfer Format

17.4.9.1 SPBSSL Pin Enable Polarity Control

The enable polarity of the SPBSSL pin can be changed with the SSLP bit in SFMAnCMNCR.

17.4.9.2 SPBCLK Output

The SPBCLK output direction during SPBSSL negation can be set with the CPOL bit in SFMAnCMNCR.

17.4.9.3 Data Transmission and Reception Timing

Data is transmitted and received at either the odd or even edges. The data transmission timing can be set to the odd or even edge with the CPHAT bit in SFMAnCMNCR. Similarly, the data reception timing can be set to the odd or even edge with the CPHAR bit in SFMAnCMNCR.

17.4.9.4 Delay Settings

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). It can be set with the SCKDL[2:0] bits in SFMAnSSLDR. t2 is the time period till the SPBSSL signal negation after the SPBCLK oscillation is stopped (SPBSSL negation delay). It can be set with the SLNDL[2:0] bits in SFMAnSSLDR. t3 is the time period required to prevent SPBSSL signal assertion for the next transfer after the end of the previous transfer (next access delay). It can be set with the SPNDL[2:0] bits in SFMAnSSLDR.

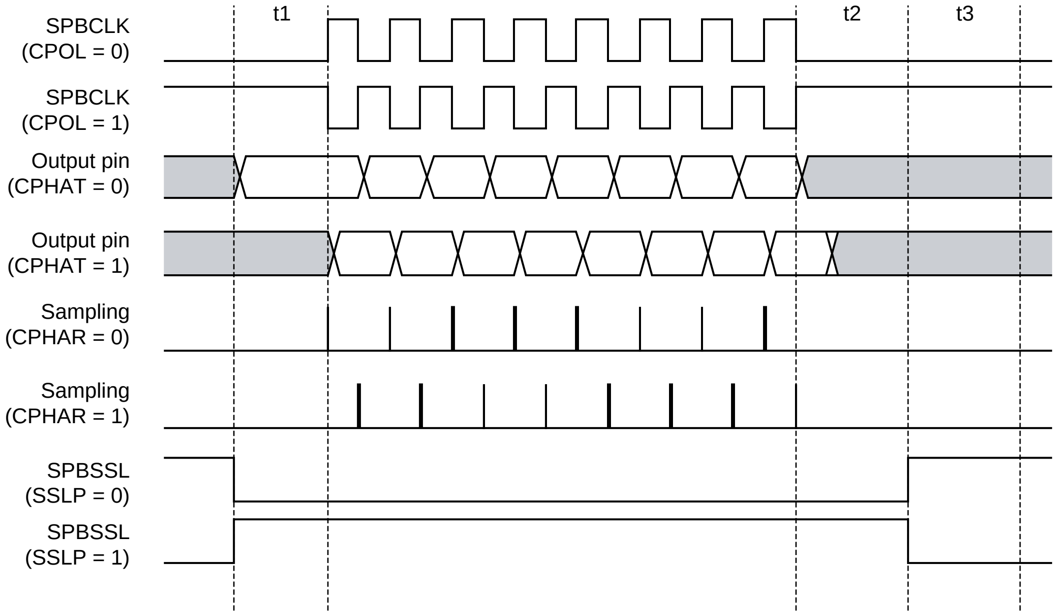


Figure 17.15 Transfer Format

17.4.10 Data Format

This module can input and output data in the order of command, optional command, address, option data, dummy cycle and data.

17.4.10.1 Data Registers

Table 17.34, Data Registers shows the input and output data.

Table 17.34 Data Registers

Data	External Address Space Read Operation	SPI Operation
Command (8 bits)	CMD[7:0] bits in SFMAnDRCMR	CMD[7:0] bits in SFMAnSMCMR
Optional command (8 bits)	OCMD[7:0] bits in SFMAnDRCMR	OCMD[7:0] bits in SFMAnSMCMR
Address (32/24 bits)	32 bits: SFMAnDREAR.EAV[5:0] + lower [25:0] bits of the read address. 24 bits: Lower [23:0] bits of the read address	32 bits: ADR[31:0] bits in SFMAnSMADR 24 bits: ADR[23:0] bits in SFMAnSMADR
Option data (8 bits × 4)	SFMAnDROPR	SFMAnSMOPR
Dummy cycle (1 to 8 cycles)	SFMAnDRDMCR	SFMAnSMDMCR (only when read)
Transfer data	Normal read: 8, 16, and 32 bits Burst read: 64 × RBURST bits	Read: SFMAnSMRDR Write: SFMAnSMWDR

17.4.10.2 Data Enable

In external address space read mode, transfer enable or disable of the command, optional command, address, option data, and dummy cycle can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], and DME bits in SFMAnDRENR, respectively. The size and number of dummy cycles can be controlled with the data read dummy cycle setting register (SFMAnDRDMCR).

Similarly, in SPI operating mode, enable or disable of the command, optional command, address, option data, dummy cycle, and transfer data can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], DME, and SPIDE[3:0] bits in SFMAnSMENR, respectively. However, disabling all the above parameters is prohibited in SPI operating mode. At least one of them except dummy cycle must be enabled. The size and number of dummy cycles can be controlled with the SPI mode dummy cycle setting register (SFMAnSMDMCR).

For the address and option data in external address space read mode; and the address, option data, and transfer data in SPI operating mode, the enable bit setting allowed is determined according to the transfer data size. For the allowed setting combinations of the enable bits and transfer data size, refer to the description of the pertinent registers.

If data is disabled, that data is skipped, and input and output of the next data is carried out. The command, optional command, address, and option data are always output. During dummy cycles, the state of the used pins is Hi-Z. In external address space read mode, data is always input; and in SPI operating mode, input and output of data is determined based on the settings of the SPIRE and SPIWE bits in SFMAnSMCR.

There are some restrictions on dummy cycle insertion; refer to the description of the DME bits in SFMAnDRENR and SFMAnSMENR for details.

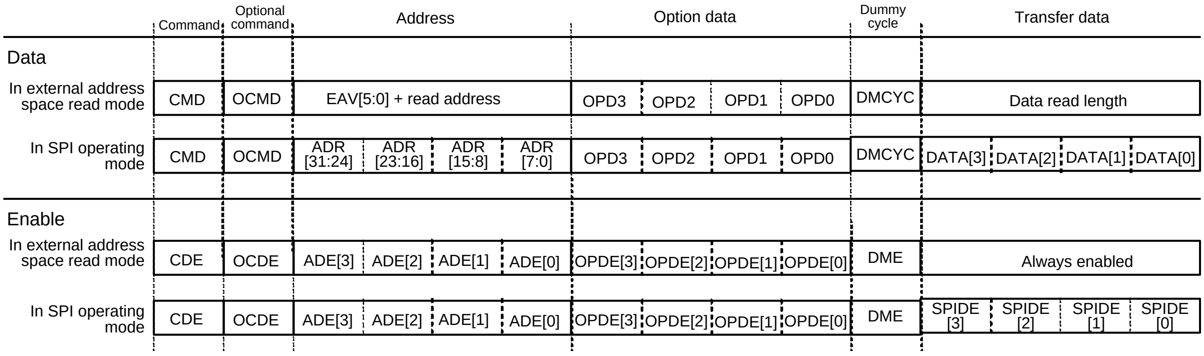


Figure 17.16 Data and Enable

17.4.10.3 Bit Size

In external address space read mode, the size of the command, optional command, address, option data, and the read data in bit units is respectively controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and DRDB[1:0] bits in SFMAnDRENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SFMAnDRDMCR.

Similarly, in SPI operating mode, the size of the command, optional command, address, option data, and read write data in bit units is controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and SPIDB[1:0] bits in SFMAnSMENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SFMAnSMDMCR.

(1) 1-bit Size

When the size is set to 1 bit, SPBMI pin will be the input pin and SPBMO pin will be the output pin. SPBIO2 and SPBIO3 pins are not used.

Figure 17.17, Transfer Format Example with 1-Bit Data Size shows the transfer format example.

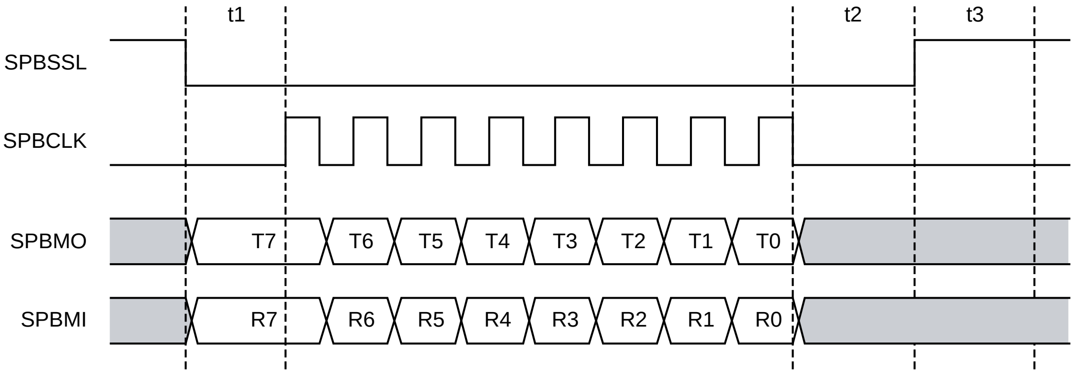


Figure 17.17 Transfer Format Example with 1-Bit Data Size

(2) 2-bit Size

When the size is set to 2 bits, SPBIO0 and SPBIO1 pins will be either the input pins or the output pins. SPBIO2 and SPBIO3 pins are not used.

Figure 17.18, Transfer Format Example with 2-Bit Data Size shows the transfer format example.

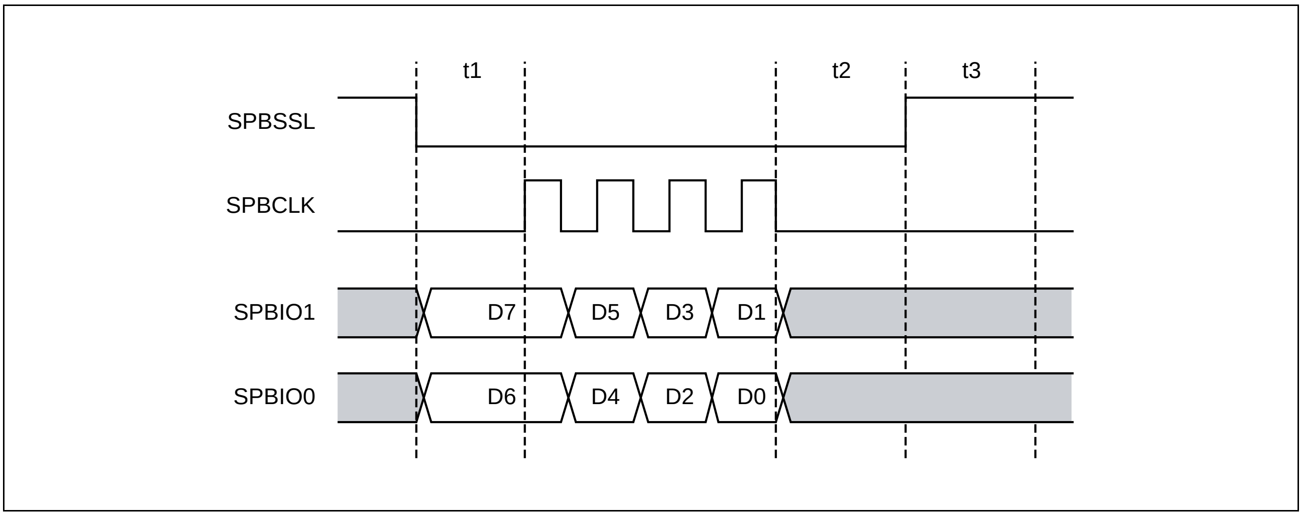


Figure 17.18 Transfer Format Example with 2-Bit Data Size

(3) 4-bit Size

When the size is set to 4 bits, SPBIO0, SPBIO1, SPBIO2, and SPBIO3 pins will be either the input pins or the output pins.

Figure 17.19, Transfer Format Example with 4-Bit Data Size shows the transfer format example.

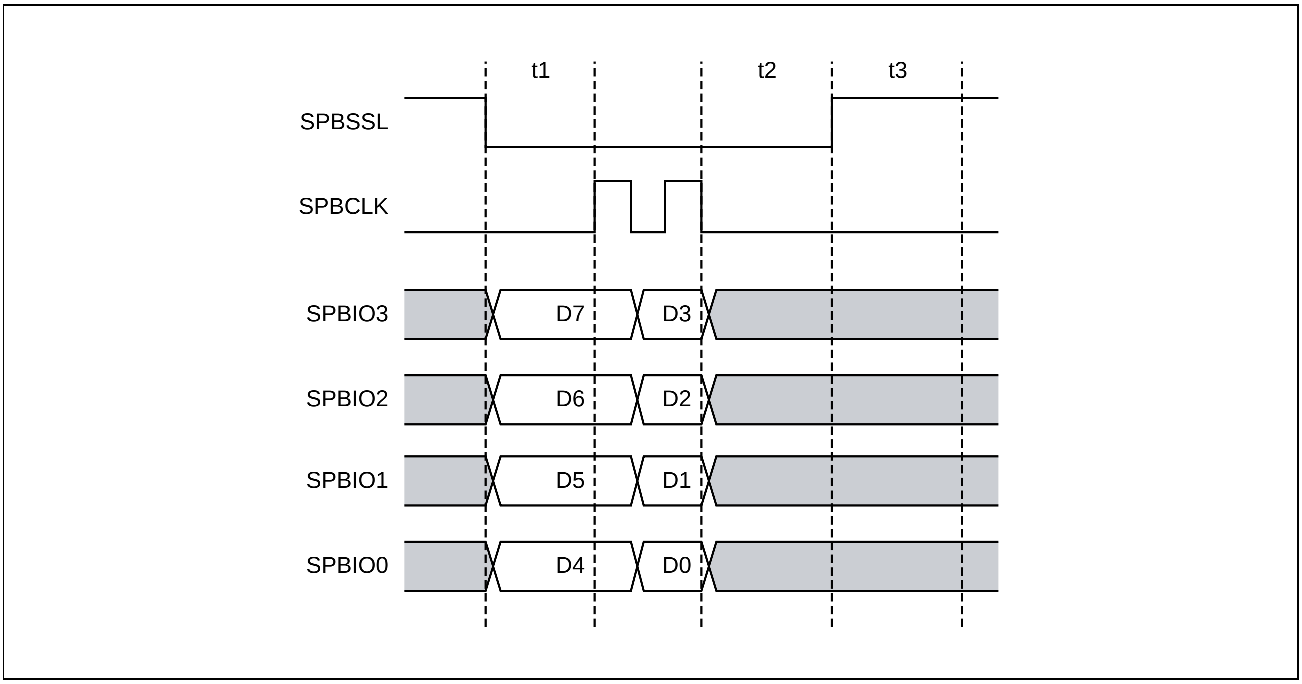


Figure 17.19 Transfer Format Example with 4-Bit Data Size

17.4.11 Data Pin Control

With this module, the status of pins can be automatically changed based on the data size to be used and the read/write settings. The pin status during the SPBSSL negation can be set with the MOIIO3, MOIIO2, MOIIO1, and MOIIO0 bits in SFMAnCMNCR. The SPBSSL and SPBCLK pins are always output pins. The status of respective pins is specified in **Table 17.35, Pin Status (1)** to **Table 17.38, Pin Status (4)**.

Table 17.35 Pin Status (1)

Pin	SPBSSL Negation	SPBSSL Assertion		
		Command, Optional Command, Address, Option Data		
		1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	MOIIO0 bit value	Output	Output	Output
SPBMI/SPBIO1	MOIIO1 bit value	Hi-Z	Output	Output
SPBIO2	MOIIO2 bit value	IO2FV bit value	IO2FV bit value	Output
SPBIO3	MOIIO3 bit value	IO3FV bit value	IO3FV bit value	Output

Table 17.36 Pin Status (2)

Pin	Transfer Data					
	External Address Space Read Operation			SPI Operation		
				SPIRE Bit = 1, SPIWE Bit = 0		
	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	IO0FV bit value	Input	Input	IO0FV bit value	Input	Input
SPBMI/SPBIO1	Input	Input	Input	Input	Input	Input
SPBIO2	IO2FV bit value	IO2FV bit value	Input	IO2FV bit value	IO2FV bit value	Input
SPBIO3	IO3FV bit value	IO3FV bit value	Input	IO3FV bit value	IO3FV bit value	Input

Table 17.37 Pin Status (3)

Pin	Transfer Data					
	SPI Operation					
	SPIRE Bit = 0, SPIWE Bit = 1			SPIRE Bit = 1, SPIWE Bit = 1		
	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	Output	Output	Output	Output	Setting prohibited	Setting prohibited
SPBMI/SPBIO1	Hi-Z	Output	Output	Input	Setting prohibited	Setting prohibited
SPBIO2	IO2FV bit value	IO2FV bit value	Output	IO2FV bit value	Setting prohibited	Setting prohibited
SPBIO3	IO3FV bit value	IO3FV bit value	Output	IO3FV bit value	Setting prohibited	Setting prohibited

Table 17.38 Pin Status (4)

Pin	Dummy Cycle		
	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	IO0FV bit value	Hi-Z	Hi-Z
SPBMI/SPBIO1	Hi-Z	Hi-Z	Hi-Z
SPBIO2	IO2FV bit value	IO2FV bit value	Hi-Z
SPBIO3	IO3FV bit value	IO3FV bit value	Hi-Z

17.4.12 SPBSSL Pin Control

Negation conditions of the SPBSSL pin are as follows.

17.4.12.1 External Address Space Read Mode

(1) Normal read operation (RBE bit in SFMAnDRCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(2) Burst read without automatic SPBSSL negation (RBE bit in SFMAnDRCR = 1, SSLE bit in SFMAnDRCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(3) Burst read with automatic SPBSSL negation (RBE bit in SFMAnDRCR = 1, SSLE bit in SFMAnDRCR = 1)
- SPBSSL negated after t2 cycle when the read address is not continuous with the previously read address
- SPBSSL negated after the SSLN bit in SFMAnDRCR is set to 1

17.4.12.2 SPI Operating Mode

(1) SPBSSL pin assertion not retained (SSLKP bit in SFMAnSMCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(2) SPBSSL pin assertion retained (SSLKP bit in SFMAnSMCR = 1)
SPBSSL not negated.
When to be negated, data should be transferred after setting the SSLKP bit to 0.

17.4.13 Flags

This module has two flag bits SSLF and TEND in SFMAnCMNSR. These bits are read-only bits.

17.4.13.1 SSLF Bit

This bit indicates the SPBSSL pin status. The status is 1 when the SPBSSL is asserted, and the status is 0 when the SPBSSL is negated.

17.4.13.2 TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.

During t1 time period, data transfer, t2 time period, t3 time period, and waiting for read access by burst read and SPBSSL automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.

When other than the above, the TEND bit is read as 1 to indicate that transfer of data has ended.

17.4.13.3 Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.

The registers which can be written to, except the SSLN bit in SFMAnDRCR, should be modified when TEND = 1. Read SFMAnSMRDR when TEND = 1. SFMAnCMNSR can always be read.

17.5 Usage Notes

17.5.1 Notes on Transfer to Read Data in SPI Operating Mode

In SPI operating mode, take note of the following points for caution when setting the SPI mode enable setting register (SFMAnSMENR) to enable transfer only for reading data.

“Transfer only for reading data” indicates transfer to read data while the CDE, OCDE, ADE[3:0], and OPDE[3:0] bits in SFMAnSMENR are all 0.

17.5.1.1 Transfer to Read Data While the Signal on the SPBSSL Pin is De-asserted

Set the SFMAnSMENR.SPIDE[3:0] bits to 1100 or 1111 when transfer only for reading data is to proceed.

Transfer will not proceed normally if the setting of the SFMAnSMENR.SPIDE[3:0] bits is 1000.

17.5.1.2 Transfer to Read Data While the Signal on the SPBSSL Pin is Asserted

When transfer only for reading data is to proceed, set the SFMAnSMENR.SPIDE[3:0] bits to 1100 or 1111, or end the immediately preceding transfer with reading data.

When the immediately preceding transfer is of a command, optional command, address, or option data, or is transfer for writing data, the subsequent transfer only for reading data will not proceed normally if the setting of the SFMAnSMENR.SPIDE[3:0] bits is 1000.

17.5.2 Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode

Be sure to set the SPIWE bit in the SFMAnSMCR register to 1 when the transfer of a command, optional command, address, or option data is started while the SPBSSL pin is being asserted in SPI operating mode.

Section 18 Multi Media Card Interface A (MMCA)

This section contains a generic description of Multi Media Card Interface A (MMCA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of MMCA.

18.1 Features of RH850/F1KH, RH850/F1KM MMCA

18.1.1 Number of Units

This microcontroller has the following number of MMCA units.

Table 18.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	—	—	1
Name	—	—	MMCAn (n = 0)

Table 18.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	—	—	—	—	—
Name	—	—	—	—	—

Table 18.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of units	—	—	—	—
Name	—	—	—	—

Table 18.4 Index

Index	Description
n	Throughout this section, the individual MMCA units are identified by the index “n”; for example, MMCAnCE_CMD_SET (n = 0) is the MMCAn command setting register.

18.1.2 Register Base Address

MMCAn base address is listed in the following table.

MMCAn register addresses are given as an offset from the base address.

Table 18.5 Register Base Address

Base Address Name	Base Address
<MMCA0_base>	FFED 9000_H

18.1.3 Clock Supply

The MMCAn clock supply is listed in the following table.

Table 18.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
MMCAn	MMCA module clock	CKSCLK_IPERI2	MMCA module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI2	Bus clock

18.1.4 Interrupt Requests

The MMCAn interrupts are listed in the following table.

Table 18.7 Interrupt Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
MMCA0			
INTMMCA0	MMCA interrupt	354	—
DMAMMCA0	MMCA DMA request	—	127

18.1.5 Reset Sources

MMCAn reset sources are listed in the following table. MMCAn is initialized by these reset sources.

Table 18.8 Reset Sources

Unit Name	Reset Source
MMCAn	All reset sources (ISORES)

18.1.6 External Input/Output Signals

External input/output signals of MMCAn are listed below.

Table 18.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
MMCA0		
MMCA0CLK	MMCA Clock	MMCA0CLK
MMCA0CMD	MMCA Command / Response	MMCA0CMD
MMCA0DAT[7:0]	MMCA Data[7:0]	MMCA0DAT[7:0]

18.2 Overview

18.2.1 Features

- Compliant with JEDEC STANDARD JESD84-A441 (neither DDR mode nor 1.8-V operation is supported)
- Supports 1-/4-/8-bit MMC bus width
- Supports Backward-compatible mode.
  High-speed mode is not supported.
- MMC Clock frequency = MMCA module clock frequency/2^k (k = 1 to 10)
- Supports block transfer
  Stream transfer is not supported.
- Supports boot operation
  The alternative boot operation is not supported.
- MMC clock frequency settings are adjustable in boot mode.
- Supports high priority interrupt (HPI)
  HPIs in the sequence of CMD6, CMD24, CMD25 (pre-defined), and CMD38 are supported.
- Supports background operation
- Interrupt requests: normal operation and error/timeout.
- DMA transfer requests: buffer write and buffer read

18.2.2 Block Diagram

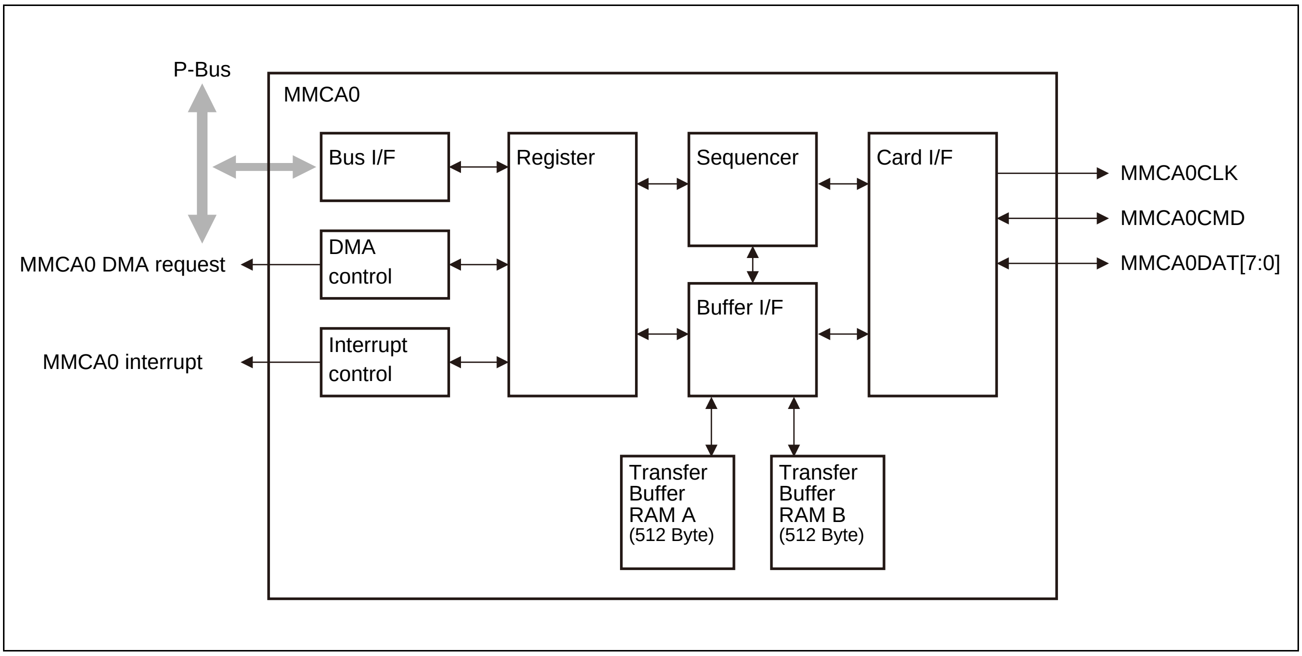


Figure 18.1 Block Diagram of MMCA

18.3 Registers

18.3.1 List of Registers

MMCA registers are listed in the following table.

For details on <MMCAn_base>, see **Section 18.1.2, Register Base Address**.

Table 18.10 List of Registers

Module Name	Register Name	Symbol	Address
MMCAn	MMCAn command setting register	MMCAnCE_CMD_SET	<MMCAn_base> + 00_H
	MMCAn argument register	MMCAnCE_ARG	<MMCAn_base> + 08_H
	MMCAn argument register for automatically-issued CMD12	MMCAnCE_ARG_CMD12	<MMCAn_base> + $0C_H$
	MMCAn command control register	MMCAnCE_CMD_CTRL	<MMCAn_base> + 10_H
	MMCAn transfer block setting register	MMCAnCE_BLOCK_SET	<MMCAn_base> + 14_H
	MMCAn clock control register	MMCAnCE_CLK_CTRL	<MMCAn_base> + 18_H
	MMCAn buffer access configuration register	MMCAnCE_BUF_ACC	<MMCAn_base> + $1C_H$
	MMCAn response register 3	MMCAnCE_RESP3	<MMCAn_base> + 20_H
	MMCAn response register 2	MMCAnCE_RESP2	<MMCAn_base> + 24_H
	MMCAn response register 1	MMCAnCE_RESP1	<MMCAn_base> + 28_H
	MMCAn response register 0	MMCAnCE_RESP0	<MMCAn_base> + $2C_H$
	MMCAn response register for automatically-issued CMD12	MMCAnCE_RESP_CMD12	<MMCAn_base> + 30_H
	MMCAn data register	MMCAnCE_DATA	<MMCAn_base> + 34_H
	MMCAn boot operation setting register	MMCAnCE_BOOT	<MMCAn_base> + $3C_H$
	MMCAn interrupt flag register	MMCAnCE_INT	<MMCAn_base> + 40_H
	MMCAn interrupt enable register	MMCAnCE_INT_EN	<MMCAn_base> + 44_H
	MMCAn status register 1	MMCAnCE_HOST_STS1	<MMCAn_base> + 48_H
	MMCAn status register 2	MMCAnCE_HOST_STS2	<MMCAn_base> + $4C_H$
	MMCAn software reset register	MMCAnCE_SWRESA	<MMCAn_base> + $7C_H$

18.3.2 MMCAnCE_CMD_SET — MMCAn Command Setting Register

MMCAnCE_CMD_SET sets a command sequence. For the setting values of MMCAnCE_CMD_SET, see **Section 18.4.4.13, Setting Values of MMCAnCE_CMD_SET**. The command sequence starts when the settings have been made in bits 31 to 16. Note that writing to MMCAnCE_CMD_SET is disabled while a command sequence is proceeding (the CMDSEQ bit in MMCAnCE_HOST_STS1 is 1).

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 00_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BOOT	CMD[5:0]						RTYP[1:0]		RBSY	—	WDAT	DWEN	CMLTE	CMD 12EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIDXC[1:0]		RCRC7C[1:0]		—	CRC 16C	BOOT ACK	CRC STE	TBIT	OPDM	—	—	SBIT	—	DATW[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W

Table 18.11 MMCAnCE_CMD_SET Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	BOOT	Boot Operation 0: Command sequence other than for boot operations 1: Command sequence for boot operations
29 to 24	CMD[5:0]	Command Index Set a command index ([45:40]).
23, 22	RTYP[1:0]	Response Type 00: No response 01: 6-byte response (R1, R1b, R3, R4, R5) 10: 17-byte response (R2) 11: Setting prohibited
21	RBSY	Response Busy Select Selects whether "busy" is involved in response reception. 0: No response busy 1: Response busy involved (R1b)
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	WDAT	Presence/Absence of Data 0: No data 1: With data
18	DWEN	Read/Write (valid when “with data” is selected) 0: Read from the card. 1: Write to the card.
17	CMLTE	Single/Multi Block Transfer Select (valid when “with data” is selected) 0: Single-block transfer 1: Multi-block transfer

Table 18.11 MMCAnCE_CMD_SET Register Contents

Bit Position	Bit Name	Function
16	CMD12EN	Automatic CMD12 Issuance (valid when multi-block transfer is selected)*[1] 0: Disables automatic CMD12 issuance. 1: Enables automatic CMD12 issuance. For details of automatic CMD12 issuance, see **Section 18.4.3.4, Automatic CMD12 Issuance.** **NOTE:** Set the transfer block size to 512 bytes. Set the RBSY bit to 0.
15, 14	RIDXC[1:0]	Response Index Check Specify the items to be checked in bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response. 00: Checks the response index (check whether matched with a command index). 01: Checks the check bits (check whether all the bits are set to 1). 10: No checking 11: Setting prohibited
13, 12	RCRC7C[1:0]	Response CRC7 Check Specify the items to be checked in bits [7:1] of a 6-byte response or of a 17-byte response. 00: Checks CRC7 (set the response type to 01). 01: Checks the check bits (check whether all the bits are set to 1) (set the response type to 01). 10: Checks internal CRC7 (R2 only) (set the response type to 10). 11: No checking
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	CRC16C	CRC16 Check in Reception (valid when "with data" and "read" are selected) 0: Checks CRC16 in reception. 1: Does not check CRC16 in reception (used when CMD14).
9	BOOTACK	Receive Boot Acknowledge (valid in boot mode) 0: Boot acknowledge is not received. 1: Boot acknowledge is received.
8	CRCSTE	CRC Status Reception (valid when "with data" and "write" are selected) 0: Receives CRC status. 1: Does not receive CRC status (used when CMD19).
7	TBIT	Transmission Bit Setting 0: Sets the transmission bit ([46]) to 1. 1: Sets the transmission bit ([46]) to 0.
6	OPDM	Open-Drain Output Mode 0: Normal output 1: Open-drain output This setting is only applied to the MMCAnCMD line.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SBIT	Read Data Start Bit Detection Setting (valid when "with data" and "read" are selected) 0: Detects a start bit when the valid MMCAnDAT signals specified by the DATW bits are all 0. 1: Detects a start bit when MMCAnDAT[0] is 0.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	DATW[1:0]	Data Bus Width Setting (valid when "with data" is selected) 00: 1 bit 01: 4 bits 10: 8 bits 11: Setting prohibited

Note 1. It is recommended to use the pre-defined multi-block transfer by setting this bit to 0 for a higher data transfer rate.

18.3.3 MMCAnCE_ARG — MMCAn Argument Register

MMCAnCE_ARG sets the argument for the command to be transmitted. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.12 MMCAnCE_ARG Register Contents

Bit Position	Bit Name	Function
31 to 0	ARG[31:0]	Set bits [39:8] of the command. **NOTE:** Set the argument of automatically-issued CMD12 by MMCAnCE_ARG_CMD12.

18.3.4 MMCAnCE_ARG_CMD12 — MMCAn Argument Register for Automatically-Issued CMD12

MMCAnCE_ARG_CMD12 is used to set the argument for the automatically-issued CMD12. This register is valid when issuing CMD12 automatically in multi-block transfer. For automatic issuance of CMD12, see **Section 18.4.3.4, Automatic CMD12 Issuance**. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C12ARG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C12ARG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.13 MMCAnCE_ARG_CMD12 Register Contents

Bit Position	Bit Name	Function
31 to 0	C12ARG[31:0]	Set bits [39:8] of the automatically-issued CMD12.

18.3.5 MMCAnCE_CMD_CTRL — MMCAn Command Control Register

MMCAnCE_CMD_CTRL is used to terminate a command sequence forcibly.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BREAK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 18.14 MMCAnCE_CMD_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BREAK	Forcible Termination of Command Sequence Writing 1 to this bit while it is 0 and then writing 0 to it discontinues the current command sequence. After this bit is set as described above, check if the value of the CMDSEQ bit in MMCAnCE_HOST_STS1 has become 0. If this is the case, execute software reset. **NOTE:** A software reset initializes the value of this register, so the setting in this register needs to be remade.

18.3.6 MMCAnCE_BLOCK_SET — MMCAn Transfer Block Setting Register

MMCAnCE_BLOCK_SET specifies the size of the block and the number of blocks for the data to be transferred. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 14_H

Value after reset: $0000\ 0200_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLKCNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLKSIZ[15:0]															
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.15 MMCAnCE_BLOCK_SET Register Contents

Bit Position	Bit Name	Function
31 to 16	BLKCNT[15:0]	Number of Blocks for Transfer **NOTE:** This setting is valid for multi-block transfer.
15 to 0	BLKSIZ[15:0]	Transfer Block Size **NOTE:** Transfer block size should be set as follows. Single-block transfer: 1 to 512 bytes Multi-block transfer: 512 bytes

18.3.7 MMCAnCE_CLK_CTRL — MMCAn Clock Control Register

MMCAnCE_CLK_CTRL controls the MMC clock and sets timeout values. Do not change the setting of this register while a command sequence is in progress.

Access:	This register can be read or written in 32-bit units.
Address:	<MMCAn_base> + 18_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMCB USBSY	—	—	—	—	—	—	CLKEN	—	—	—	—	CLKDIV[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SRSPTO[1:0]		SRBSYTO[3:0]				SRWDTO[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 18.16 MMCAnCE_CLK_CTRL Register Contents

Bit Position	Bit Name	Function
31	MMCBUSBSY	MMC Clock Output control / Division setting prohibition 0: MMC bus is NOT busy (During the command sequence + 10 cycle) 1: MMC bus is busy (During the command sequence + 10 cycle) When a command sequence is started with MMCAnCE_CMD_SET register setting, MMCBUSBSY bit is set to 1 at the same time as CMDSEQ bit is set to 1. After CMDSEQ bit is set to 0 at the command sequence end, MMCBUSBSY bit becomes 0 after 10 cycle of MMC Clock. **NOTE:** When MMC bus is busy, do not set CLKEN bit and the CLKDIV bit of the MMCAnCE_CLK_CTRL register.
30 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CLKEN	MMC Clock Output Control 0: Does not output the MMC clock (fixed to low level). 1: Outputs the MMC clock.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	CLKDIV[3:0]	MMC Clock Frequency Setting 0000: MMCA module clock/2^{1} 0001: MMCA module clock/2^{2} : 0111: MMCA module clock/2^{8} 1000: MMCA module clock/2^{9} 1001: MMCA module clock/2^{10} 1010 to 1111: Setting prohibited For details of MMC clock frequency settings in boot operations, see **Section 18.4.3.5, MMC Clock Frequency in Boot Operations** and **Section 18.3.12, MMCAnCE_BOOT — MMCAn Boot Operation Setting Register**.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	SRSPTO[1:0]	Response Timeout Setting 00: 64 MMC clock cycles 01: 128 MMC clock cycles 10: 256 MMC clock cycles 11: Setting prohibited

Table 18.16 MMCAnCE_CLK_CTRL Register Contents

Bit Position	Bit Name	Function
11 to 8	SRBSYTO[3:0]	Response Busy Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
7 to 4	SRWDTO[3:0]	Write Data/Read Data Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.8 MMCAnCE_BUF_ACC — MMCAn Buffer Access Configuration Register

MMCAnCE_BUF_ACC configures the method of accessing data registers and mode of DMA transfer. Do not set this register again during a command sequence. For explanation of the buffers, see **Section 18.4.3.3, Buffer Structure and Buffer Access**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 1C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DMATYP	DMAWEN	DMAREN	—	—	—	—	—	—	—	ATYP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.17 MMCAnCE_BUF_ACC Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	DMATYP	DMA Transfer setting When DMAWEN or DMAREN is set to 1, set to 1
25	DMAWEN	Buffer Write DMA Transfer Request Enable 0: Disables DMA transfer request for buffer writing. 1: Enables DMA transfer request for buffer writing.
24	DMAREN	Buffer Read DMA Transfer Request Enable 0: Disables DMA transfer request for buffer reading. 1: Enables DMA transfer request for buffer reading.
23 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	ATYP	Buffer access selection 0: When not swapped byte-wise. 1: When swapped byte-wise. **NOTE:** For buffer access, see **Section 18.4.3.3, Buffer Structure and Buffer Access**.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.9 MMCAnCE_RESP3 to MMCAnCE_RESP0 — MMCAn Response Registers 3 to 0

MMCAnCE_RESP3 to MMCAnCE_RESP0 are the registers in which the response that has been received is stored. For the formats of response values, see **Section 18.4.3.1, Command/Response Formats**.

Access:	These registers can be read only in 32-bit units.
Address:	MMCAnCE_RESP3: <MMCAn_base> + 20$_H$ MMCAnCE_RESP2: <MMCAn_base> + 24$_H$ MMCAnCE_RESP1: <MMCAn_base> + 28$_H$ MMCAnCE_RESP0: <MMCAn_base> + 2C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

MMCAnCE_RESP3

Table 18.18 MMCAnCE_RESP3 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[127:96]	Bits [127:96] of a 17-byte response are stored.

MMCAnCE_RESP2

Table 18.19 MMCAnCE_RESP2 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[95:64]	Bits [95:64] of a 17-byte response are stored.

MMCAnCE_RESP1

Table 18.20 MMCAnCE_RESP1 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[63:32]	Bits [63:32] of a 17-byte response are stored.

MMCAnCE_RESP0

Table 18.21 MMCAnCE_RESP0 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[31:0]	Bits [39:8] of a 6-byte response or bits [31:0] of a 17-byte response are stored. **NOTE:** The response to an automatically issued CMD12 is stored in the MMCAnCE_RESP_CMD12 register.

18.3.10 MMCAnCE_RESP_CMD12 — MMCAn Response Register for Automatically-Issued CMD12

MMCAnCE_RESP_CMD12 is a register in which the response to the automatically-issued CMD12 is stored.

Access: This register can be read only in 32-bit units.

Address: <MMCAn_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP12[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP12[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.22 MMCAnCE_RESP_CMD12 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP12[31:0]	Bits [39:8] of the response to automatically-issued CMD12 are stored.

18.3.11 MMCAnCE_DATA — MMCAn Data Register

MMCAnCE_DATA is used to access the buffers of this module. For the write/read data formats, see **Section 18.4.3.2, Data Block Format**.

Access:	This register can be read or written in 32-bit units.
Address:	<MMCAn_base> + 34_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.23 MMCAnCE_DATA Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Buffer write/read data [31:0]

18.3.12 MMCAnCE_BOOT — MMCAn Boot Operation Setting Register

MMCAnCE_BOOT controls the MMC clock and sets timeout values in boot mode. Do not set this register again during a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $3C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BTCLKDIV[3:0]				SBTACKTO[3:0]				S1STBTDATTO[3:0]				SBTDATTO[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.24 MMCAnCE_BOOT Register Contents

Bit Position	Bit Name	Function
31 to 28	BTCLKDIV[3:0]	MMC Clock Frequency Setting in Boot Mode 0000: MMCA module clock/2^{1} 0001: MMCA module clock/2^{2} 0010: MMCA module clock/2^{3} 0011: MMCA module clock/2^{4} 0100 to 1111: Settings prohibited Set these bits to a value lower than that in the CLKDIV bits of MMCAnCE_CLK_CTRL. For MMC clock frequency in boot mode, see **Section 18.4.3.5, MMC Clock Frequency in Boot Operations**.
27 to 24	SBTACKTO[3:0]	Boot Acknowledge Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
23 to 20	S1STBTDATTO [3:0]	1st Boot Data Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
19 to 16	SBTDATTO[3:0]	Interval Between Boot Data Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.13 MMCAnCE_INT — MMCAn Interrupt Flag Register

MMCAnCE_INT indicates various statuses during execution of a command sequence. Each bit is set when its setting condition has been met. To clear flag(s), write 0 only to the bit(s) to be cleared and write 1 to the other bits. For the handling of this module in the case of an error or timeout, see **Section 18.4.3.8, Handling of this Module in the Case of Error/Timeout**.

Access:	This register can be read or written in 32-bit units.
Address:	<MMCAn_base> + 40$_{H}$
Value after reset:	0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CMD12 DRE	CMD12 RBE	CMD12 CRE	DTRAN E	BUF RE	BUF WEN	BUF REN	—	—	RBSYE	CRSPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD VIO	BUF VIO	—	—	WDAT ERR	RDAT ERR	RIDX ERR	RSP ERR	—	—	—	CRCS TO	WDAT TO	RDAT TO	RBSY TO	RSP TO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. Writing 0 initializes the bit. Writing 1 is ignored.

Table 18.25 MMCAnCE_INT Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write 1.
26	CMD12DRE	Automatic CMD12 Issuance & Buffer Read Complete [Setting condition] Response busy for automatically-issued CMD12 and buffer reading have been completed. [Clearing condition] Writing a 0 to this bit. **NOTE:** When CMD12DRE has been set, CMD12RBE, CMD12CRE, and BUFRE have also been set. So, these bits should be cleared as well.
25	CMD12RBE	Automatic CMD12 Issuance Response Busy Complete [Setting condition] Reception of the response and response busy for an automatically-issued CMD12 have been completed. [Clearing condition] Writing a 0 to this bit. **NOTE:** When CMD12RBE has been set, CMD12CRE has also been set. So, this bit should be cleared as well. When CMD12RBE is set during a multi-block write, DTRANE is also set. So clear the bit as well.
24	CMD12CRE	Automatic CMD12 Response Complete [Setting condition] The response to an automatically-issued CMD12 has been received. [Clearing condition] Writing a 0 to this bit.

Table 18.25 MMCAnCE_INT Register Contents

Bit Position	Bit Name	Function
23	DTRANE	Data Transmission Complete [Setting conditions] Transmission of all blocks of data has been completed. – When configured to receive CRC status: Completion of busy (data busy) after reception of CRC status – When configured not to receive CRC status: Completion of data transmission [Clearing condition] Writing a 0 to this bit.
22	BUFRE	Buffer Read Complete [Setting conditions] – Other than in boot operations All blocks of data have been received and the data have been read from the buffer. – In boot operations All blocks of data have been received and the data have been read from the buffer, MMCAnCMD has been modified from 0 to 1, and 48 MMC clock cycles have elapsed. [Clearing condition] Writing a 0 to this bit
21	BUFWEN	Buffer Write Ready [Setting conditions] The buffer has become empty and ready for writing. [Clearing condition] Writing a 0 to this bit. **NOTE:** When writing data to MMCAnCE_DATA by the CPU, this bit should be cleared first and the data corresponding to the block size set in MMCAnCE_BLOCK_SET should be written. Note that this bit is not set when DMA transfer request for buffer writing is enabled.
20	BUFREN	Buffer Read Ready [Setting conditions] Transfer block size of data have been stored in the buffer and it has become ready for reading. [Clearing condition] Writing a 0 to this bit. **NOTE:** When reading data from MMCAnCE_DATA by the CPU, this bit should be cleared first and the data corresponding to the block size set in MMCAnCE_BLOCK_SET should be read. Note that this bit is not set when DMA transfer request for buffer reading is enabled.
19, 18	Reserved	When read, the value after reset is returned. When writing, write 1.
17	RBSYE	Response Busy Complete [Setting condition] Reception of a response and response busy have been completed. [Clearing condition] Writing a 0 to this bit. **NOTE:** When RBSYE has been set, CRSPE has also been set. So, this bit should be cleared as well. Completion of reception of the response and response busy for automatically-issued CMD12 is reflected in CMD12RBE.

Table 18.25 MMCAnCE_INT Register Contents

Bit Position	Bit Name	Function
16	CRSPE	Command/Response Complete [Setting conditions] – Other than in boot operations When configured not to receive response: A command has been transmitted. When configured to receive 6- or 17-byte response: A response has been received. – In boot operations When reception of boot acknowledge has been selected: The boot acknowledge pattern has been received. [Clearing condition] Writing a 0 to this bit. **NOTE:** Completion of reception of the response to automatically-issued CMD12 is reflected in CMD12CRE.
15	CMDVIO	Command Issuance Error [Setting conditions] Illegal setting has been made in MMCAnCE_CMD_SET or MMCAnCE_BLOCK_SET. – During execution of a command sequence: Writing to CMD[5:0] in MMCAnCE_CMD_SET. (The command sequence is not stopped automatically.) – At the start of command sequence: ● Writing to CMD[5:0] in MMCAnCE_CMD_SET when the registers have been set for one of the following combinations of selection. ● No response + response busy ● No response + with data + not during boot operations ● No data + automatic CMD12 issuance ● With data + single-block transfer + automatic CMD12 issuance ● With data + response busy + automatic CMD12 issuance ● With data + transfer block size = 0 ● With data + transfer block size ≥ 513 ● With data + multi-block transfer + number of blocks for transfer = 0 ● Boot operations + no data ● Boot operations + write ● Boot operations + response busy ● Boot operations + automatic CMD12 issuance ● Boot acknowledge reception + not during boot operations [Clearing condition] Writing a 0 to this bit.
14	BUFVIO	Buffer Access Error [Setting conditions] Illegal buffer access has been attempted. – MMCAnCE_ DATA has been accessed exceeding the block size set in BLKSIZ[15:0] in MMCAnCE_BLOCK_SET. – While data is being read from the card: MMCAnCE_DATA has been accessed with BUFREN not set (when DMA is used, with no DMA transfer request asserted for buffer reading). – While data is being written to the card: MMCAnCE_DATA has been accessed with BUFWEN not set (when DMA is used, with no DMA transfer request asserted for buffer writing). [Clearing condition] Writing a 0 to this bit. **NOTE:** When BUFVIO has been set, the command sequence is not stopped automatically. If an error occurs, this bit will be set.
13, 12	Reserved	When read, the value after reset is returned. When writing, write 1.

Table 18.25 MMCAnCE_INT Register Contents

Bit Position	Bit Name	Function
11	WDATERR	Write Data Error [Setting conditions] Error is found in the data that has been written. – Error is in the status of the CRC status. – Error is in the end bits of the CRC status. [Clearing condition] Writing a 0 to this bit. **NOTE:** When WDATERR has been set, the command sequence is stopped automatically.
10	RDATERR	Read Data Error [Setting conditions] Error is found in the read data. – Error is in CRC16 of the read data. – Error is in the end bits of the read data. [Clearing condition] Writing a 0 to this bit. **NOTE:** When RDATERR has been set, the command sequence is stopped automatically.
9	RIDXERR	Response Index Error [Setting conditions] Error has been found in the index value of the response. – When an error has been found in bits [45:40] of a 6-byte response (including automatically-issued CMD12) or bits [133:128] of a 17-byte response. (The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.) [Clearing condition] Writing a 0 to this bit. **NOTE:** When RIDXERR has been set, the command sequence is stopped automatically.
8	RSPERR	Response Error [Setting conditions] Error has been found in the response values of the response. – Transmission bit in the response is 1. – Error is in the end bits of the response. – When an error has been found in bits [7:1] of a 6-byte response (including automatically-issued CMD12) or a 17-byte response. (The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.) – Error in the boot acknowledge pattern – Error in the end bits of the boot acknowledge [Clearing condition] Writing a 0 to this bit. **NOTE:** When RSPERR has been set, the command sequence is stopped automatically.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write 1.
4	CRCSTO	CRC Status Timeout [Setting condition] CRC status could not be received. [Clearing condition] Writing a 0 to this bit. **NOTE:** The command sequence is not stopped even if CRCSTO is set.
3	WDATTO	Write Data Timeout [Setting condition] The busy status remains unchanged after the period set by SRWDTO in MMCAnCE_CLK_CTRL after the CRC status was received. [Clearing condition] Writing a 0 to this bit. **NOTE:** The command sequence is not stopped even if WDATTO is set.

Table 18.25 MMCAnCE_INT Register Contents

Bit Position	Bit Name	Function
2	RDATTO	Read Data Timeout [Setting conditions] – Other than in boot operations ● Read data could not be received within the period set by SRWDTO in MMCAnCE_CLK_CTRL after the read command was transmitted. ● Read data could not be received within the period set by SRWDTO in MMCAnCE_CLK_CTRL after the read data was received. – In boot operations ● The first read data could not be received within the period set by S1STBTDATTO in MMCAnCE_BOOT. ● Read data could not be received within the period set by SBTDATTO in MMCAnCE_BOOT after the read data was received. [Clearing condition] Writing a 0 to this bit. **NOTE:** The command sequence is not stopped even if RDATTO is set.
1	RBSYTO	Response Busy Timeout [Setting condition] The busy status remains unchanged after the period set by SRBSYTO in MMCAnCE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. [Clearing condition] Writing a 0 to this bit. **NOTE:** The command sequence is not stopped even if RBSYTO is set.
0	RSPTO	Response Timeout [Setting conditions] – Other than in boot operations Response could not be received within the period set by SRSPTO in MMCAnCE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. – In boot operations When reception of boot acknowledge has been selected: The boot acknowledge could not be received within the period set by SBTACKTO in MMCAnCE_BOOT. [Clearing condition] Writing a 0 to this bit. **NOTE:** The command sequence is not stopped even if RSPTO is set.

18.3.14 MMCAnCE_INT_EN — MMCAn Interrupt Enable Register

MMCAnCE_INT_EN controls output of the MMCAnCE_INT-related interrupt signals. For details on interrupt requests, see **Section 18.4.1, Interrupt Requests**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 44_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MCMD 12DRE	MCMD 12RBE	MCMD 12CRE	MDTRA NE	MBUF RE	MBUF WEN	MBUF REN	—	—	MRB SYEE	MCR SPEE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCMD VIO	MBUF VIO	—	—	MWDA TERR	MRDAT ERR	MRIDX ERR	MRSP ERR	—	—	—	MCRC STO	MWDA TTO	MRDAT TO	MRBSY TO	MRSP TO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 18.26 MMCAnCE_INT_EN Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	MCMD12DRE	CMD12DRE Interrupt Enable 0: Disables interrupt output by the CMD12DRE flag. 1: Enables interrupt output by the CMD12DRE flag.
25	MCMD12RBE	CMD12RBE Interrupt Enable 0: Disables interrupt output by the CMD12RBE flag. 1: Enables interrupt output by the CMD12RBE flag.
24	MCMD12CRE	CMD12CRE Interrupt Enable 0: Disables interrupt output by the CMD12CRE flag. 1: Enables interrupt output by the CMD12CRE flag.
23	MDTRANE	DTRANE Interrupt Enable 0: Disables interrupt output by the DTRANE flag. 1: Enables interrupt output by the DTRANE flag.
22	MBUFRE	BUFRE Interrupt Enable 0: Disables interrupt output by the BUFRE flag. 1: Enables interrupt output by the BUFRE flag.
21	MBUFWEN	BUFWEN Interrupt Enable 0: Disables interrupt output by the BUFWEN flag. 1: Enables interrupt output by the BUFWEN flag.
20	MBUFREN	BUFREN Interrupt Enable 0: Disables interrupt output by the BUFREN flag. 1: Enables interrupt output by the BUFREN flag.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MRBSYE	RBSYE Interrupt Enable 0: Disables interrupt output by the RBSYE flag. 1: Enables interrupt output by the RBSYE flag.
16	MCRSPE	CRSPE Interrupt Enable 0: Disables interrupt output by the CRSPE flag. 1: Enables interrupt output by the CRSPE flag.

Table 18.26 MMCAnCE_INT_EN Register Contents

Bit Position	Bit Name	Function
15	MCMDVIO	CMDVIO Interrupt Enable 0: Disables interrupt output by the CMDVIO flag. 1: Enables interrupt output by the CMDVIO flag.
14	MBUFVIO	BUFVIO Interrupt Enable 0: Disables interrupt output by the BUFVIO flag. 1: Enables interrupt output by the BUFVIO flag.
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MWDATERR	WDATERR Interrupt Enable 0: Disables interrupt output by the WDATERR flag. 1: Enables interrupt output by the WDATERR flag.
10	MRDATERR	RDATERR Interrupt Enable 0: Disables interrupt output by the RDATERR flag. 1: Enables interrupt output by the RDATERR flag.
9	MRIDXERR	RIDXERR Interrupt Enable 0: Disables interrupt output by the RIDXERR flag. 1: Enables interrupt output by the RIDXERR flag.
8	MRSPERR	RSPERR Interrupt Enable 0: Disables interrupt output by the RSPERR flag. 1: Enables interrupt output by the RSPERR flag.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	MCRCSTO	CRCSTO Interrupt Enable 0: Disables interrupt output by the CRCSTO flag. 1: Enables interrupt output by the CRCSTO flag.
3	MWDATTO	WDATTO Interrupt Enable 0: Disables interrupt output by the WDATTO flag. 1: Enables interrupt output by the WDATTO flag.
2	MRDATTO	RDATTO Interrupt Enable 0: Disables interrupt output by the RDATTO flag. 1: Enables interrupt output by the RDATTO flag.
1	MRBSYTO	RBSYTO Interrupt Enable 0: Disables interrupt output by the RBSYTO flag. 1: Enables interrupt output by the RBSYTO flag.
0	MRSPTO	RSPTO Interrupt Enable 0: Disables interrupt output by the RSPTO flag. 1: Enables interrupt output by the RSPTO flag.

18.3.15 MMCAnCE_HOST_STS1 — MMCAn Status Register 1

MMCAnCE_HOST_STS1 indicates the number of blocks that have been transferred, the states of the MMCAnCMD line and MMCAnDAT lines, the index of the response that has been received, and whether a command sequence is in progress.

Access: This register can be read only in 32-bit units.

Address: <MMCAn_base> + 48_H

Value after reset: X0XX 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD SEQ	CMD SIG	RSPIDX[5:0]						DATSIG[7:0]							
Value after reset	0	—	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCVBLK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.27 MMCAnCE_HOST_STS1 Register Contents

Bit Position	Bit Name	Function
31	CMDSEQ	Command Sequence in Progress 0: Command sequence is in the initial state. 1: Command sequence is being executed.
30	CMDSIG	MMCAnCMD State Indicates the state on the MMCAnCMD line.
29 to 24	RSPIDX[5:0]	Response Index Indicate bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response.
23 to 16	DATSIG[7:0]	MMCAnDAT[7:0] State Indicate the state on the MMCAnDAT[7:0] lines. Note: When a communication error or a timeout error occurs, MMCAnDAT[0] may remain 0.
15 to 0	RCVBLK[15:0]	Number of Transferred Blocks Indicate the number of blocks that have been transferred. When the DWEN bit in MMCAnCE_CMD_SET is 0: Number of blocks read from the card When the DWEN bit in MMCAnCE_CMD_SET is 1: Number of blocks written to the card

18.3.16 MMCAnCE_HOST_STS2 — MMCAn Status Register 2

MMCAnCE_HOST_STS2 indicates timeout and error statuses.

Access: This register can be read only in 32-bit units.

Address: <MMCAn_base> + $4C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRCSTE	CRC16E	AC12CRCE	RSPCRC7E	CRCSTEBE	RDATEBE	AC12REBE	RSPEBE	AC12IDXE	RSPIDXE	BTACKPATE	BTACKEBE	—	CRCST[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	STRDATTO	DATBSYTO	CRCSTTO	AC12BSYTO	RSPBSYTO	AC12RSPTO	STRSPTO	BTACKTO	1STBTDATTO	BTDATTO	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.28 MMCAnCE_HOST_STS2 Register Contents

Bit Position	Bit Name	Function
31	CRCSTE	CRC Status Error This bit is set to 1 when an error is found in the CRC status value.
30	CRC16E	Read Data CRC16 Error This bit is set to 1 when an error is found in CRC16 in the read data.
29	AC12CRCE	Automatic CMD12 Response CRC7 Error This bit is set to 1 when an error is found in bits [7:1] of the response to the automatically-issued CMD12. **NOTE:** The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.
28	RSPCRC7E	Command Response CRC7 Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in bits [7:1] of a 6-byte response or a 17-byte response. **NOTE:** The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.
27	CRCSTEBE	CRC Status End Bit Error This bit is set to 1 when an error is found in the end bits in CRC status.
26	RDATEBE	Read Data End Bit Error This bit is set to 1 when an error is found in the end bits in the read data.
25	AC12REBE	Automatic CMD12 Response End Bit Error This bit is set to 1 when an error is found in the end bits of the response to the automatically-issued CMD12.
24	RSPEBE	Command Response End Bit Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in the end bits of the response.
23	AC12IDXE	Automatic CMD12 Response Index Error This bit is set to 1 when an error is found in bits [45:40] of the response to the automatically-issued CMD12. **NOTE:** The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.
22	RSPIDXE	Command Response Index Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response. **NOTE:** The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.

Table 18.28 MMCAnCE_HOST_STS2 Register Contents

Bit Position	Bit Name	Function
21	BTACKPATE	Boot Acknowledge Pattern Error This bit is set to 1 when an error is found in the boot acknowledge pattern.
20	BTACKEBE	Boot Acknowledge End Bit Error This bit is set to 1 when an error is found in the end bits of the boot acknowledge.
19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	CRCST[2:0]	CRC Status/Boot Acknowledge Pattern Indication This bit indicates the value for CRC status that was received or the pattern value from the boot-acknowledge.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	STRDATTO	Read Data Timeout (valid other than in boot operations) This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after a read command was transmitted. This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after a read data was received.
13	DATBSYTO	Data Busy Timeout This bit is set to 1 if busy status remains unchanged after the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after the CRC status was received.
12	CRCSTTO	CRC Status Timeout This bit is set to 1 if CRC status could not be received.
11	AC12BSYTO	Automatic CMD12 Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
10	RSPBSYTO	Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
9	AC12RSPTO	Automatic CMD12 Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in MMCAnCE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
8	STRSPTO	Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in MMCAnCE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
7	BTACKTO	Boot Acknowledge Timeout In boot operations, this bit is set to 1 if boot acknowledge is not received within the period set by the SBTACKTO bits in MMCAnCE_BOOT.
6	1STBTDATTO	1st Boot Data Timeout In boot operations, this bit is set to 1 if the 1st read data is not received within the period set by the S1STBTDATTO bits in MMCAnCE_BOOT.
5	BTDATTO	Interval between Boot Data Timeout In boot operations, this bit is set to 1 if read data is not received within the period set by the SBTDATTO bits in MMCAnCE_BOOT after a read data was received.
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.17 MMCAnCE_SWRESA — MMCAn Software Reset Register

MMCAnCE_SWRESA controls software reset of this module.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $7C_H$

Value after reset: $0000\ 0004_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SWRST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.29 MMCAnCE_SWRESA Register Contents

Bit Position	Bit Name	Function
31	SWRST	Software Reset 0: Software reset cleared (normal operation). 1: Executes software reset. When this bit is set to 1, the values of all registers are initialized. (SWRST is not initialized.)
30 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.4 Operations

18.4.1 Interrupt Requests

The table below shows the specification of the interrupt requests. This module generates two types of interrupt requests: normal operation and error/timeout. When a bit in the flag register is set to 1 and also the corresponding bit in the interrupt enable register is set to 1 (enabled), an interrupt request is asserted.

Table 18.30 Specification of Interrupt Requests

Flag Register	Bit	Enable Register	Bit	Interrupt Request
MMCAnCE_INT	CMD12DRE	MMCAnCE_INT_EN	MCMD12DRE	Normal operation interrupt
	CMD12RBE		MCMD12RBE	
	CMD12CRE		MCMD12CRE	
	DTRANE		MDTRANE	
	BUFRE		MBUFRE	
	BUFWEN		MBUFWEN	
	BUFREN		MBUFREN	
	RBSYE		MRBSYE	
	CRSPE		MCRSPE	
	CMDVIO		MCMDVIO	Error/timeout interrupt
	BUFVIO		MBUFVIO	
	WDATERR		MWDATERR	
	RDATERR		MRDATERR	
	RIDXERR		MRIDXERR	
	RSPERR		MRSPERR	
	CRCSTO		MCRCSTO	
	WDATTO		MWDATTO	
	RDATTO		MRDATTO	
	RBSYTO		MRBSYTO	
	RSPTO		MRSPTO	

18.4.2 DMA Specifications

This module has two types of DMA transfer requests: for buffer reading and for buffer writing.

The method of DMA transfer is configured by MMCAnCE_BUF_ACC.

18.4.2.1 DMA for Buffer Writing

The DMA transfer request is asserted for buffer writing when the buffer has become empty while the DMAWEN bit in MMCAnCE_BUF_ACC is set to 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in MMCAnCE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in MMCAnCE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFWEN bit in MMCAnCE_INT will not be asserted during DMA transfer.

If an error has occurred during DMA transfer or DMA transfer is forcibly terminated, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

18.4.2.2 DMA for Buffer Reading

The DMA transfer request is asserted for buffer reading when the buffer stores data of the block size specified in MMCAnCE_BLOCK_SET while the DMAREN bit in MMCAnCE_BUF_ACC is set to 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in MMCAnCE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in MMCAnCE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFREN bit in MMCAnCE_INT will not be asserted during DMA transfer.

If an error has occurred during DMA transfer or DMA transfer is forcibly terminated, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

18.4.3 Operations

18.4.3.1 Command/Response Formats

The figure below shows the format of the command to be transferred. The command index that is set in the CMD[5:0] bits in MMCAnCE_CMD_SET and the argument set in the ARG[31:0] bits in MMCAnCE_ARG are reflected in the command.

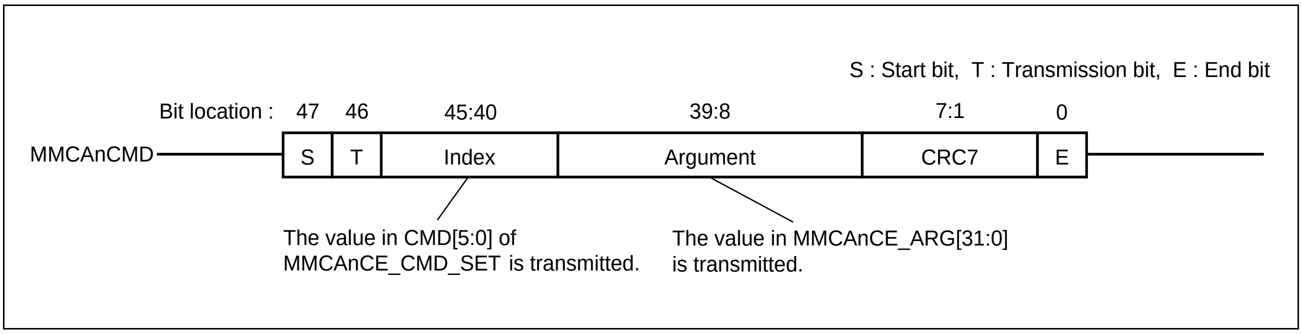


Figure 18.2 Command Format

Figure 18.3, Format of 6-Byte Response and **Figure 18.4, Format of 17-Byte Response (R2)** show the formats when a 6-byte response and 17-byte response (R2) are received, respectively. The received response is stored in MMCAnCE_RESP0 or MMCAnCE_RESP3 to MMCAnCE_RESP0. The items to be checked are set by the RIDXC bits and the RCRC7C bits in MMCAnCE_CMD_SET.

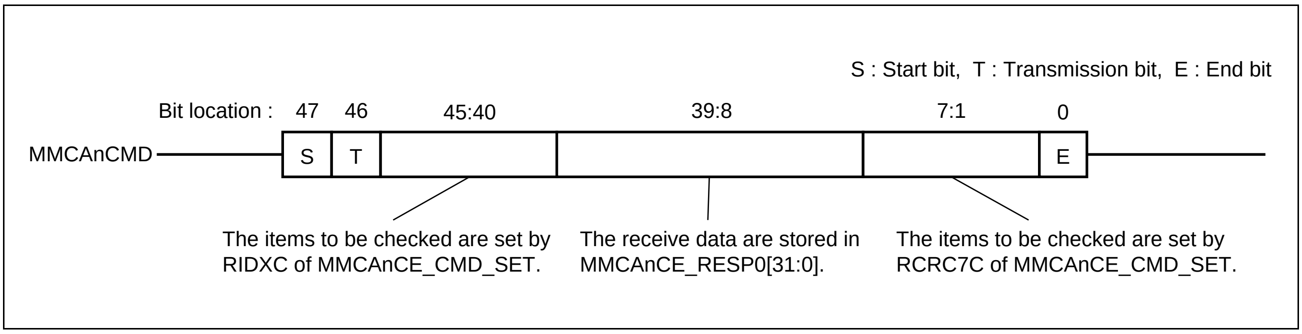


Figure 18.3 Format of 6-Byte Response

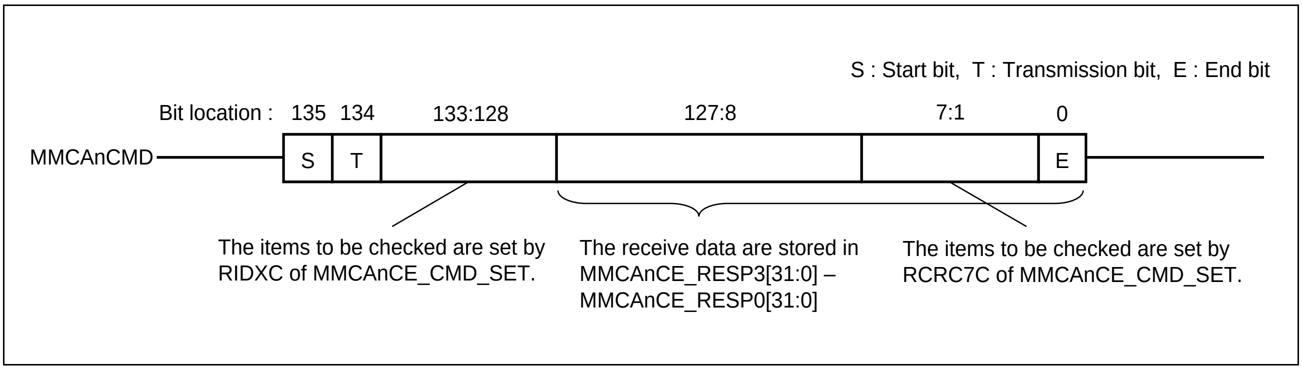


Figure 18.4 Format of 17-Byte Response (R2)

18.4.3.2 Data Block Format

The figure below shows the format of data blocks. For D0 to D3 in the figure, see **Section 18.4.3.3, Buffer Structure and Buffer Access**. When data are written to the card, the data stored in the buffer are transmitted. When data are read from the card, the received data are stored in the buffer.

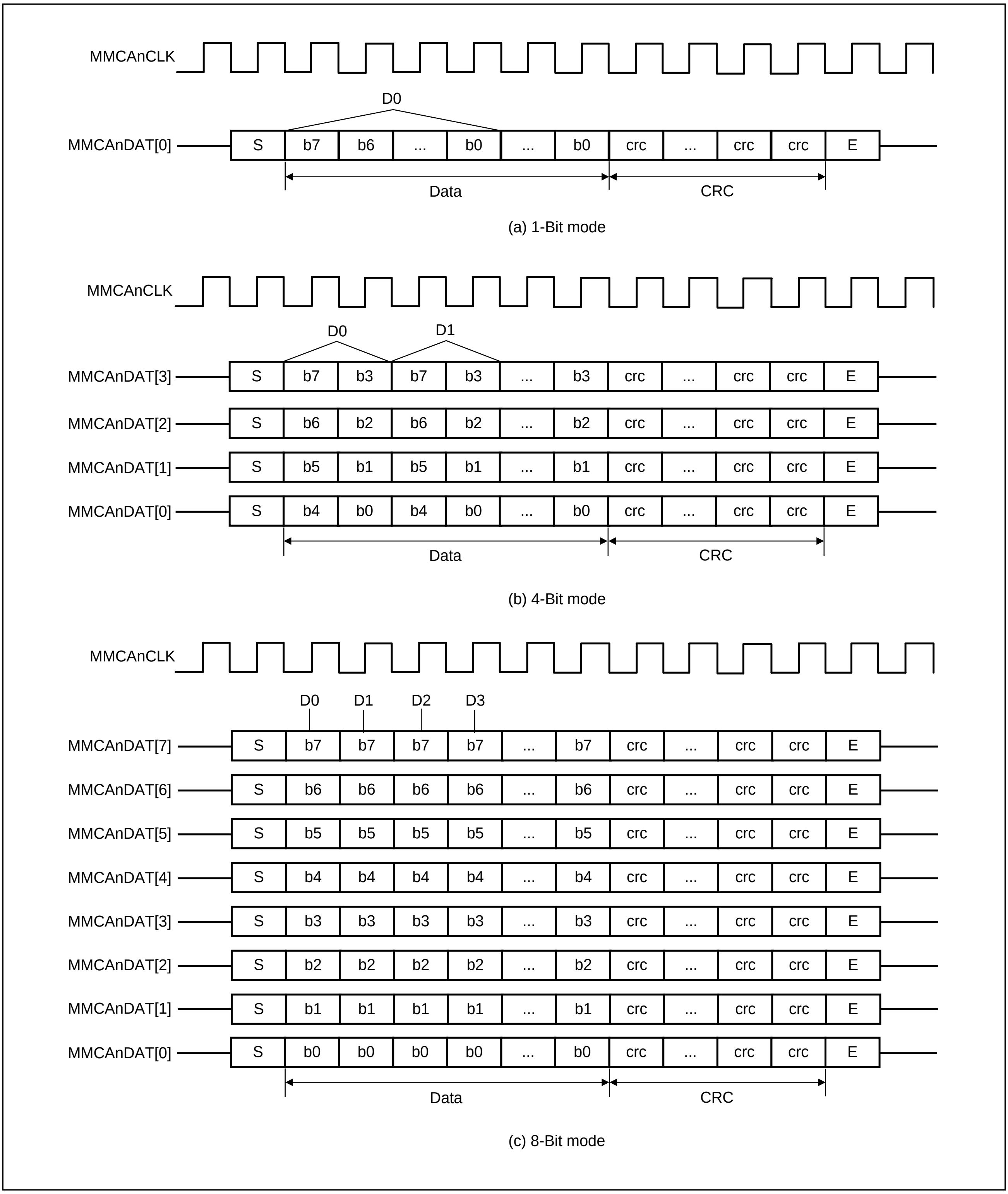


Figure 18.5 Data Block Format

18.4.3.3 Buffer Structure and Buffer Access

As shown in **Figure 18.6, Double Buffer Structure**, this module has two 512-byte RAM units for double buffering. If a block of data (512 bytes) stored in one buffer is transmitted during multi-block writing, the next block of data can be transmitted as soon as the other buffer is full. If a block of received data (512 bytes) is stored in one buffer during multi-block reading, the next block of received data can be stored in the other buffer once it is empty.

If neither buffer is empty during multi-block reading, the MMC clock is stopped, which in turn temporarily stops reception. Once either of the buffers becomes empty, supply of the MMC clock signal is re-started and data reception is resumed.

The buffers are accessed by MMCAnCE_DATA. If the transfer block size is set to 4 × n + 1 or 4 × n + 3, access should be made for 4 × (n + 1) bytes in 32-bit access. (n = 0, 1, 2, …, 127).

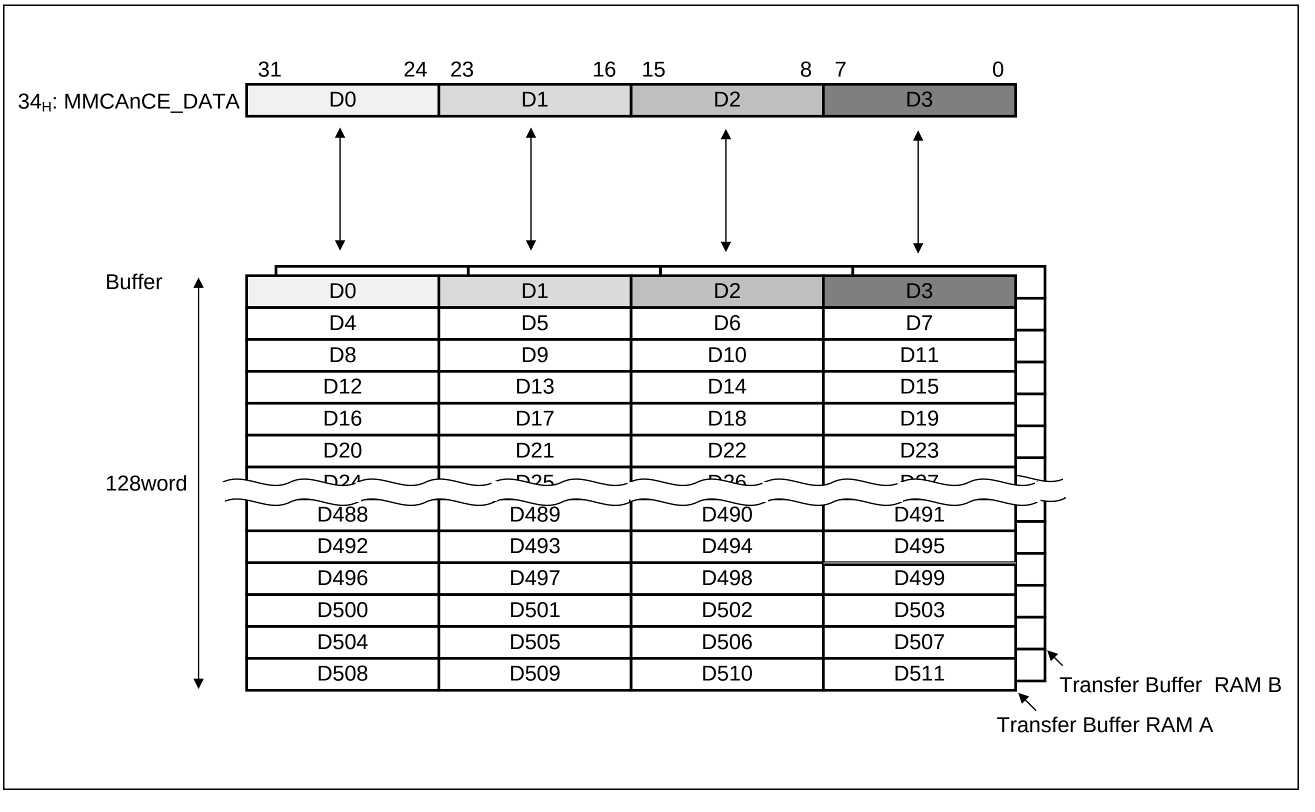


Figure 18.6 Double Buffer Structure

This module has the buffer access select function that allows byte-wise swapping of data when the buffer is accessed by writing to or reading from MMCAnCE_DATA. This function is enabled by the setting of MMCAnCE_BUFF_ACC. The figure below shows the specification of 32-bit accesses.

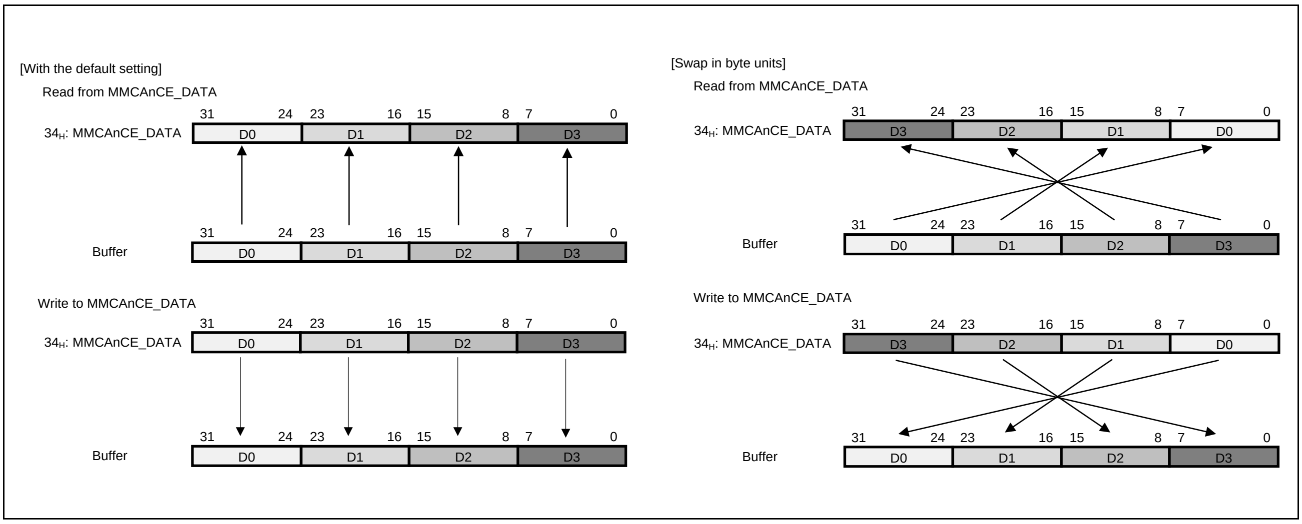


Figure 18.7 Specification of Byte-Swapping

18.4.3.4 Automatic CMD12 Issuance

This module has the function that automatically issues CMD12 when multi-block transfer is performed with the CMD12EN bit in MMCAnCE_CMD_SET set to 1.

The figure below shows the timing of automatic CMD12 issuance in multi-block read. CMD12 is issued such that the end bit of the command is sent two bits before the end bit of the data during reception of the last block.

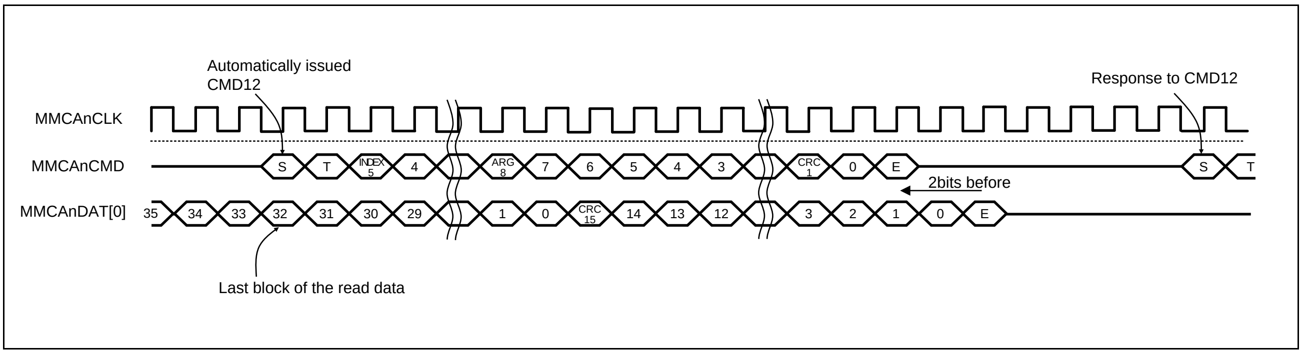


Figure 18.8 Timing of Automatically-Issued CMD12 in Multi-Block Read (1-Bit Mode)

The figure below shows the timing of automatic CMD12 issuance in multi-block write. CMD12 is issued after the data busy after transmission of the last block has ended.

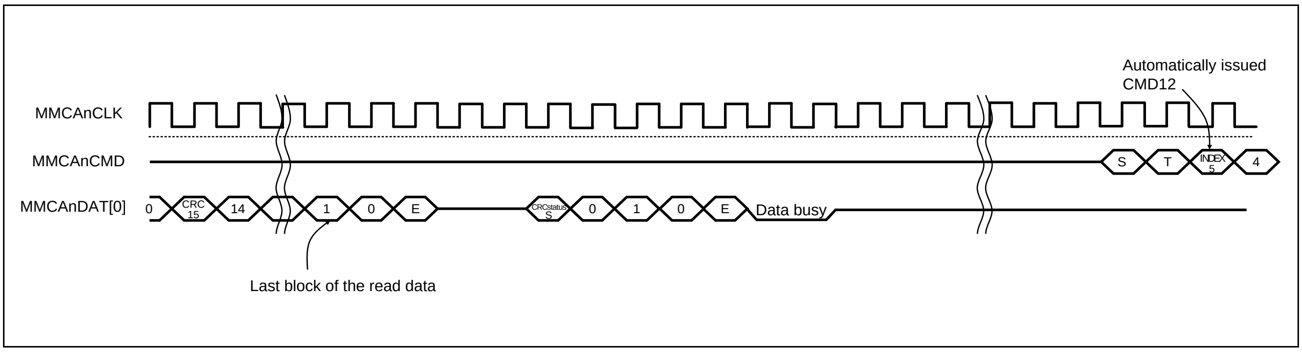


Figure 18.9 Timing of Automatically-Issued CMD12 in Multi-Block Write (1-Bit Mode)

The argument for the automatically-issued CMD12 is set in MMCAnCE_ARG_CMD12. Bits [39:8] of the response to the CMD12 is stored in MMCAnCE_RESP_CMD12. "Busy" is involved in response reception.

18.4.3.5 MMC Clock Frequency in Boot Operations

The figure below shows the timing for changing the MMC clock frequency in boot operations. In boot operations, the MMC clock frequency can be switched to the value corresponding to the setting of the BTCLKDIV bits of MMCAnCE_BOOT 74 MMC clock cycles after MMCAnCMD is modified from 1 to 0. Alternatively, it can be switched to the value corresponding to the setting of the CLKDIV bits of MMCAnCE_CLK_CTRL 48 MMC clock cycles after MMCAnCMD is modified from 0 to 1.

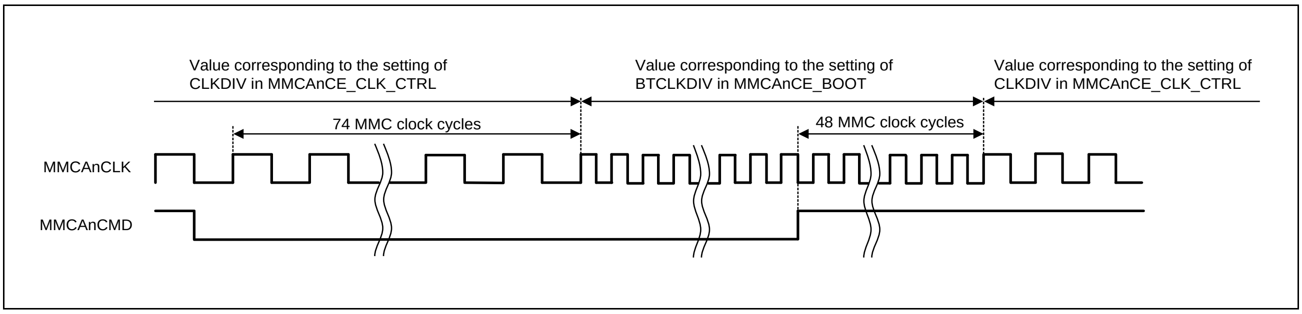


Figure 18.10 MMC Clock Frequency in Boot Operations

18.4.3.6 High Priority Interrupt (HPI)

The HPI should be processed according to the following procedure. See **Section 18.3, Registers**, and **Section 18.4.4, Examples of Setting**.

(a) To execute the HPI during a write to the device

- Terminate the command sequence forcibly.
- Wait until the CMDSEQ bit in MMCAnCE_HOST_STS1 becomes 1.
- Issue CMD12 (R1) to cause a device state transition from rcv to prg. If the device is already in the prg state here, the device does not output a response.
- Issue CMD13 (R1).
- Issue the HPI command.*[1]

(b) To execute the HPI in the response busy state not during a write to the device

- Terminate the command sequence forcibly.
- Wait until the CMDSEQ bit in MMCAnCE_HOST_STS1 turns 0.
- Issue CMD13 (R1).
- Issue the HPI command.*[1]

HPIs in the sequence of CMD6, CMD24, CMD25 (pre-defined), and CMD38 are supported.

Note 1. CMD12 (R1b) or CMD13 (R1b) differs depending on the e-MMC connected.

18.4.3.7 Background Operation

The background operation should be processed according to the following procedure. See **Section 18.3, Registers** and **Section 18.4.4, Examples of Setting**.

To execute background operation, issue CMD6 (R1) to write to the BKOPS_START byte in the EXT_CSD register of the device.

To confirm that background operation has been completed, issue CMD6 (R1) and then CMD13 (R1) to check the device state, or poll the MMCAnDAT[0]. If the device state is trans after issuing CMD13 (R1) or MMCAnDAT[0] is high, background operation has been completed.

To suspend background operation, use the HPI (see **Section 18.4.3.6, High Priority Interrupt (HPI)**).

18.4.3.8 Handling of this Module in the Case of Error/Timeout

This module may not be stopped when an error has occurred. If an error has occurred and the CMDSEQ bit in MMCAnCE_HOST_STS1 is still indicating that the command sequence is in progress, terminate the sequence forcibly and execute software reset. Note that the data for transmission or received data that had been stored in the buffers at the time of error occurrence are not guaranteed.

This module is not stopped when a timeout has occurred. Before issuing the next command after the timeout has occurred, terminate the sequence forcibly and execute software reset.

18.4.4 Examples of Setting

This section shows the procedures for executing typical command sequences.

18.4.4.1 Legends

The figure below shows the legends for the symbols used in the figures in the following subsections.

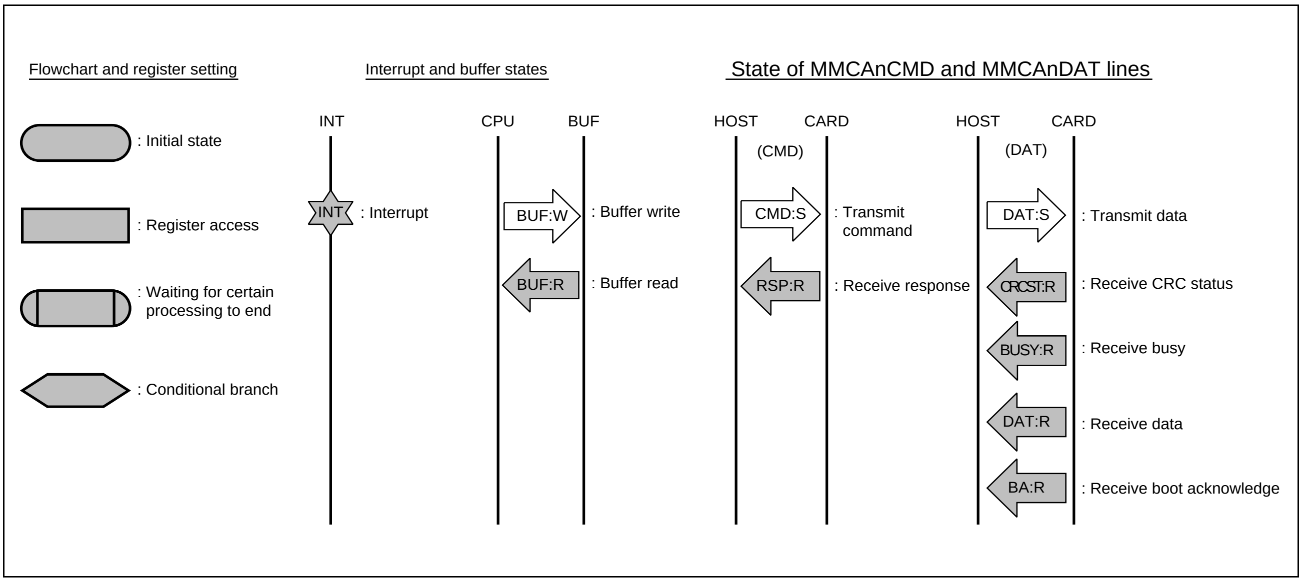
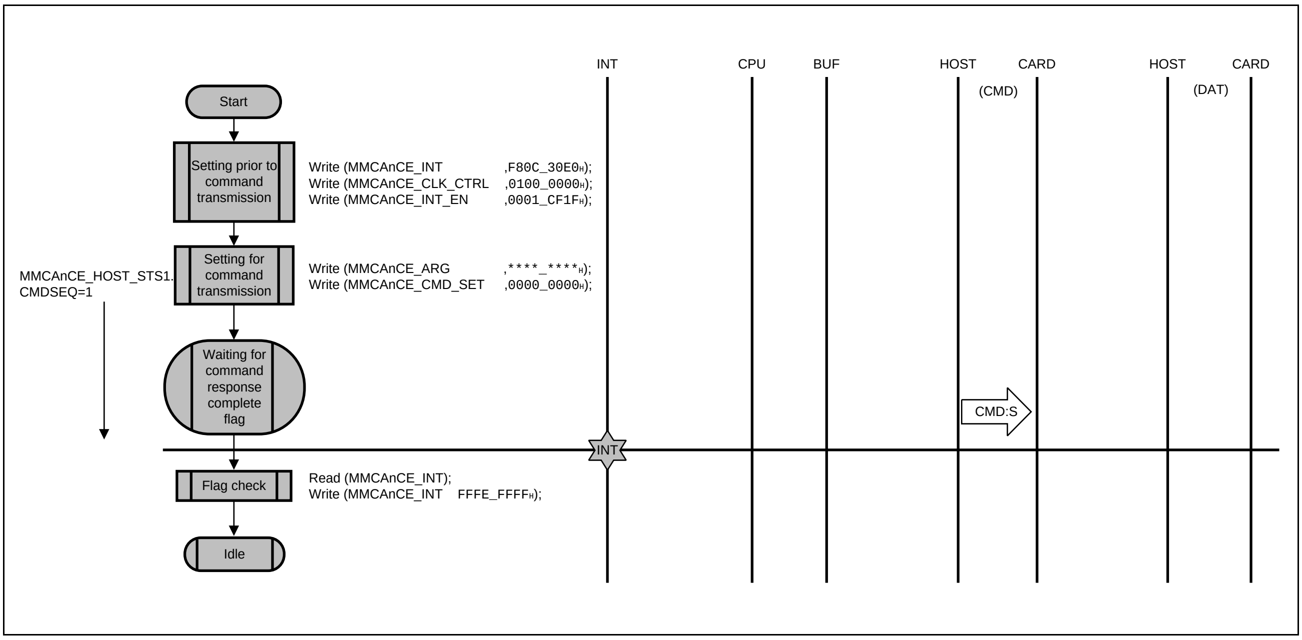


Figure 18.11 Legends for Symbols Used in Figures

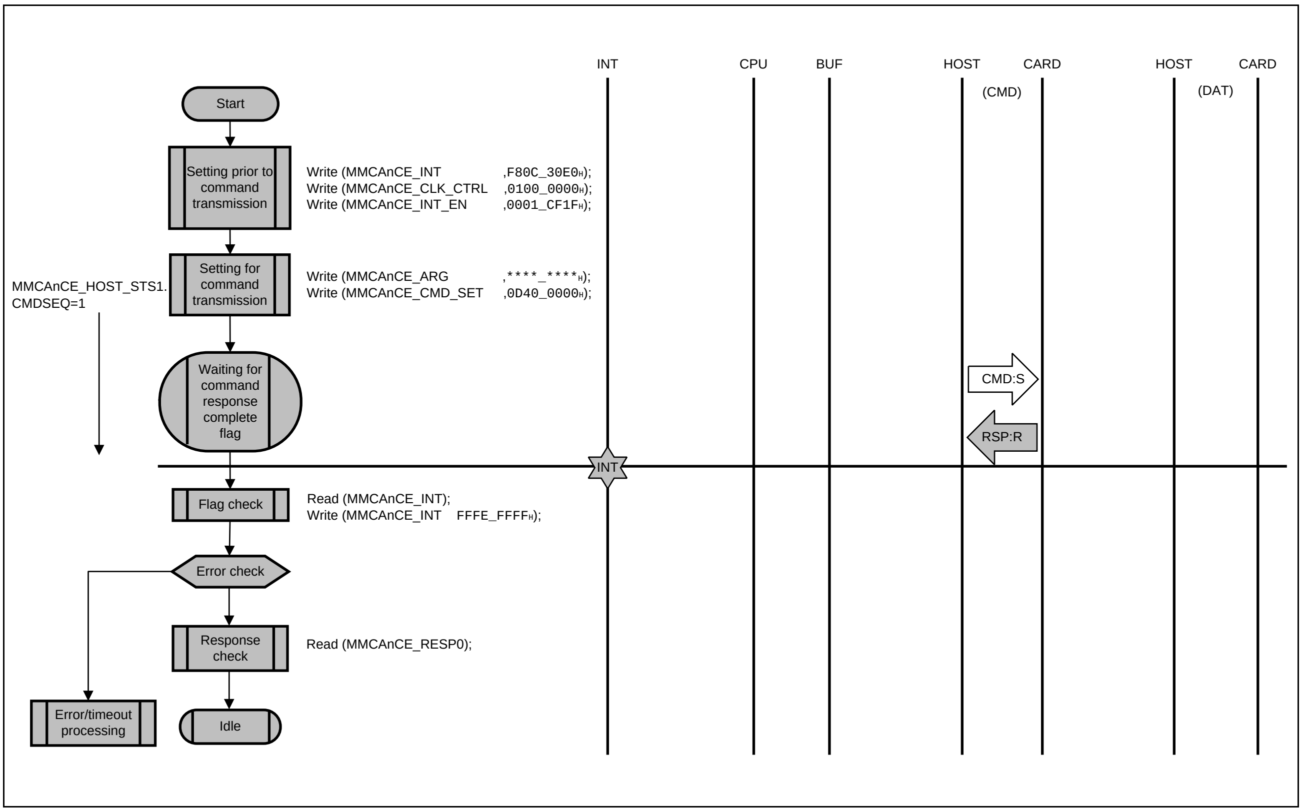
18.4.4.2 Command Transmission



```
Write (MMCAnCE_INT          ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL     ,0100_0000H);
Write (MMCAnCE_INT_EN       ,0001_CF1FH);

Write (MMCAnCE_ARG          ,****_****H);
Write (MMCAnCE_CMD_SET      ,0000_0000H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    FFFE_FFFFH);
```

Figure 18.12 Command Transmission (CMD0)

18.4.4.3 Command Transmission → Response Reception



```
Write (MMCAnCE_INT          ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL     ,0100_0000H);
Write (MMCAnCE_INT_EN       ,0001_CF1FH);
```

```
Write (MMCAnCE_ARG          ,****_****H);
Write (MMCAnCE_CMD_SET      ,0D40_0000H);
```

```
Read (MMCAnCE_INT);
Write (MMCAnCE_INT    FFFE_FFFFH);
```

```
Read (MMCAnCE_RESP0);
```

Figure 18.13 Command Transmission to Response Reception (CMD13)

18.4.4.4 Command Transmission → Response Reception (with Response Busy)

- When the busy time period is less than the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL

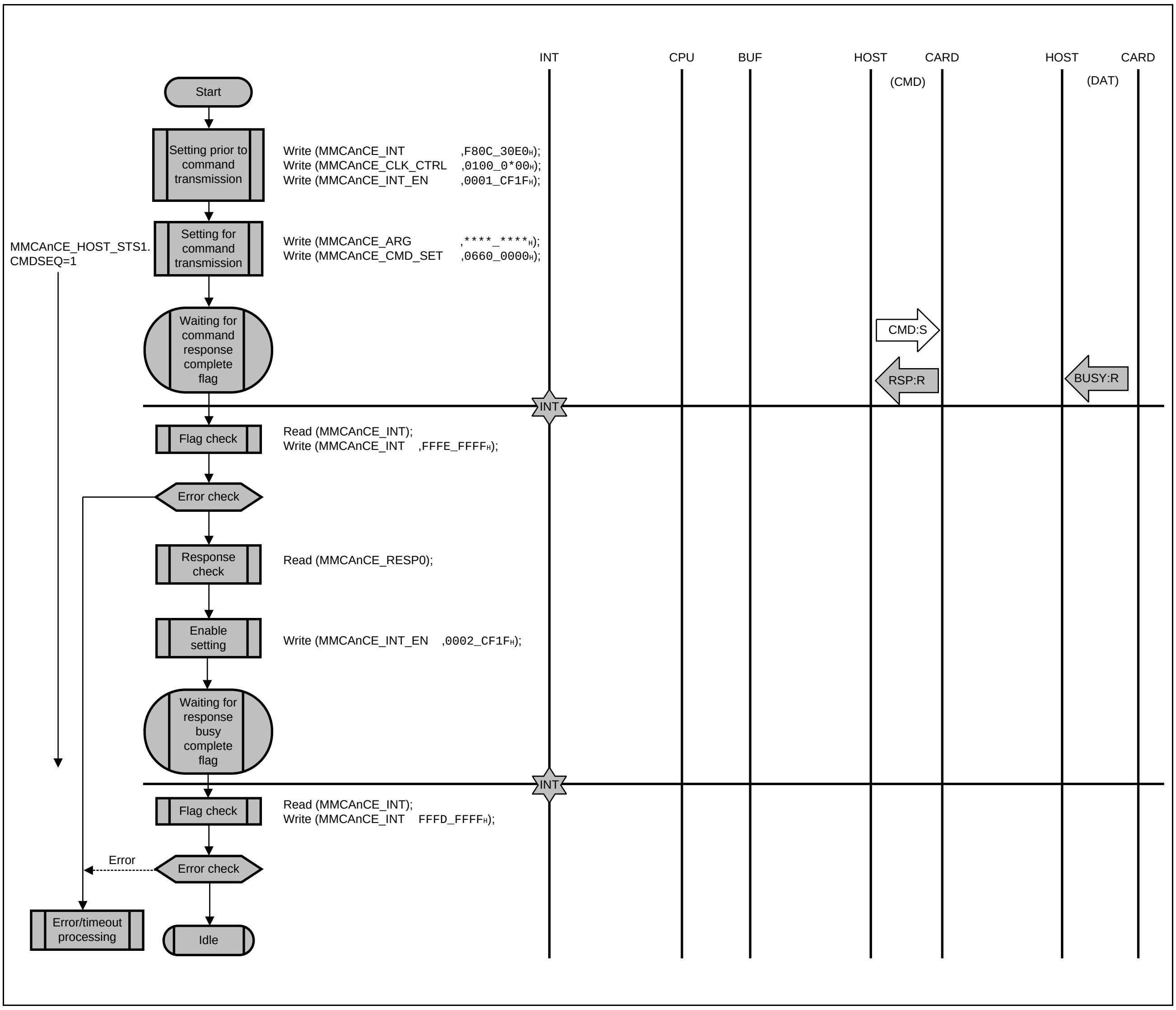


Figure 18.14 Command Transmission to Response Reception (with Response Busy) (CMD6)

- When the busy time period may be equal to or beyond the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL

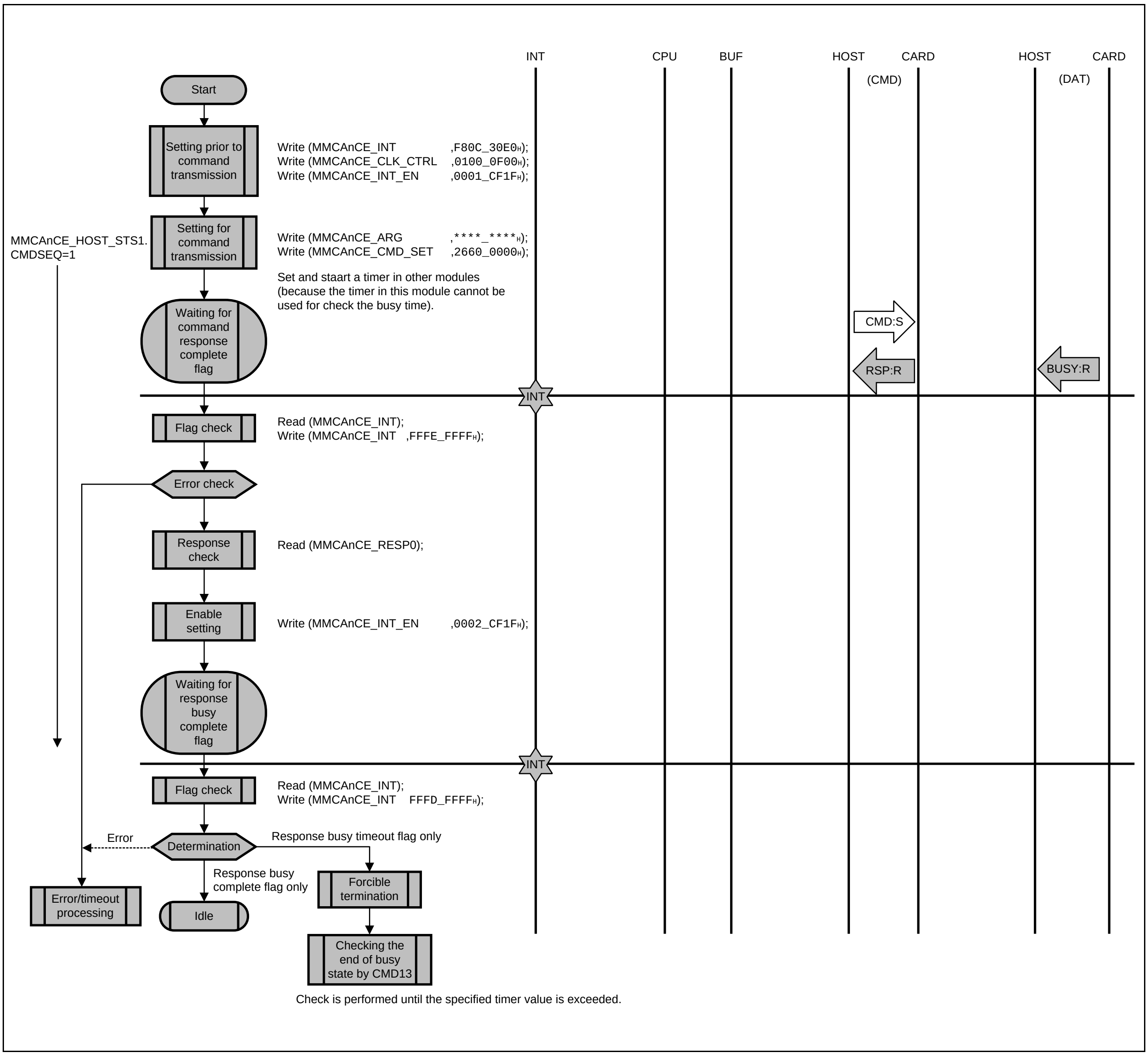


Figure 18.15 Command Transmission to Response Reception (with Response Busy) (CMD38)

18.4.4.5 Single-Block Read

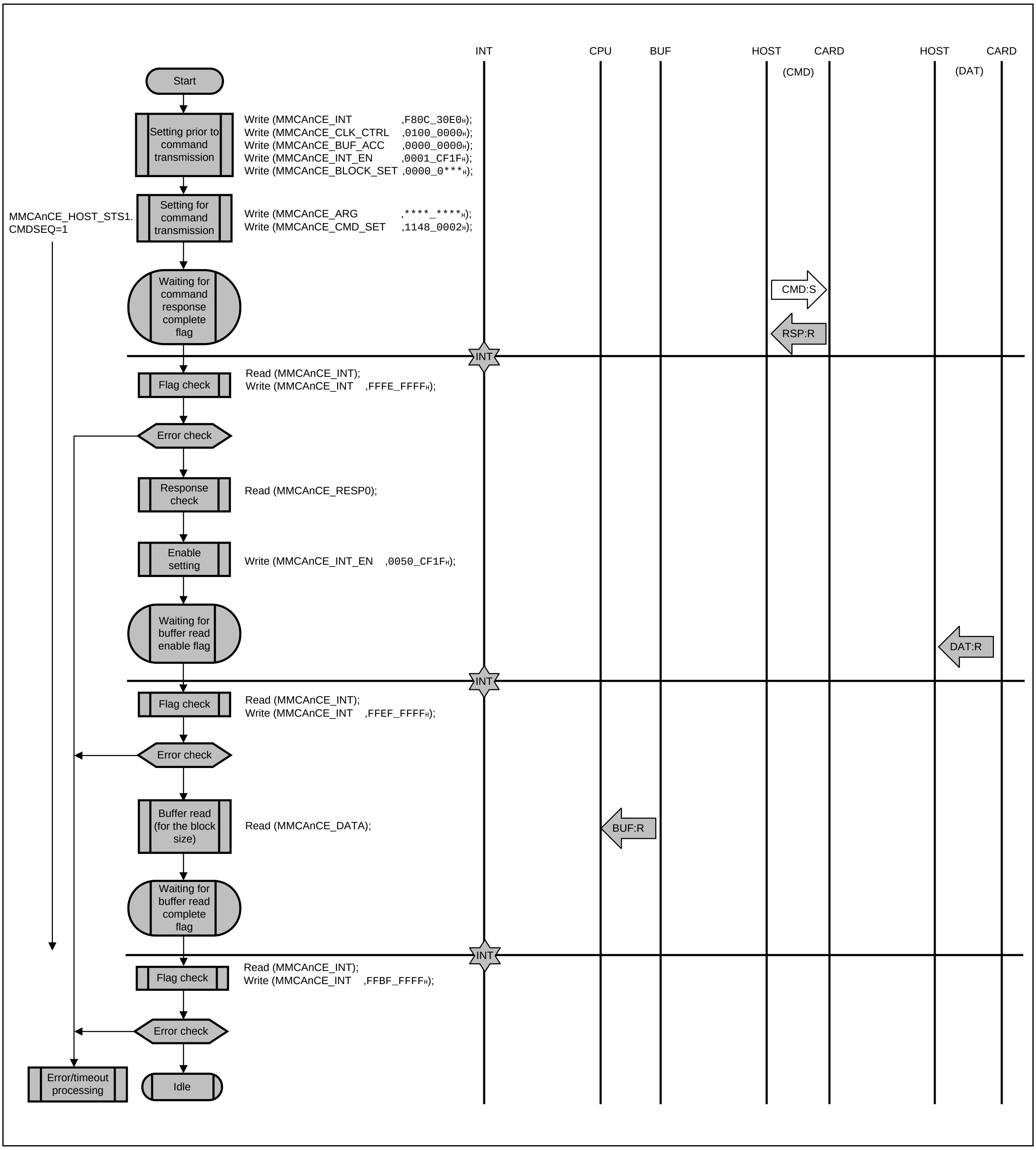
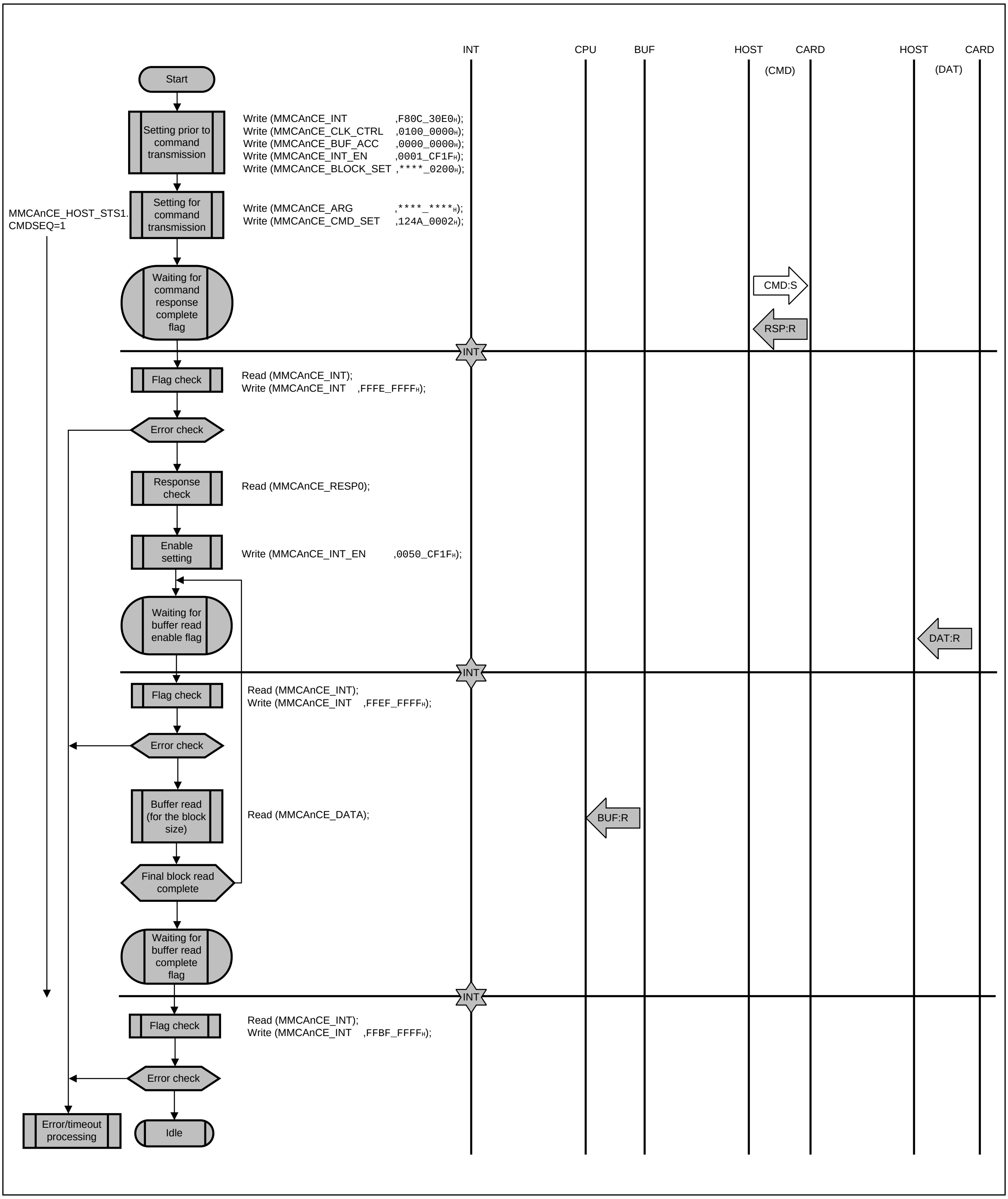


Figure 18.16 Single-Block Read (CMD17)

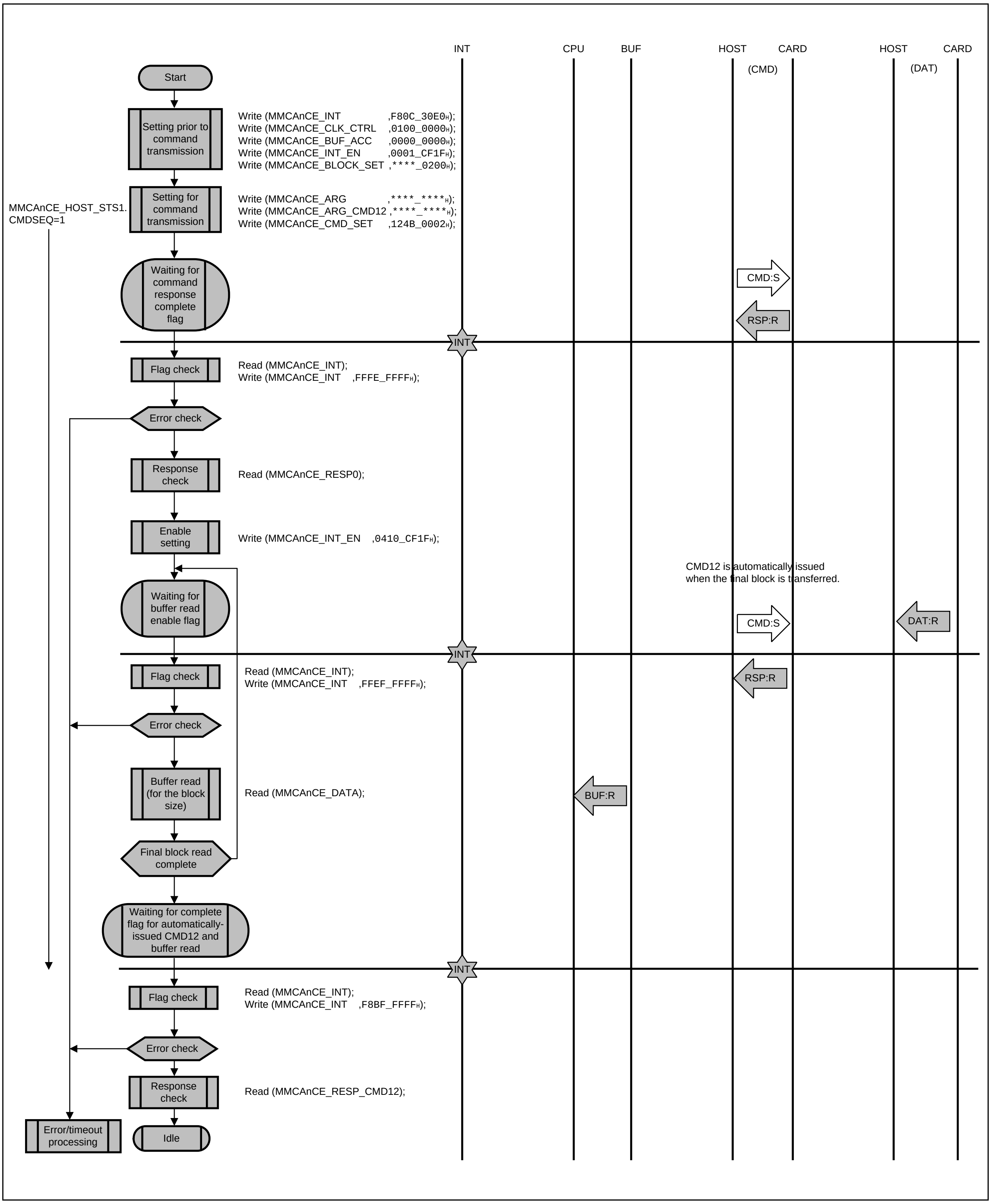
18.4.4.6 Multi-Block Read



```
Write (MMCAnCE_INT          ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL     ,0100_0000H);
Write (MMCAnCE_BUF_ACC      ,0000_0000H);
Write (MMCAnCE_INT_EN       ,0001_CF1FH);
Write (MMCAnCE_BLOCK_SET    ,****_0200H);

Write (MMCAnCE_ARG          ,****_****H);
Write (MMCAnCE_CMD_SET      ,124A_0002H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFFE_FFFFH);

Read (MMCAnCE_RESP0);

Write (MMCAnCE_INT_EN       ,0050_CF1FH);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFEF_FFFFH);

Read (MMCAnCE_DATA);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFBF_FFFFH);
```

Figure 18.17 Multi-Block Read (CMD18 Pre-Defined)

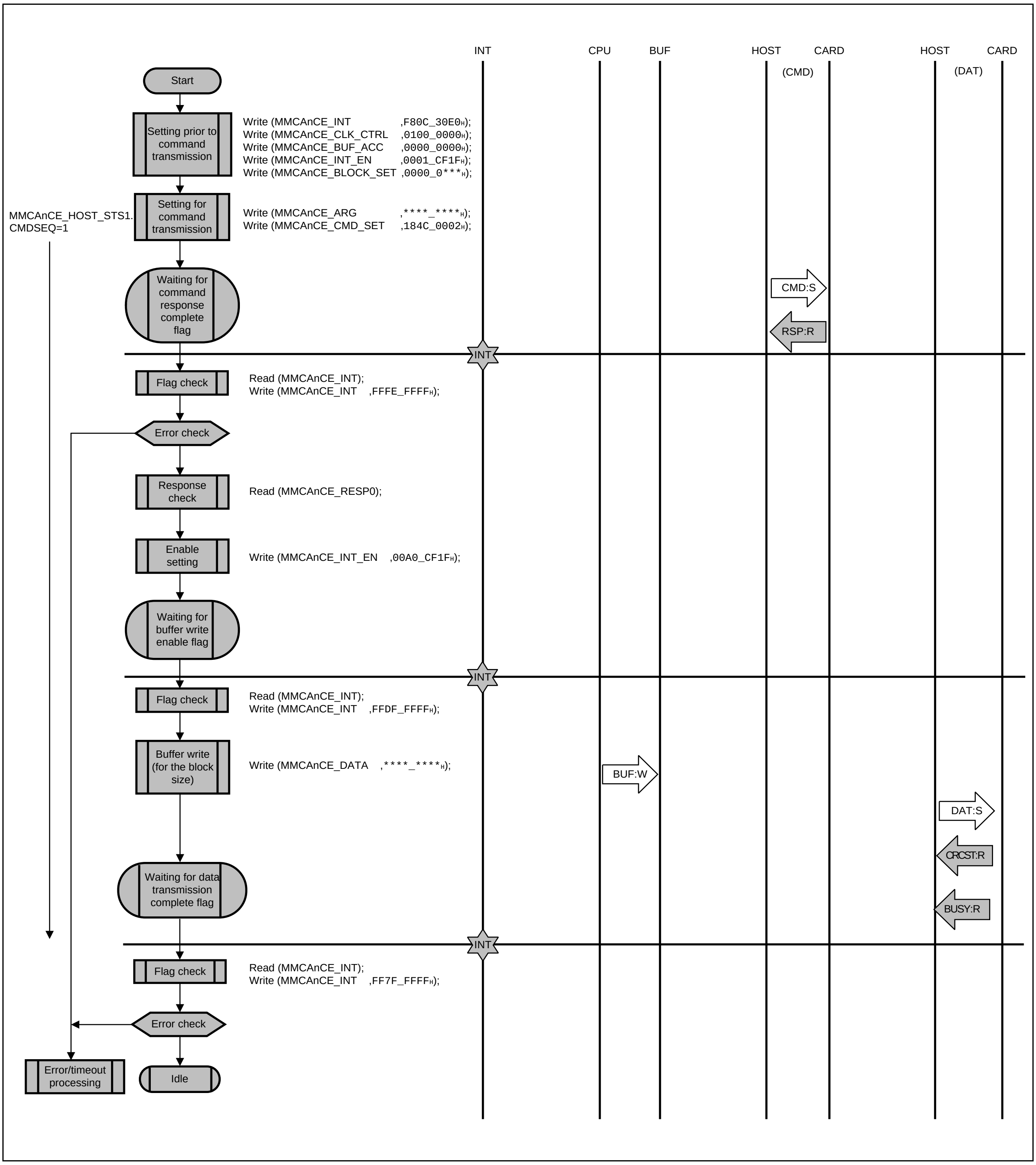
18.4.4.7 Multi-Block Read (with Automatic CMD12 Issuance)



```
Write (MMCAnCE_INT           ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL      ,0100_0000H);
Write (MMCAnCE_BUF_ACC       ,0000_0000H);
Write (MMCAnCE_INT_EN        ,0001_CF1FH);
Write (MMCAnCE_BLOCK_SET     ,****_0200H);

Write (MMCAnCE_ARG           ,****_****H);
Write (MMCAnCE_ARG_CMD12     ,****_****H);
Write (MMCAnCE_CMD_SET       ,124B_0002H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFFE_FFFFH);

Read (MMCAnCE_RESP0);

Write (MMCAnCE_INT_EN    ,0410_CF1FH);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFEF_FFFFH);

Read (MMCAnCE_DATA);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,F8BF_FFFFH);

Read (MMCAnCE_RESP_CMD12);
```

Figure 18.18 Multi-Block Read (with Automatic CMD12 Issuance) (CMD18 Open-Ended)

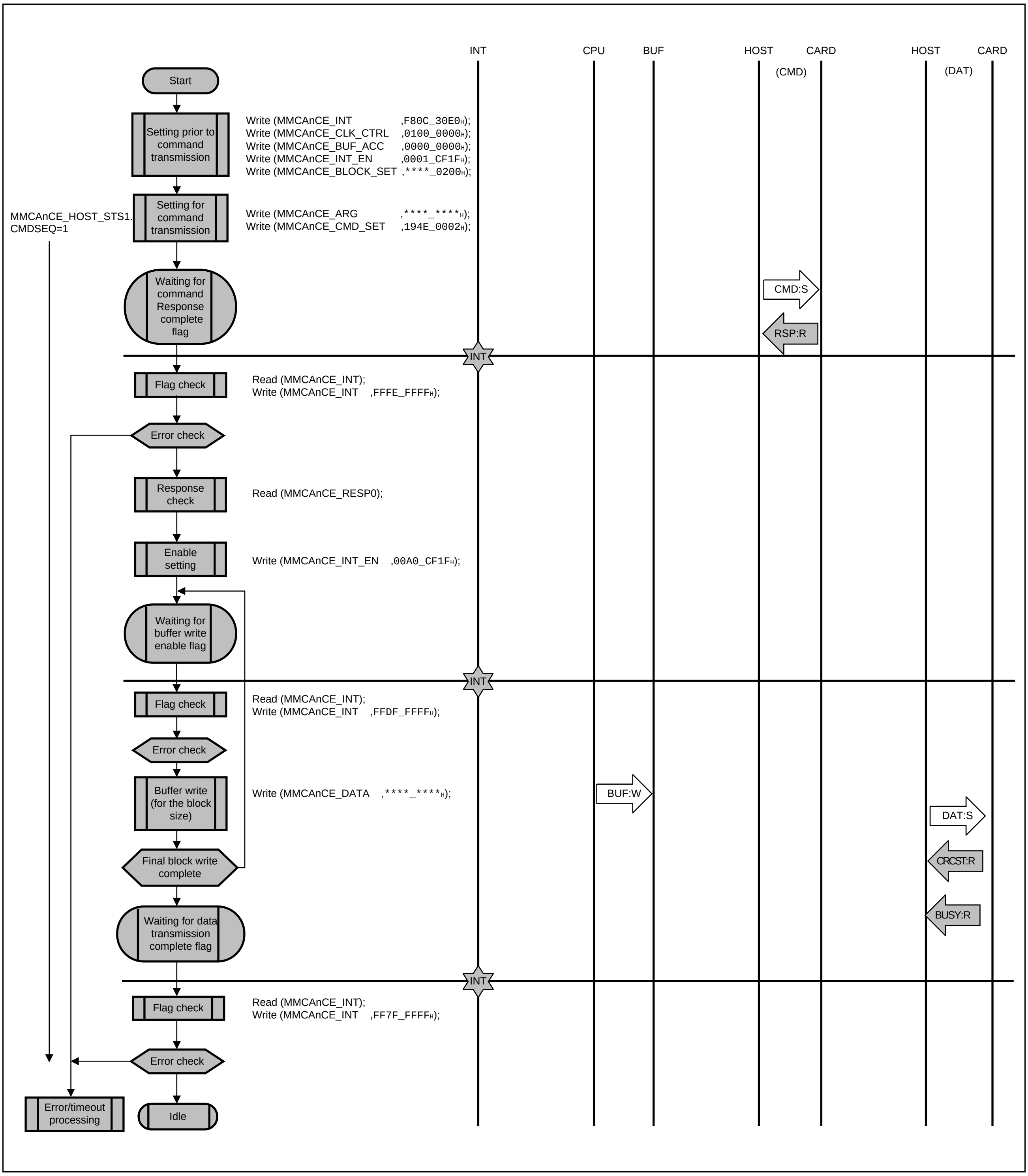
18.4.4.8 Single-Block Write



```
Write (MMCAnCE_INT          ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL     ,0100_0000H);
Write (MMCAnCE_BUF_ACC      ,0000_0000H);
Write (MMCAnCE_INT_EN       ,0001_CF1FH);
Write (MMCAnCE_BLOCK_SET    ,0000_0***H);

Write (MMCAnCE_ARG          ,****_****H);
Write (MMCAnCE_CMD_SET      ,184C_0002H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFFE_FFFFH);

Read (MMCAnCE_RESP0);

Write (MMCAnCE_INT_EN    ,00A0_CF1FH);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FFDF_FFFFH);

Write (MMCAnCE_DATA    ,****_****H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT    ,FF7F_FFFFH);
```

Figure 18.19 Single-Block Write (CMD24)

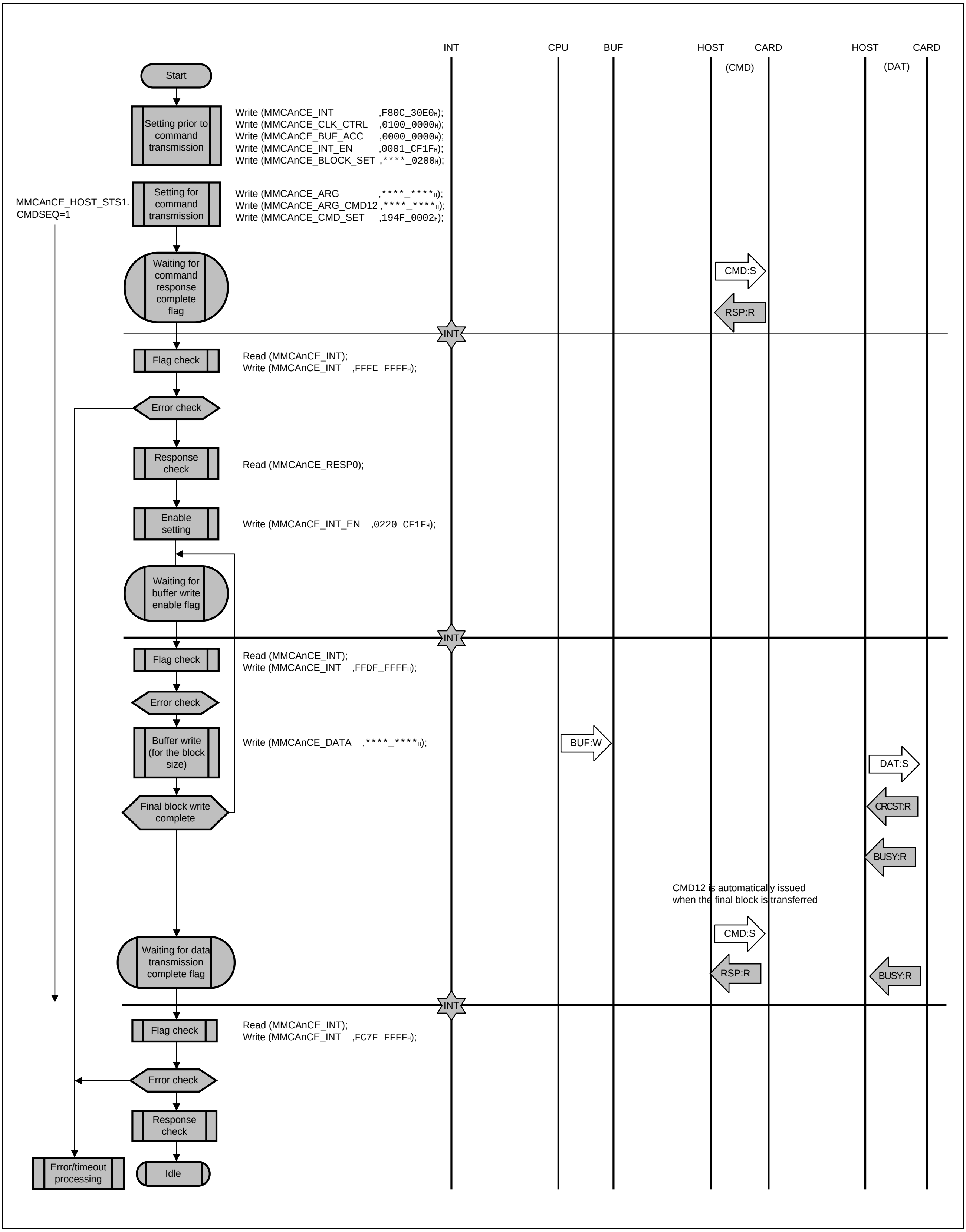
18.4.4.9 Multi-Block Write



```
Write (MMCAnCE_INT        ,F80C_30E0H);
Write (MMCAnCE_CLK_CTRL   ,0100_0000H);
Write (MMCAnCE_BUF_ACC    ,0000_0000H);
Write (MMCAnCE_INT_EN     ,0001_CF1FH);
Write (MMCAnCE_BLOCK_SET  ,****_0200H);

Write (MMCAnCE_ARG        ,****_****H);
Write (MMCAnCE_CMD_SET    ,194E_0002H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT   ,FFFE_FFFFH);

Read (MMCAnCE_RESP0);

Write (MMCAnCE_INT_EN   ,00A0_CF1FH);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT   ,FFDF_FFFFH);

Write (MMCAnCE_DATA   ,****_****H);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT   ,FF7F_FFFFH);
```

Figure 18.20 Multi-Block Write (CMD25 Pre-Defined)

18.4.4.10 Multi-Block Write (with Automatic CMD12 Issuance)



Write (MMCAnCE_INT ,F80C_30E0$_H$);
Write (MMCAnCE_CLK_CTRL ,0100_0000$_H$);
Write (MMCAnCE_BUF_ACC ,0000_0000$_H$);
Write (MMCAnCE_INT_EN ,0001_CF1F$_H$);
Write (MMCAnCE_BLOCK_SET ,****_0200$_H$);

Write (MMCAnCE_ARG ,****_****$_H$);
Write (MMCAnCE_ARG_CMD12 ,****_****$_H$);
Write (MMCAnCE_CMD_SET ,194F_0002$_H$);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT ,FFFE_FFFF$_H$);

Read (MMCAnCE_RESP0);

Write (MMCAnCE_INT_EN ,0220_CF1F$_H$);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT ,FFDF_FFFF$_H$);

Write (MMCAnCE_DATA ,****_****$_H$);

Read (MMCAnCE_INT);
Write (MMCAnCE_INT ,FC7F_FFFF$_H$);

Figure 18.21 Multi-Block Write (with Automatic CMD12 Issuance) (CMD25 Open-Ended)

18.4.4.11 Boot Operations

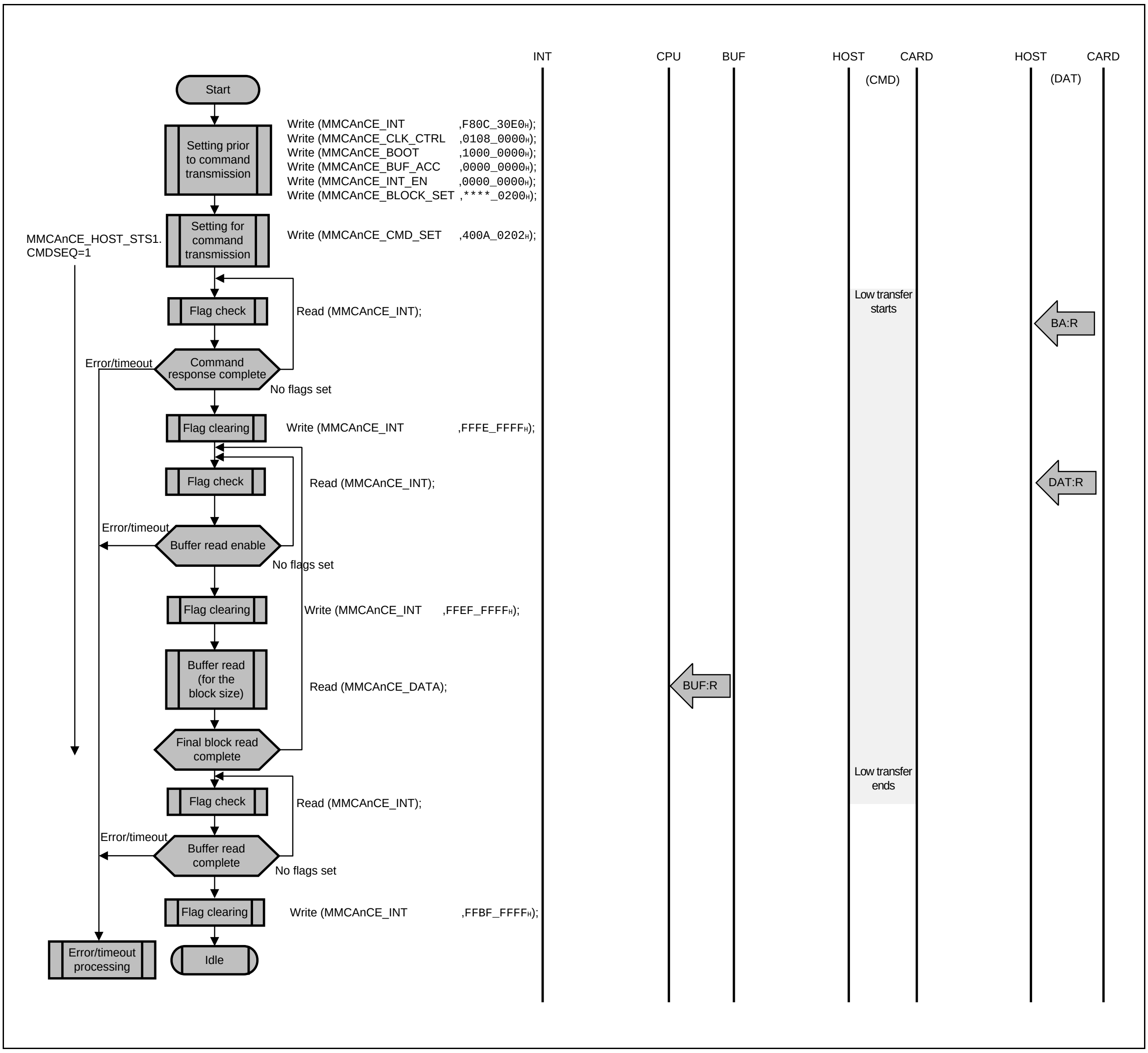


Figure 18.22 Boot Operations (with Boot Acknowledge)

18.4.4.12 Forcible Termination

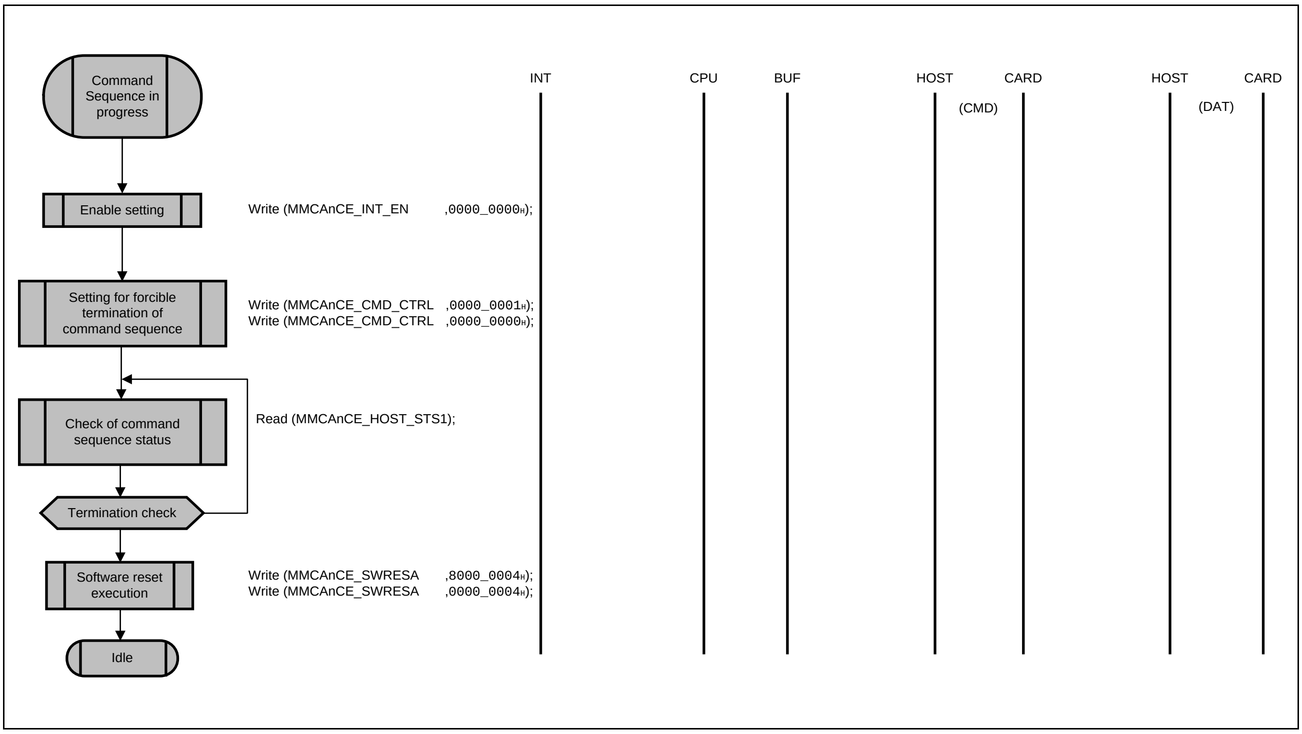


Figure 18.23 Forcible Termination

18.4.4.13 Setting Values of MMCAnCE_CMD_SET

The tables below lists the setting values required to issue commands.

Table 18.31 Setting Values of MMCAnCE_CMD_SET

Command	Response	MMCAnCE_CMD_SET																							Remarks
		Reserved	BOOT	CMD[5:0]	RTYP[1:0]	RBSY	Reserved	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	Reserved	CRC16C	BOOTACK	CRCSTE	TBIT	OPDM	Reserved	Reserved	SBIT	Reserved	DATW[1:0]	
CMD0	—	0	0	000000	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD1	R3	0	0	000001	01	0	0	0	0	0	0	01	01	0	0	0	0	0	0	0	0	0	0	00	
CMD2	R2	0	0	000010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD3	R1	0	0	000011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD4	—	0	0	000100	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD5	R1	0	0	000101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD6	R1	0	0	000110	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Background operation
	R1b	0	0	000110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD7	R1	0	0	000111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	000111	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD8	R1	0	0	001000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD9	R2	0	0	001001	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD10	R2	0	0	001010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD12	R1	0	0	001100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD13	R1	0	0	001101	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD14	R1	0	0	001110	01	0	0	1	0	0	0	00	00	0	1	0	0	0	0	0	0	1	0	**	
CMD15	—	0	0	001111	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD16	R1	0	0	010000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD17	R1	0	0	010001	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD18	R1	0	0	010010	01	0	0	1	0	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	010010	01	0	0	1	0	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD19	R1	0	0	010011	01	0	0	1	1	0	0	00	00	0	0	0	1	0	0	0	0	0	0	**	
CMD23	R1	0	0	010111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD24	R1	0	0	011000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD25	R1	0	0	011001	01	0	0	1	1	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	011001	01	0	0	1	1	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD26	R1	0	0	011010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD27	R1	0	0	011011	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD28	R1b	0	0	011100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD29	R1b	0	0	011101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD30	R1	0	0	011110	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	

Table 18.31 Setting Values of MMCAnCE_CMD_SET

Command	Response	MMCAnCE_CMD_SET																							Remarks
		Reserved	BOOT	CMD[5:0]	RTYP[1:0]	RBSY	Reserved	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	Reserved	CRC16C	BOOTACK	CRCSTE	TBIT	OPDM	Reserved	Reserved	SBIT	Reserved	DATW[1:0]	
CMD31	R1	0	0	011111	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD35	R1	0	0	100011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD36	R1	0	0	100100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD38	R1b	0	0	100110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD39	R4	0	0	100111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD40	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Send CMD
	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	1	1	0	0	0	0	00	Send RSP
CMD42	R1	0	0	101010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD55	R1	0	0	110111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD56	R1	0	0	111000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Read
	R1	0	0	111000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Write
Boot Operation		0	1	000000	00	0	0	1	0	1	0	00	00	0	0	*	0	0	0	0	0	0	0	**	

18.5 Detection and Correction of Errors in MMCA RAM

For details of this ECC function, see **Section 40A.2.6, ECC for Peripheral RAM**.

Section 19 Clocked Serial Interface G (CSIG)

This section contains a generic description of the Clocked Serial Interface G (CSIG).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of CSIG.

19.1 Features of RH850/F1KH, RH850/F1KM CSIG

19.1.1 Number of Units

This microcontroller has the following number of CSIG units.

Each CSIG unit has single channel interface.

Table 19.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	5	5	5
Name	CSIGn (n = 0 to 4)	CSIGn (n = 0 to 4)	CSIGn (n = 0 to 4)

Table 19.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	2	4	4	4
Name	CSIGn (n = 0)	CSIGn (n = 0, 1)	CSIGn (n = 0 to 3)	CSIGn (n = 0 to 3)	CSIGn (n = 0 to 3)

Table 19.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)

Table 19.4 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual CSIG units are identified by the index “n”: for example, CSIGnCTL0 (n = 0 to 4) is the CSIGn control register 0.

Table 19.5 Index (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual CSIG units are identified by the index “n”: for example, CSIGnCTL0 (n = 0 to 3) is the CSIGn control register 0.

Table 19.6 Index (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual CSIG units are identified by the index “n”: for example, CSIGnCTL0 (n = 0) is the CSIGn control register 0.

19.1.2 Register Base Addresses

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses.

Table 19.7 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<CSIG0_base>	FFD8 8000$_H$
<CSIG1_base>	FFD8 A000$_H$
<CSIG2_base>	FFD8 C000$_H$
<CSIG3_base>	FFD8 E000$_H$
<CSIG4_base>	FFD9 2000$_H$

Table 19.8 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<CSIG0_base>	FFD8 8000$_H$
<CSIG1_base>	FFD8 A000$_H$
<CSIG2_base>	FFD8 C000$_H$
<CSIG3_base>	FFD8 E000$_H$

Table 19.9 Register Base Address (RH850/F1KM-S1)

Base Address Name	Base Address
<CSIG0_base>	FFD8 8000$_H$

19.1.3 Clock Supply

The CSIG clock supply is shown in the following table.

Table 19.10 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
CSIGn	PCLK	CKSCLK_ICSI	Communication clock
	Register access clock	CPUCLK_L, CKSCLK_ICSI	Bus clock

19.1.4 Interrupt Requests

CSIG interrupt requests are listed in the following table.

Table 19.11 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CSIG0			
INTCSIGTIC	Communication status interrupt	27,118	8
INTCSIGTIR	Receive status interrupt	28,119	9
INTCSIGTIRE	Communication error interrupt	57	—
CSIG1			
INTCSIGTIC	Communication status interrupt	223	66
INTCSIGTIR	Receive status interrupt	224	67
INTCSIGTIRE	Communication error interrupt	225	—
CSIG2			
INTCSIGTIC	Communication status interrupt	326	78
INTCSIGTIR	Receive status interrupt	327	79
INTCSIGTIRE	Communication error interrupt	328	—
CSIG3			
INTCSIGTIC	Communication status interrupt	329	125
INTCSIGTIR	Receive status interrupt	330	126
INTCSIGTIRE	Communication error interrupt	331	—
CSIG4			
INTCSIGTIC	Communication status interrupt	98	39
INTCSIGTIR	Receive status interrupt	99	40
INTCSIGTIRE	Communication error interrupt	100	—

Table 19.12 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CSIG0			
INTCSIGTIC	Communication status interrupt	27,118	8
INTCSIGTIR	Receive status interrupt	28,119	9
INTCSIGTIRE	Communication error interrupt	57	—
CSIG1			
INTCSIGTIC	Communication status interrupt	223	66
INTCSIGTIR	Receive status interrupt	224	67
INTCSIGTIRE	Communication error interrupt	225	—
CSIG2			
INTCSIGTIC	Communication status interrupt	326	78
INTCSIGTIR	Receive status interrupt	327	79
INTCSIGTIRE	Communication error interrupt	328	—
CSIG3			
INTCSIGTIC	Communication status interrupt	329	125
INTCSIGTIR	Receive status interrupt	330	126
INTCSIGTIRE	Communication error interrupt	331	—

Table 19.13 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CSIG0			
INTCSIGTIC	Communication status interrupt	27,118	8
INTCSIGTIR	Receive status interrupt	28,119	9
INTCSIGTIRE	Communication error interrupt	57	—

19.1.5 Reset Sources

CSIG reset sources are listed in the following table. CSIG is initialized by these reset sources.

Table 19.14 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
CSIGn	All reset sources (ISORES)

19.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

Table 19.15 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC[*1]
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO[*1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG0SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG0RYI
CSIGTRYO	Ready / busy output signal	CSIG0RYO
CSIG1		
CSIGTSCK	Serial clock signal	CSIG1SC[*1]
CSIGTSI	Serial data input signal	CSIG1SI
CSIGTSO	Serial data output signal	CSIG1SO[*1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG1SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG1RYI
CSIGTRYO	Ready / busy output signal	CSIG1RYO
CSIG2		
CSIGTSCK	Serial clock signal	CSIG2SC[*1]
CSIGTSI	Serial data input signal	CSIG2SI
CSIGTSO	Serial data output signal	CSIG2SO[*1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG2SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG2RYI
CSIGTRYO	Ready / busy output signal	CSIG2RYO
CSIG3		
CSIGTSCK	Serial clock signal	CSIG3SC[*1]
CSIGTSI	Serial data input signal	CSIG3SI
CSIGTSO	Serial data output signal	CSIG3SO[*1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG3SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG3RYI
CSIGTRYO	Ready / busy output signal	CSIG3RYO
CSIG4		
CSIGTSCK	Serial clock signal	CSIG4SC[*1]
CSIGTSI	Serial data input signal	CSIG4SI
CSIGTSO	Serial data output signal	CSIG4SO[*1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG4SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG4RYI
CSIGTRYO	Ready / busy output signal	CSIG4RYO

Note 1. For the port pins that are used as CSIGnSO and CSIGnSC, set the output driver strength to high (PDSCn_m = 1).

Table 19.16 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC*[1]
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO*[1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG0SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG0RYI
CSIGTRYO	Ready / busy output signal	CSIG0RYO
CSIG1		
CSIGTSCK	Serial clock signal	CSIG1SC*[1]
CSIGTSI	Serial data input signal	CSIG1SI
CSIGTSO	Serial data output signal	CSIG1SO*[1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG1SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG1RYI
CSIGTRYO	Ready / busy output signal	CSIG1RYO
CSIG2		
CSIGTSCK	Serial clock signal	CSIG2SC*[1]
CSIGTSI	Serial data input signal	CSIG2SI
CSIGTSO	Serial data output signal	CSIG2SO*[1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG2SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG2RYI
CSIGTRYO	Ready / busy output signal	CSIG2RYO
CSIG3		
CSIGTSCK	Serial clock signal	CSIG3SC*[1]
CSIGTSI	Serial data input signal	CSIG3SI
CSIGTSO	Serial data output signal	CSIG3SO*[1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG3SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG3RYI
CSIGTRYO	Ready / busy output signal	CSIG3RYO

Note 1. For the port pins that are used as CSIGnSO and CSIGnSC, set the output driver strength to high (PDSCn_m = 1).

Table 19.17 External Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC*[1]
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO*[1]
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG0SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG0RYI
CSIGTRYO	Ready / busy output signal	CSIG0RYO

Note 1. For the port pins that are used as CSIGnSO and CSIGnSC, set the output driver strength to high (PDSCn_m = 1).

19.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSIGnSO (CSIGTSO) output are shown in the following table. See **Section 19.5.10, Error Detection** for details on data consistency checking.

Table 19.18 Port Pins for Data Consistency Checking (RH850/F1KH-D8)

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2
CSIG1		
CSIGTSO	P11_9	ALT_OUT1
CSIG2		
CSIGTSO	P12_5	ALT_OUT3
CSIG3		
CSIGTSO	P20_1	ALT_OUT4
CSIG4		
CSIGTSO	P1_3	ALT_OUT3
	P23_6	ALT_OUT1

Table 19.19 Port Pins for Data Consistency Checking (RH850/F1KM-S4)

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2
CSIG1		
CSIGTSO	P11_9	ALT_OUT1
CSIG2		
CSIGTSO	P12_5	ALT_OUT3
CSIG3		
CSIGTSO	P20_1	ALT_OUT4

Table 19.20 Port Pins for Data Consistency Checking (RH850/F1KM-S1)

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2

19.2 Overview

19.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable.
- Slave select input signal ($\overline{\text{CSIGTSSI}}$) is available.
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
- Clock phases and data phases are selectable.
- Data transfer with MSB first or LSB first is selectable.
- Transfer data length is selectable from 7 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data.
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (INTCSIGTIC, INTCSIGTIR, INTCSIGTIRE)
- Built-in LBM (Loop Back Mode) function for self-test

19.2.2 Functional Overview Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTSCK (output in master mode, input in slave mode)
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

The CSIGnCTL2 register is used to select whether the CSIG should be operated in master mode or slave mode.

Additional signals can be used for external control and monitoring:

- $\overline{\text{CSIGTSSI}}$: Slave select input signal
- CSIGTRYO: Ready/busy output signal (handshake signal)
- CSIGTRYI: Ready/busy input signal (handshake signal)

Data transmission is bit-wise and serial and synchronous to the transmission clock.

The following table shows the most important registers for setting up CSIG.

Table 19.21 Main Registers of CSIG

Register	Function
CSIGnCTL0	Provides and stops operating clock and enables/disables data transmission and data reception.
CSIGnCTL1	Controls options such as interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIGnCFG0	Configures the communication protocol.

19.2.3 Block Diagram

The following block diagram shows the main components of the CSIG.

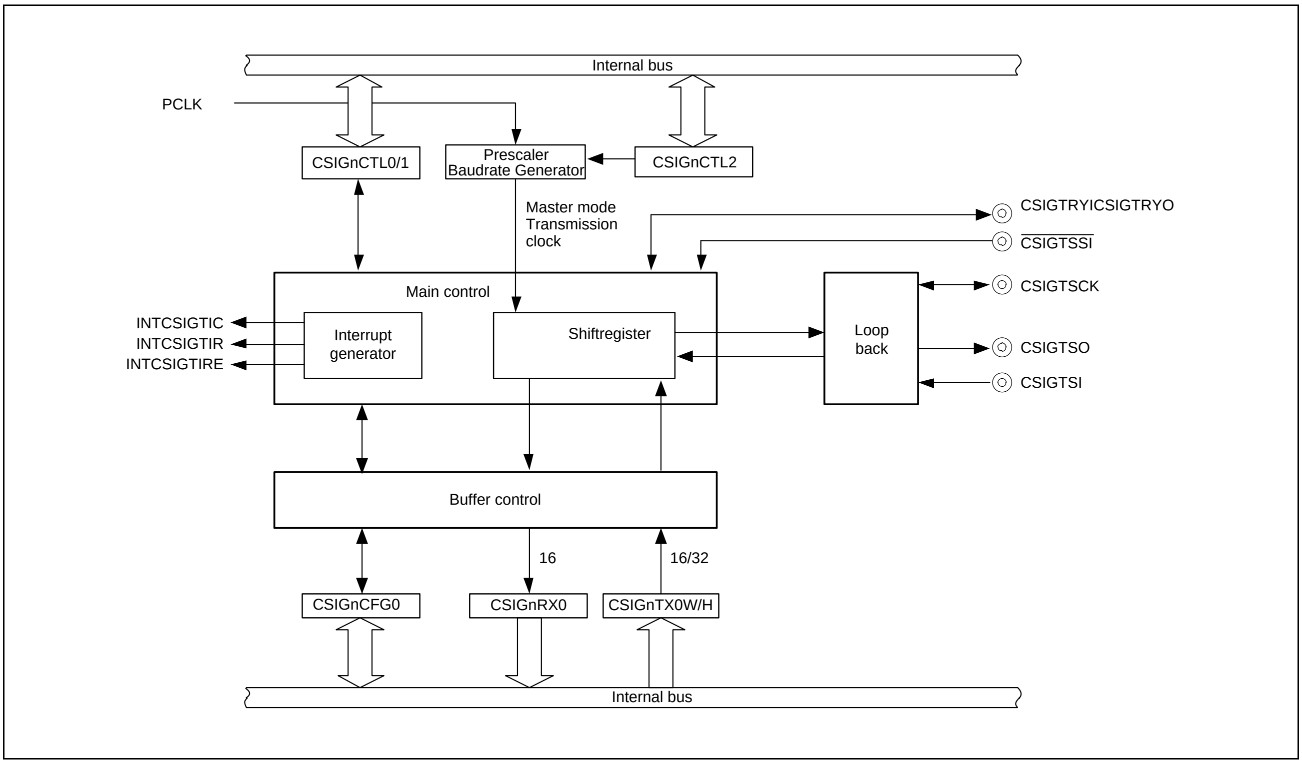


Figure 19.1 CSIG Block Diagram

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

19.3 Registers

19.3.1 List of Registers

CSIG registers are listed in the following table.

For details on <CSIGn_base>, see **Section 19.1.2, Register Base Addresses**.

Table 19.22 List of Registers

Module Name	Register Name	Symbol	Address
CSIGn	CSIGn control register 0	CSIGnCTL0	<CSIGn_base> + 0000_H
	CSIGn control register 1	CSIGnCTL1	<CSIGn_base> + 0010_H
	CSIGn control register 2	CSIGnCTL2	<CSIGn_base> + 0014_H
	CSIGn status register 0	CSIGnSTR0	<CSIGn_base> + 0004_H
	CSIGn status clear register 0	CSIGnSTCR0	<CSIGn_base> + 0008_H
	CSIGn Rx-only mode control register 0	CSIGnBCTL0	<CSIGn_base> + 1000_H
	CSIGn configuration register 0	CSIGnCFG0	<CSIGn_base> + 1010_H
	CSIGn transmission register 0 for word access	CSIGnTX0W	<CSIGn_base> + 1004_H
	CSIGn transmission register 0 for half word access	CSIGnTX0H	<CSIGn_base> + 1008_H
	CSIGn reception register 0	CSIGnRX0	<CSIGn_base> + $100C_H$
	CSIGn emulation register	CSIGnEMU	<CSIGn_base> + 0018_H

19.3.2 CSIGnCTL0 — CSIGn Control Register 0

This register controls the operation clock, and enables or disables transmission/reception.

Access: This register can be read or written in 8-bit units.

Address: <CSIGn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIGnPWR	CSIGnTXE	CSIGnRXE	—	—	—	—	CSIGnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 19.23 CSIGnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGnPWR	Controls operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. Clock supply to internal circuits stops. If CSIGnPWR is cleared during communication, ongoing communication is aborted. In this case, communication setting must be started from the beginning.
6	CSIGnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception. 0: Reception disabled 1: Reception enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGnMBS	When setting CSIGnCTL0.CSIGnPWR = 1, be sure to set to this bit to 1. (The value after reset is “0”.)

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.3 CSIGnCTL1 — CSIGn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

Access: This register can be read or written in 32-bit units.

Address: <CSIGn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn CKR	CSIGn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn EDLE	—	CSIGn DCS	—	CSIGn LBM	CSIGn SIT	CSIGn HSE	CSIGn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 19.24 CSIGnCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	CSIGnCKR	CSIGTSCK clock inversion function 0: Default level of CSIGTSCK is high. 1: Default level of CSIGTSCK is low. The CSIGnCKR bit is used in combination with the CSIGnCFG0.CSIGnDAP bit. For details, see **Section 19.3.8, CSIGnCFG0 — CSIGn Configuration Register 0**.
16	CSIGnSLIT	Selects the timing of interrupt INTCSIGTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGnTX0W/H is empty and available for storing the next data. For details, see **Section 19.4.2, INTCSIGTIC (Communication Status Interrupt)**.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	CSIGnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, see **Section 19.5.5.2, Data Length Selection with Extended Data Length**.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSIGnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, see **Section 19.5.10.1, Data Consistency Check**.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated Loop-back mode can be set only in master mode. Set this bit to 0 in slave mode. For details, see **Section 19.5.9, Loop-Back Mode**.

Table 19.24 CSIGnCTL1 Register Contents

Bit Position	Bit Name	Function
2	CSIGnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, see **Section 19.4.1, Interrupt Delay**.
1	CSIGnHSE	Enables/disables handshake function. 0: Handshake function disabled 1: Handshake function enabled For details, see **Section 19.5.8, Handshake Function**.
0	CSIGnSSE	Enables/disables slave select function. 0: Input signal $\overline{\text{CSIGTSSI}}$ disabled. 1: Input signal $\overline{\text{CSIGTSSI}}$ enabled. If the slave select function is not used, this bit must be set to 0 (see also **Section 19.5.2, Master/Slave Connections**).

Details about CSIGnCTL1.CSIGnSSE:

Table 19.25 Operation of the Slave Select Function during Reception

CSIGnCTL0.CSIGnRXE	CSIGnCTL1.CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Receive Operation
0	—	—	Reception disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 19.26 Operation of the Slave Select Function during Transmission

CSIGnCTL0.CSIGnTXE	CSIGnCTL1.CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Transmit Operation
0	—	—	Transmission disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.4 CSIGnCTL2 — CSIGn Control Register 2

This register selects the communication clock.

Access: This register can be read or written in 16-bit units.

Address: <CSIGn_base> + 0014_H

Value after reset: $E000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnPRS[2:0]			—	CSIGnBRS[11:0]											
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.27 CSIGnCTL2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 13</td><td>CSIGnPRS[2:0]</td><td>Selects the value of the prescaler.
<table>
<tr><th>CSIGn PRS2</th><th>CSIGn PRS1</th><th>CSIGn PRS0</th><th>Prescaler Output (PRSOUT)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>PCLK (master mode)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>PCLK / 2 (master mode)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>PCLK / 4 (master mode)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>PCLK / 8 (master mode)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>PCLK / 16 (master mode)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>PCLK / 32 (master mode)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>PCLK / 64 (master mode)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock via CSIGTSCK (slave mode)</td></tr>
</table></td></tr>
<tr><td>12</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>11 to 0</td><td>CSIGnBRS[11:0]</td><td>Selects the transfer clock frequency.
Settings of the CSIGnBRS[11:0] bits are valid only in master mode. They are ignored in slave mode.
<table>
<tr><th>CSIGnBRS[11:0]</th><th>Transfer Clock Frequency of CSIGTSCK</th></tr>
<tr><td>0</td><td>BRG is stopped</td></tr>
<tr><td>1</td><td>PCLK / (2^{α} × 1 × 2)</td></tr>
<tr><td>2</td><td>PCLK / (2^{α} × 2 × 2)</td></tr>
<tr><td>3</td><td>PCLK / (2^{α} × 3 × 2)</td></tr>
<tr><td>4</td><td>PCLK / (2^{α} × 4 × 2)</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>4095</td><td>PCLK / (2^{α} × 4095 × 2)</td></tr>
</table>
Note: α = 0 to 6 (value set by CSIGnPRS[2:0])</td></tr>
</table>

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.5 CSIGnSTR0 — CSIGn Status Register 0

This register indicates the status of the CSIG.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIGn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn TSF	—	—	—	CSIGn DCE	—	CSIGn PE	CSIGn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.28 CSIGnSTR0 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	CSIGnTSF	Transfer Status Flag 0: Idle state 1: Communication is in progress or being prepared The timing to set or clear this bit is as follows:

Master Mode	Timing to Set	Timing to Clear
Tx-only mode	Writing to transmission register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	Reading from reception register	

Slave Mode	Timing to Set	Timing to Clear
Tx-only mode	Writing to transmission register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	CSIGTSCK input	

Bit Position	Bit Name	Function
6 to 4	Reserved	When read, the value after reset is returned.
3	CSIGnDCE	Data Consistency Check Error Flag 0: No data consistency check error detected 1: Data consistency check error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnDCEC. When setting to 1 due to data consistency check error detection and clearing to 0 by CSIGnSTCR0.CSIGnDCEC occur simultaneously, setting to 1 due to data consistency check error detection takes precedence. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.
2	Reserved	When read, the value after reset is returned.

Table 19.28 CSIGnSTR0 Register Contents

Bit Position	Bit Name	Function
1	CSIGnPE	Parity Error Flag 0: No parity error detected 1: Parity error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnPEC. When setting to 1 due to parity error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnPEC occur simultaneously, setting to 1 due to parity error detection takes precedence. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.
0	CSIGnOVE	Overrun Error Flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence. This bit is initialized by the value of CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.6 CSIGnSTCR0 — CSIGn Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access:	This register can be read or written in 16-bit units. When read, the value 0000$_H$ is always returned.
Address:	<CSIGn_base> + 0008$_H$
Value after reset:	0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn DCEC	—	CSIGn PEC	CSIGn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 19.29 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGnDCEC	Controls the data consistency check error flag clear command. 0: No operation. Read value is always 0. 1: Clears the data consistency check error flag (CSIGnSTR0.CSIGnDCE).
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIGnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clears the parity error flag (CSIGnSTR0.CSIGnPE).
0	CSIGnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clears the overrun error flag (CSIGnSTR0.CSIGnOVE).

19.3.7 CSIGnBCTL0 — CSIGn Rx-Only Mode Control Register 0

This register enables/disables the data transfer in Rx-only mode.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <CSIGn_base> + 1000_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CSIGnSCE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Table 19.30 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGnSCE	Disables/enables the start of the next data reception by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, see **Section 19.5.4.2, Receive-Only Mode**.

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.8 CSIGnCFG0 — CSIGn Configuration Register 0

This register configures the communication protocols such as data length, parity, transfer direction, clock phase, and data phase.

Access: This register can be read or written in 32-bit units.

Address: <CSIGn_base> + 1010$_H$

Value after reset: 0000 0000$_H$

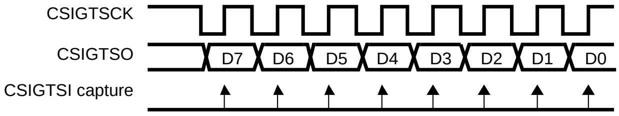
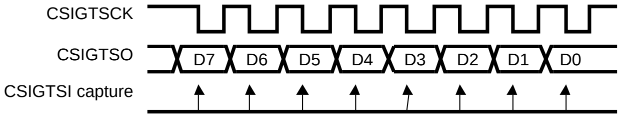
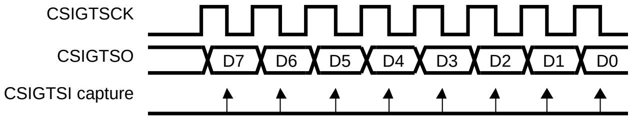
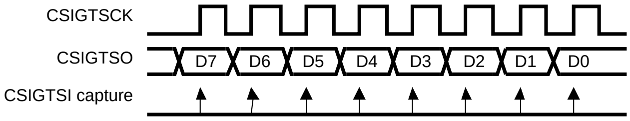
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGnPS[1:0]		CSIGnDLS[3:0]				—	—	—	—	—	CSIGnDIR	—	CSIGnDAP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.31 CSIGnCFG0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31, 30</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>29, 28</td><td>CSIGnPS[1:0]</td><td>Specifies parity.
<table>
<tr><th>CSIGnPS1</th><th>CSIGnPS0</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Does not transmit any parity bit.</td><td>Does not wait for reception of the parity bit.</td></tr>
<tr><td>0</td><td>1</td><td>Adds a parity bit fixed to 0.</td><td>Waits for reception of the parity bit but does not evaluate it.</td></tr>
<tr><td>1</td><td>0</td><td>Adds the odd parity bit.</td><td>Waits for the odd parity bit.</td></tr>
<tr><td>1</td><td>1</td><td>Adds the even parity bit.</td><td>Waits for the even parity bit.</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIGnDLS[3:0]</td><td>Specifies data length.
0: Data length is 16 bits
1: Data length is 1 bit
2: Data length is 2 bits
...
15: Data length is 15 bits
CAUTION: Do not set bits CSIGnCFG0.CSIGnDLS[3:0] to values 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. Transmitting two consecutive data with a data length of less than 7 bits is prohibited.</td></tr>
<tr><td>23 to 19</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>18</td><td>CSIGnDIR</td><td>Selects the serial data direction.
0: Data is transmitted/received with MSB first
1: Data is transmitted/received with LSB first</td></tr>
<tr><td>17</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 19.31 CSIGnCFG0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>16</td><td>CSIGnDAP</td><td>Data Phase Selection
Used in conjunction with CSIGnCTL1.CSIGnCKR bit to select the data phase. Refer to the following table for the selectable clock phases and data phases.
<table>
<tr><th>CSIGnCTL1.
CSIGnCKR</th><th>CSIGn
DAP</th><th>Clock and Data Phase Selection</th></tr>
<tr><td>0</td><td>0</td><td>
</td></tr>
<tr><td>0</td><td>1</td><td>
</td></tr>
<tr><td>1</td><td>0</td><td>
</td></tr>
<tr><td>1</td><td>1</td><td>
</td></tr>
</table></td></tr>
<tr><td>15 to 0</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.9 CSIGnTX0W — CSIGn Transmission Register 0 for Word Access

This register stores the transmission data and specifies the extended data length.

Access:	This register can be read or written in 32-bit units.
Address:	<CSIGn_base> + 1004_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGn EDL	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.32 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	CSIGnEDL	Specifies the extended data length. 0: Normal operation 1: Extended data length enabled The associated data is transmitted as 16-bit data. This bit can only be set if CSIGnCTL1.CSIGnEDLE = 1.
28 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.10 CSIGnTX0H — CSIGn Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of CSIGnTX0W register.

The settings specified by the upper 16 bits of CSIGnTX0W are applied to the transmission.

Access: This register can be read or written in 16-bit units.

Address: <CSIGn_base> + 1008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.33 CSIGnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.11 CSIGnRX0 — CSIGn Reception Register 0

This register stores the received data.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <CSIGn_base> + $100C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnRX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.34 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX[15:0]	Received Data These bits are initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When reading, the values of these bits must be read at least 1 clock before the generation of CSIGTIR interrupt.

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.12 CSIGnEMU — CSIGn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 8-bit or 1-bit units.
Write to this register when EPC.SVSTOP = 0.

Address: <CSIGn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIGnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 19.35 CSIGnEMU Register Contents

Bit Position	Bit Name	Function
7	CSIGnSVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. ● When the EPC.SVSTOP bit is set to 0: Continues transmit/receive operation regardless of the setting of this bit. ● When the EPC.SVSTOP bit is set to 1: 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When configuring this register, see **Table 19.36, List of Cautions when Configuring the Registers**.

19.3.13 List of Cautions

Table 19.36 List of Cautions when Configuring the Registers

Register Name	Bit Name	Cautions
CSIGnCTL0	CSIGnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIGnCTL0	CSIGnTXE CSIGnRXE	Do not modify any of these bits while CSIGnCTL0.CSIGnPWR = 0. (These bits can be modified simultaneously with the CSIGnCTL0.CSIGnPWR bit.) Do not modify these bits while CSIGnSTR0.CSIGnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIGnCTL0	CSIGnMBS	When setting CSIGnCTL0.CSIGnPWR = 1, be sure to set to this bit to 1. (The value after reset is “0”.) This bit must be modified simultaneously with CSIGnCTL0.CSIGnPWR bit.
CSIGnCTL1	CSIGnCKR	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnSLIT CSIGnEDLE CSIGnDCS CSIGnHSE	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnLBM	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting this bit to 1 is prohibited in slave mode.
CSIGnCTL1	CSIGnSSE	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIGnCTL1	CSIGnSIT	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIGnCTL2	CSIGnPRS[2:0] CSIGnBRS[11:0]	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of the maximum transfer clock frequency is as follows. ● Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4) ● Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
CSIGnSTR0	CSIGnTSF	Writing to this bit is prohibited, and only reading is permitted.
CSIGnSTR0	CSIGnDCE CSIGnPE CSIGnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0.
CSIGnBCTL0	CSIGnSCE	Write to this bit before CSIGnRX0 is read. Fix the CSIGnSCE bit to 0 when the transfer mode is transmit mode or transmit/receive mode.
CSIGnCFG0	CSIGnPS[1:0] CSIGnDLS[3:0] CSIGnDIR CSIGnDAP	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnTX0W	CSIGnEDL	This bit is valid only when CSIGnCTL1.CSIGnEDLE = 1.
CSIGnTX0W CSIGnTX0H		Write access to these bits are prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnRX0		These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0. Read access to this bit is prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnEMU	CSIGnSVSDIS	Modification of this bit is prohibited while SVSTOP = 1.

19.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIGTIC (communication status interrupt)
- INTCSIGTIR (reception status interrupt)
- INTCSIGTIRE (communication error interrupt)

19.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half cycle of the transmission clock CSIGTSCK. This function is not available in slave mode.

The delay is specified by setting CSIGnCTL1.CSIGnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIGnCTL1.CSIGnSIT = 1 (interrupt delay enabled), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

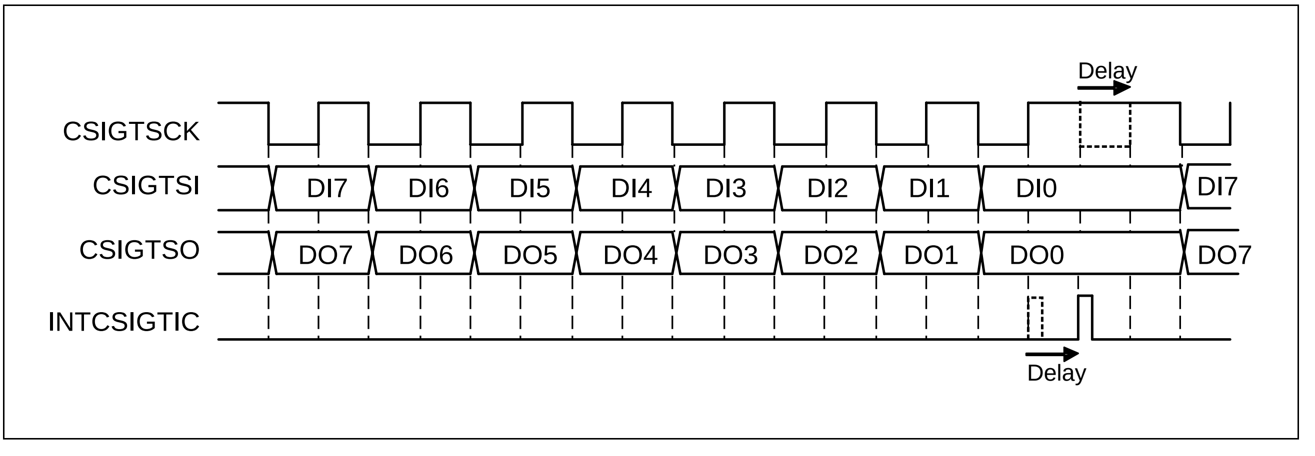


Figure 19.2 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

19.4.2 INTCSIGTIC (Communication Status Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits), and CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing).

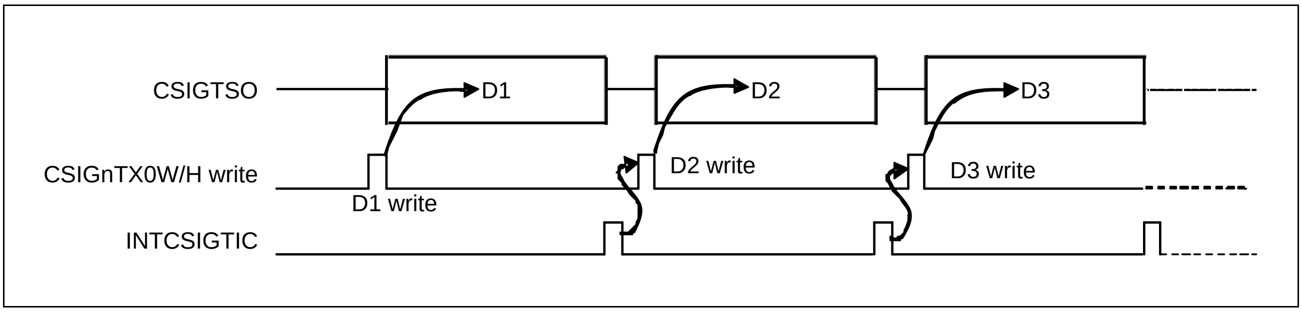


Figure 19.3 Generation of INTCSIGTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)

However, INTCSIGTIC can also be set up to occur when the CSIGnTX0W/H register is empty and available for receiving the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

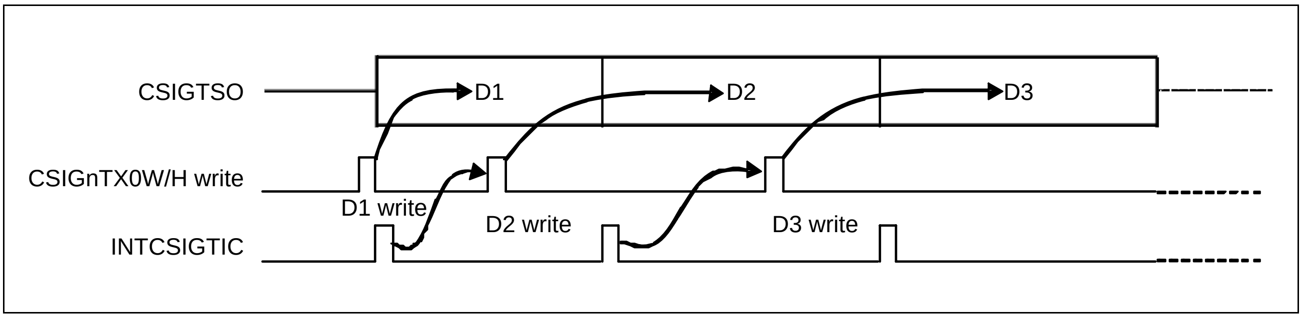


Figure 19.4 Generation of INTCSIGTIC at the Beginning of Communication

19.4.3 INTCSIGTIR (Reception Status Interrupt)

This interrupt is generated in receive-only and transmit/receive mode after data has been received and is available in the reception register. It can be used to trigger a DMA for reading the received data from CSIGnRX0 register.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

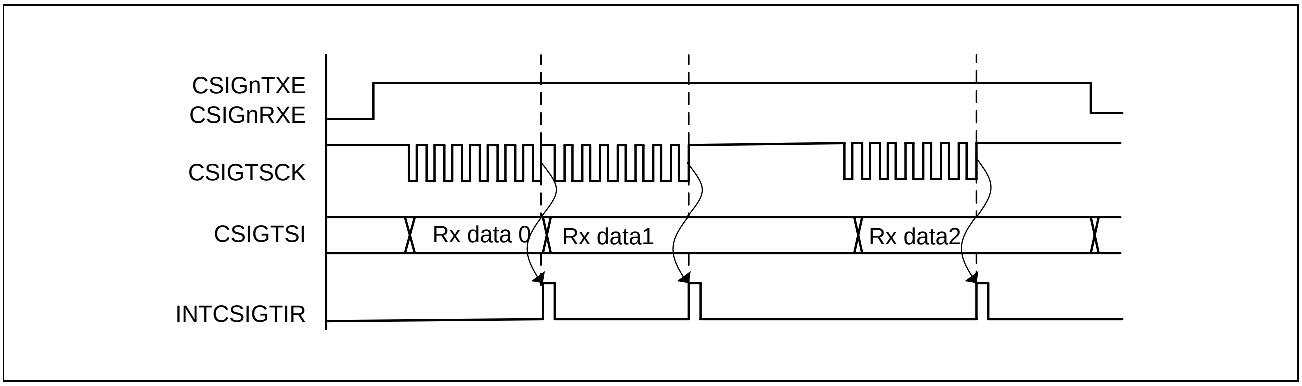


Figure 19.5 Generation of INTCSIGTIR

19.4.4 INTCSIGTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 19.37 Data Error Types

Error Type	Communication Status After Error Interrupt	Note
Parity error	Interrupt is generated and communication continues.	—
Data consistency check error	Interrupt is generated and communication continues.	—
Overrun error*1	When CSIGnCTL1.CSIGnHSE = 0 (handshake function disabled) in slave mode, interrupt is generated and communication continues.	When CSIGnCTL1.CSIGnHSE = 1 (handshake function enabled) in slave mode, communication stops due to the handshake. An interrupt is not generated and an overrun error does not occur.

Note 1. In master mode, overrun errors do not occur.
In slave mode, communication cannot be stopped.

The type of error that caused the generation of INTCSIGTIRE is indicated in register CSIGnSTR0.

For details about the various error types, see **Section 19.5.10, Error Detection**.

19.5 Operation

19.5.1 Master/Slave Mode

Whether CSIG operates in master mode or in slave mode depends on the setting of bits CSIGnCTL2.CSIGnPRS[2:0]. If master mode is selected, the source of the transmission clock must be selected as well.

19.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the built-in baud rate generator (BRG) and supplied to the slave via signal CSIGTSCK.

Master mode is enabled by setting bits CSIGnCTL2.CSIGnPRS[2:0] to values other than 111_B. In master mode, the frequency of BRG can be configured by setting bits CSIGnCTL2. CSIGnPRS[2:0] and bits CSIGnCTL2.CSIGnBRS[11:0].

The default level of CSIGTSCK depends on the CSIGTSCK clock inversion function bit; it is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

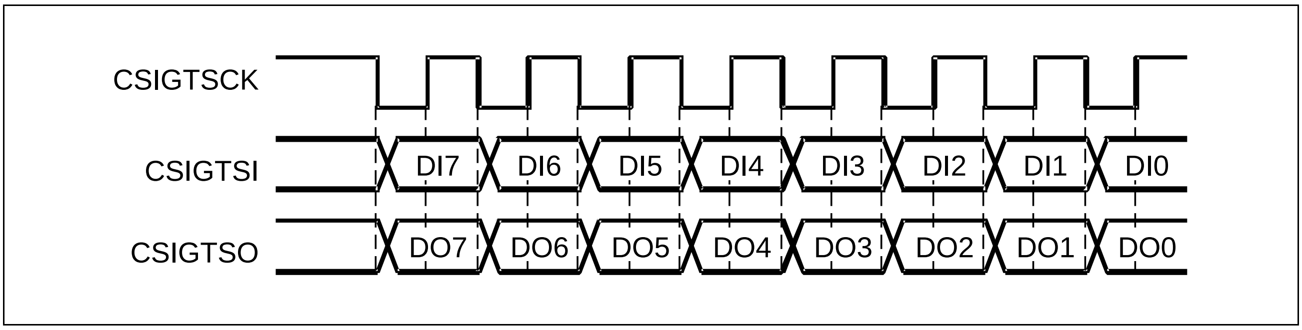


Figure 19.6 Transmission/Reception in Master Mode

19.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is supplied via the signal CSIGTSCK. Transmit/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting bits CSIGnCTL2.CSIGnBRS[11:0] to 000_H.

The example below shows the communication in slave mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

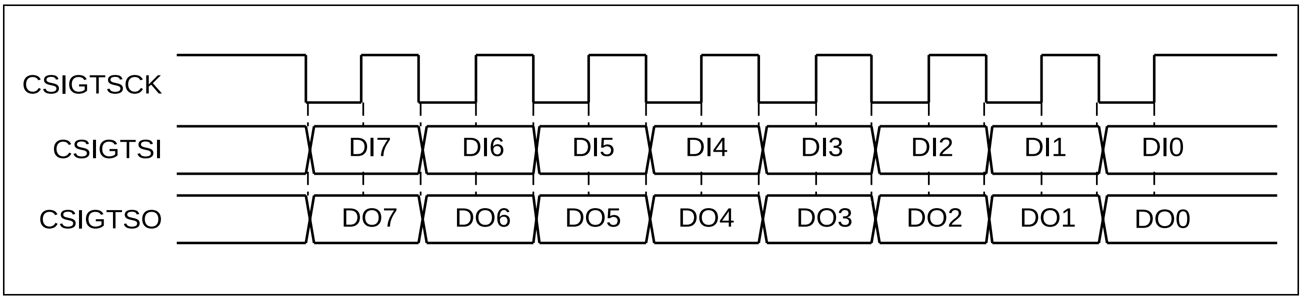


Figure 19.7 Transmission/Reception in Slave Mode

19.5.2 Master/Slave Connections

19.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

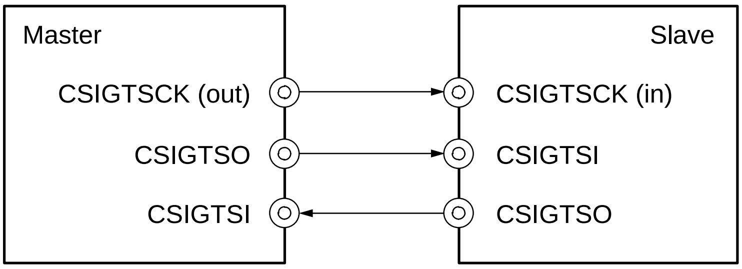


Figure 19.8 Direct Master/Slave Connection

19.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this case, the master must provide one slave select (SS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIGTSSI}}$ of the slave.

The $\overline{\text{CSIGTSSI}}$ signal can be enabled or disabled by the CSIGnCTL1.CSIGnSSE bit.

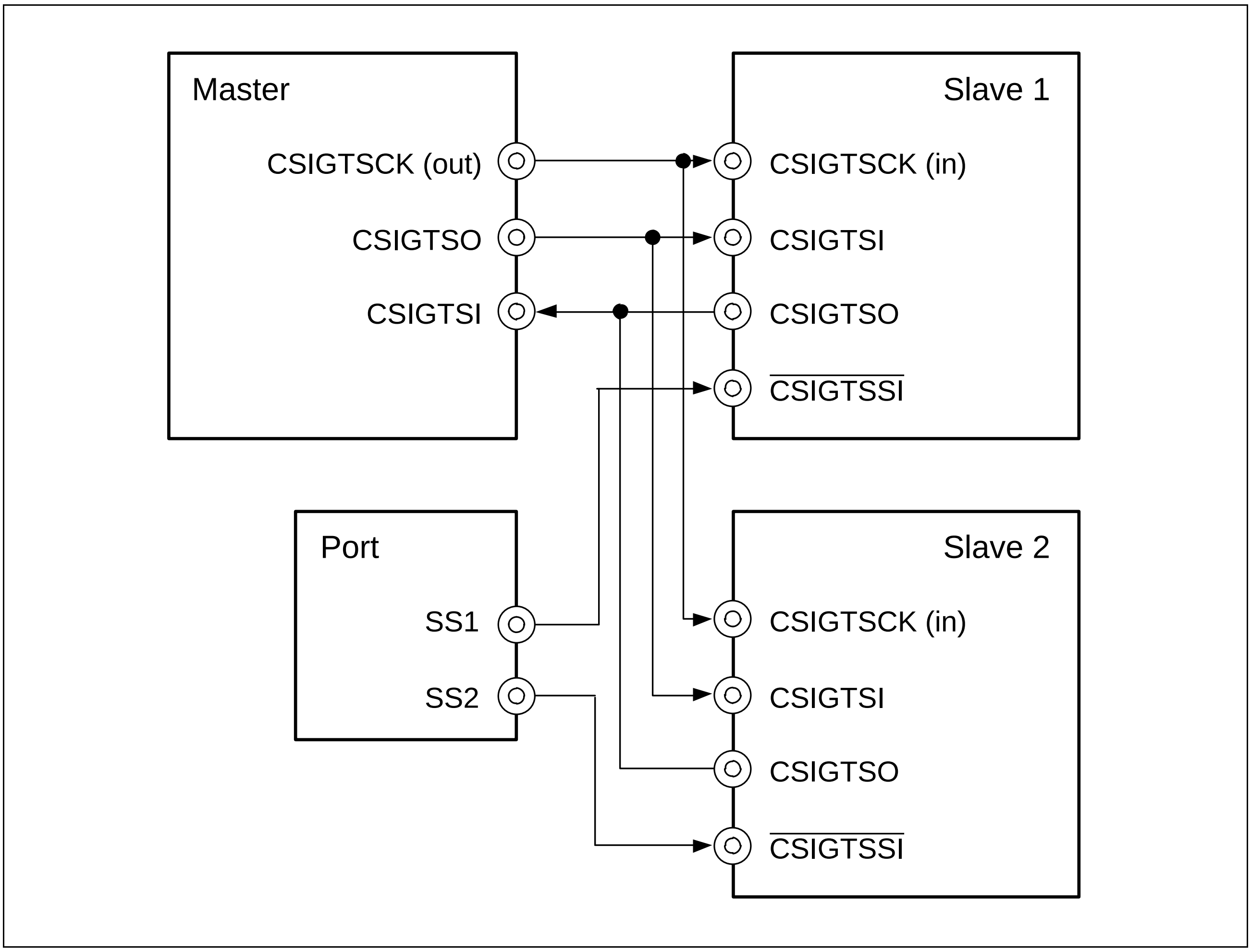


Figure 19.9 Master to Multiple Slaves Connection

A slave is selected (enabled) when its $\overline{\text{CSIGTSSI}}$ signal is low.

If it is not selected, the slave will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIGnCTL0.CSIGnTXE = 1), the CSIGTSO output buffer of the slave which is not selected is disabled and set to input mode in order to avoid interference with the outputs of other selected slaves.

19.5.3 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

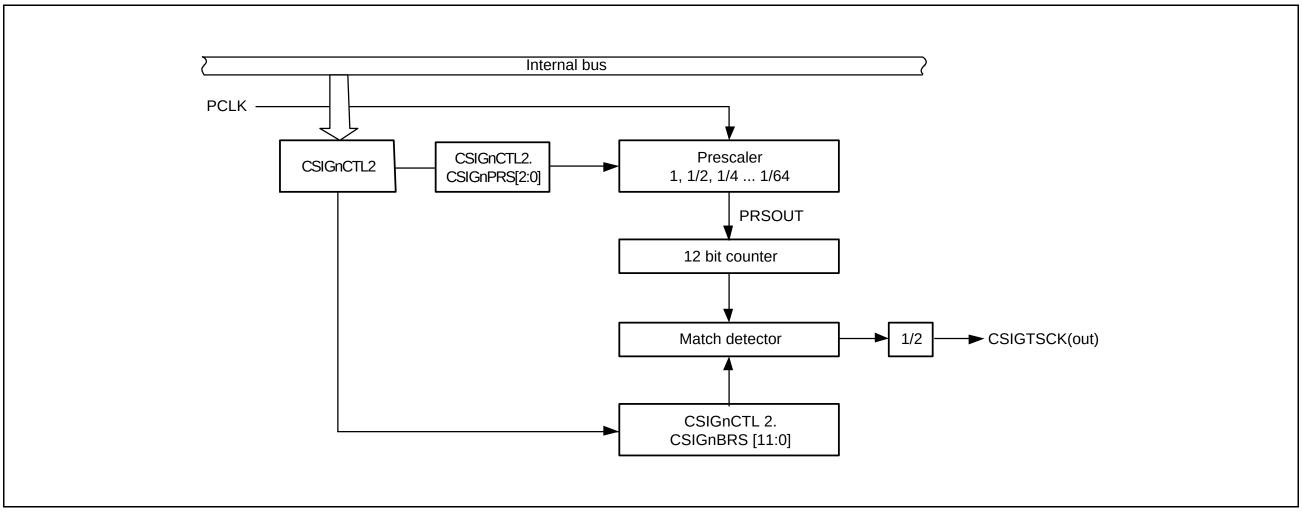


Figure 19.10 BRG Block Diagram

Setting CSIGnCTL2.CSIGnBRS[11:0] to 000_H disables the BRG.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIGTSCK) = PCLK / (division ratio of PCLK) = PCLK / ($2^{\alpha} \times k \times 2$) where:

α = CSIGnCTL2.CSIGnPRS[2:0] = 0 to 6

k = CSIGnCTL2.CSIGnBRS[11:0] = 1 to 4095

Transfer clock frequency upper and lower limits

When specifying the transfer clock frequency, please note the following:

- For the maximum transfer clock frequency of this product in master mode or slave mode, refer to the CSIG timing shown in the electrical characteristics. In addition, in either mode, specify a frequency within the defined range.
- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

19.5.4 Data Transfer Modes

19.5.4.1 Transmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 places CSIG in transmit-only mode.
Transmission starts when data to be transmitted is written in the CSIGnTX0W or CSIGnTX0H register.

CAUTION

When the mode transitions from one of the receive modes to transmit-only mode, the data in the CSIGnRX0 buffer becomes undefined after completion of the first transmission.
Therefore, the reception register CSIGnRX0 has to be read before switching to transmit-only mode.

19.5.4.2 Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGnRX0 register.
All subsequent receptions are triggered by reads from the CSIGnRX0 register, as long as CSIGnBCTL0.CSIGnSCE = 1.

Moreover, CSIGnBCTL0.CSIGnSCE has to be set to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Set CSIGnBCTL0.CSIGnSCE = 1.
2. Read CSIGnRX0 (dummy data).
3. Wait for the reception interrupt INTCSIGTIR.
4. Read CSIGnRX0 (received data).
   In case more data receptions follow at step 3, continue to read until all data is received.
   Before reading the last received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0.

In slave mode, reception starts when the communication clock CSIGTSCK is received from the master. In this case, it is not necessary to read data from the CSIGnRX0 register of the slave.

NOTE

In slave mode, any previously received data must be read from the reception register CSIGnRX0 in order to avoid any overwrite situation.

19.5.4.3 Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when the data to be transmitted is written to the CSIGnTX0W or CSIGnTX0H register.

19.5.5 Data Length Selection

19.5.5.1 Data Length Selection without Extended Length

Transmission data length is selectable from 7 to 16 bits using the CSIGnDLS[3:0] bits in the CSIGnCFG0 register. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

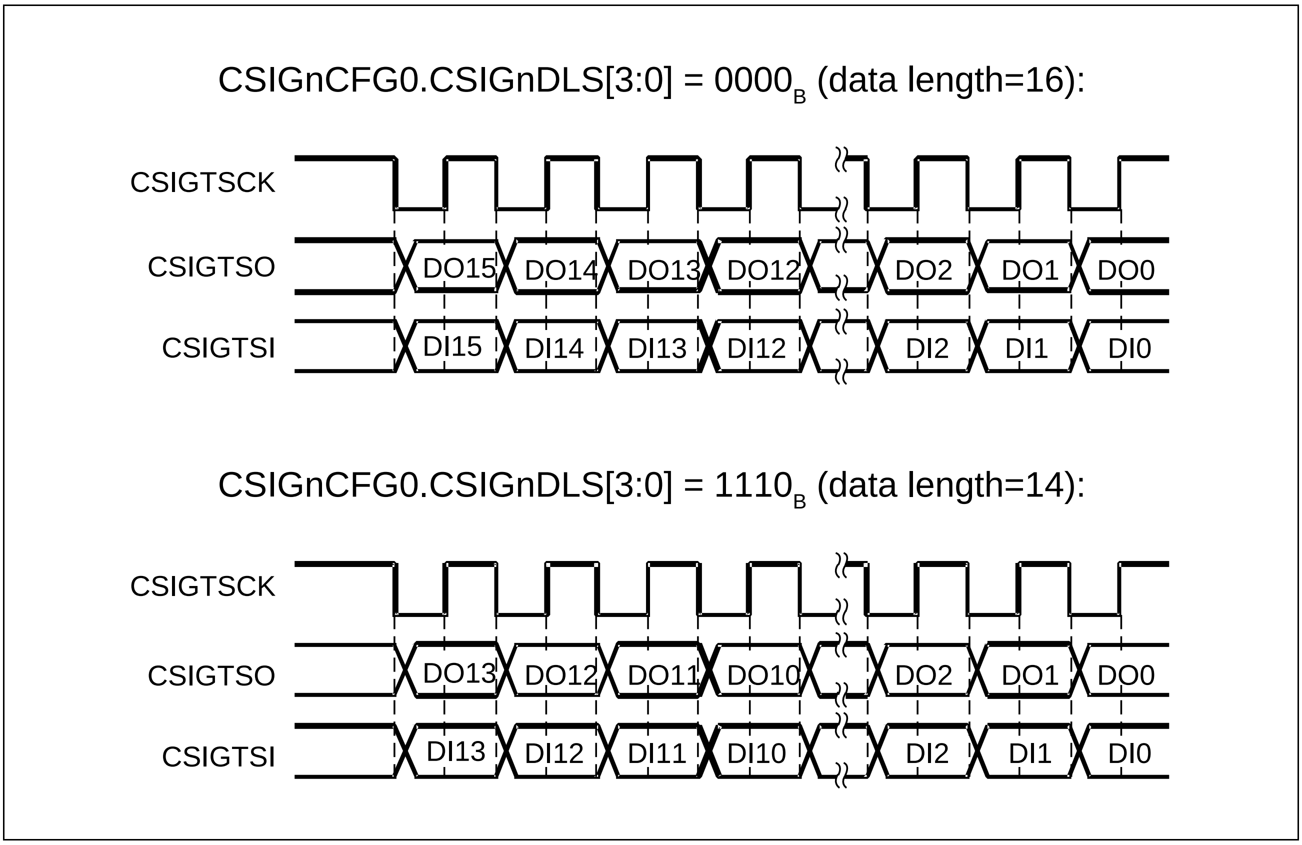


Figure 19.11 Data Length Selection Function

19.5.5.2 Data Length Selection with Extended Data Length

If the data to be transmitted/received exceeds 16 bits, the extended data length (EDL) function can be used.

The EDL function is enabled by setting the CSIGnCTL1.CSIGnEDLE bit to 1.

The following describes how the EDL function works and how to specify the EDL setting.

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIGnCFG0.CSIGnDLS[3:0] bits.
- Set the CSIGnTX0W.CSIGnEDL bit to 1 to transmit the 16-bit blocks. In this case, the data written to the CSIGnTX0W register is sent as 16-bit data regardless of the setting of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- The transfer completes after the data with the specified data length (the remainder when CSIGnTX0W.CSIGnEDL = 0) has been sent.

Example

Example of sending 40 bits of data, 123456789A$_H$:

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8D.
- To transmit the data 123456789A$_H$ with MSB first, write the following sequence to CSIGnTX0W:
  - 2000 1234$_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - 2000 5678$_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - 0000 009A$_H$ (CSIGnTX0W.CSIGnEDL = 0)

The following figure illustrates the timing.

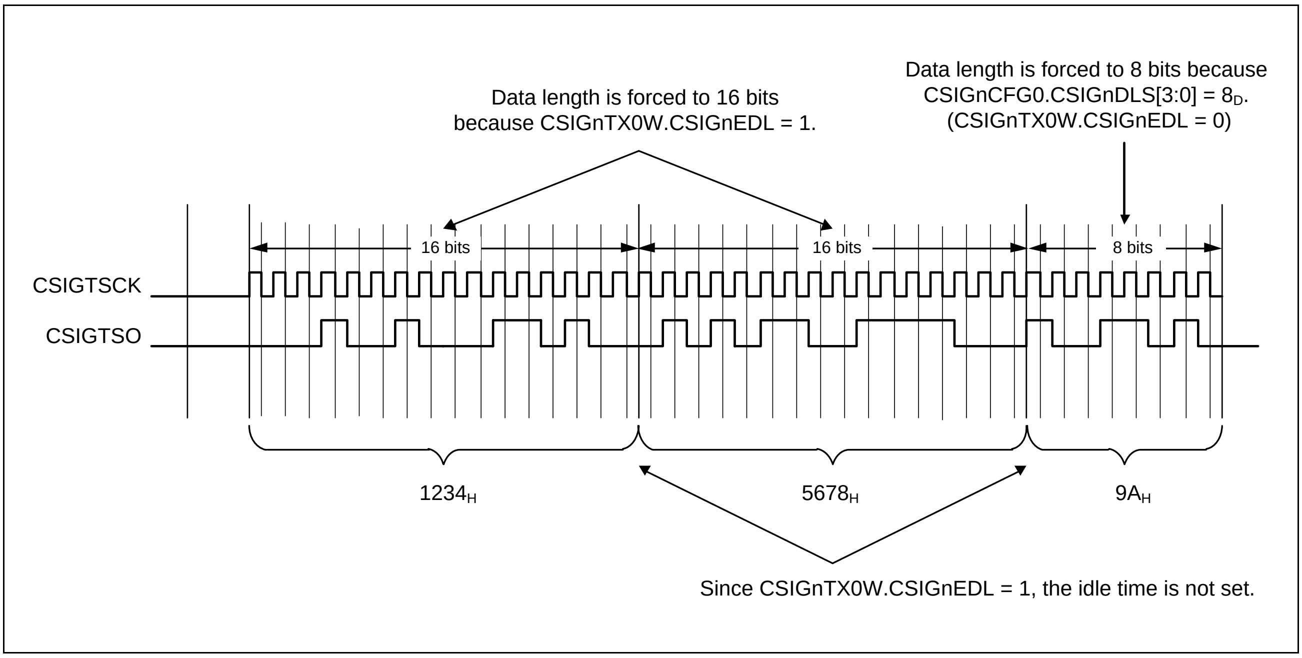


Figure 19.12 EDL Timing Diagram

NOTES

1. Data length with less than 7 bits can be set only when EDL mode is used.
2. It is not possible to transmit two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. Example for setting the data direction:
    - Data to be transmitted: 123456_H
    - MSB first:
      Set CSIGnCFG0.CSIGnDIR to 0.
      Write 2000 1234_H to CSIGnTX0W (EDL bit = 1).
      Write 0000 0056_H to CSIGnTX0W (EDL bit = 0).
    - LSB first:
      Set CSIGnCFG0.CSIGnDIR to 1.
      Write 2000 3456_H to CSIGnTX0W (EDL bit = 1).
      Write 0000 0012_H to CSIGnTX0W (EDL bit = 0).
5. EDL mode cannot be used in the slave mode configured for receive-only mode.
   (CSIGnCTL2.CSIGnPRS[2:0] = 111_B, CSIGnCTL0.CSIGnTXE = 0, CSIGnCTL0.CSIGnRXE = 1)

19.5.6 Serial Data Direction Selection Function

The serial data direction is selectable using the CSIGnDIR bit in the CSIGnCFG0 register. The examples below show the communication for 8-bit data (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B):

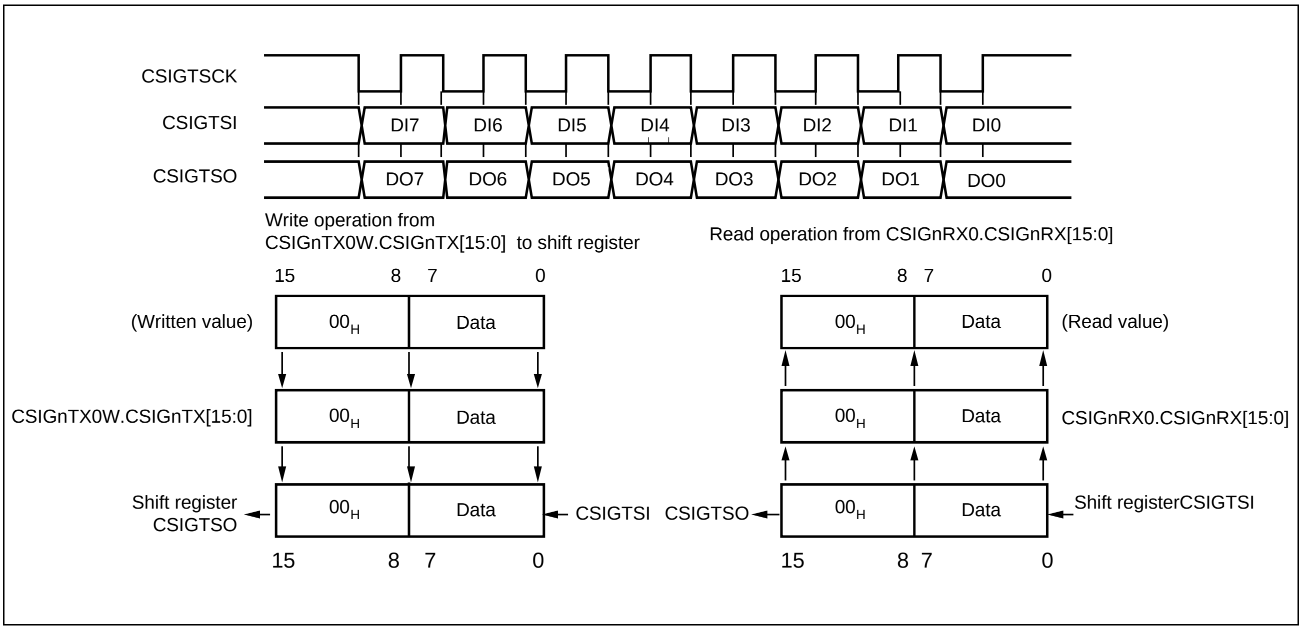


Figure 19.13 Serial Data Direction Select Function — MSB First (CSIGnDIR = 0)

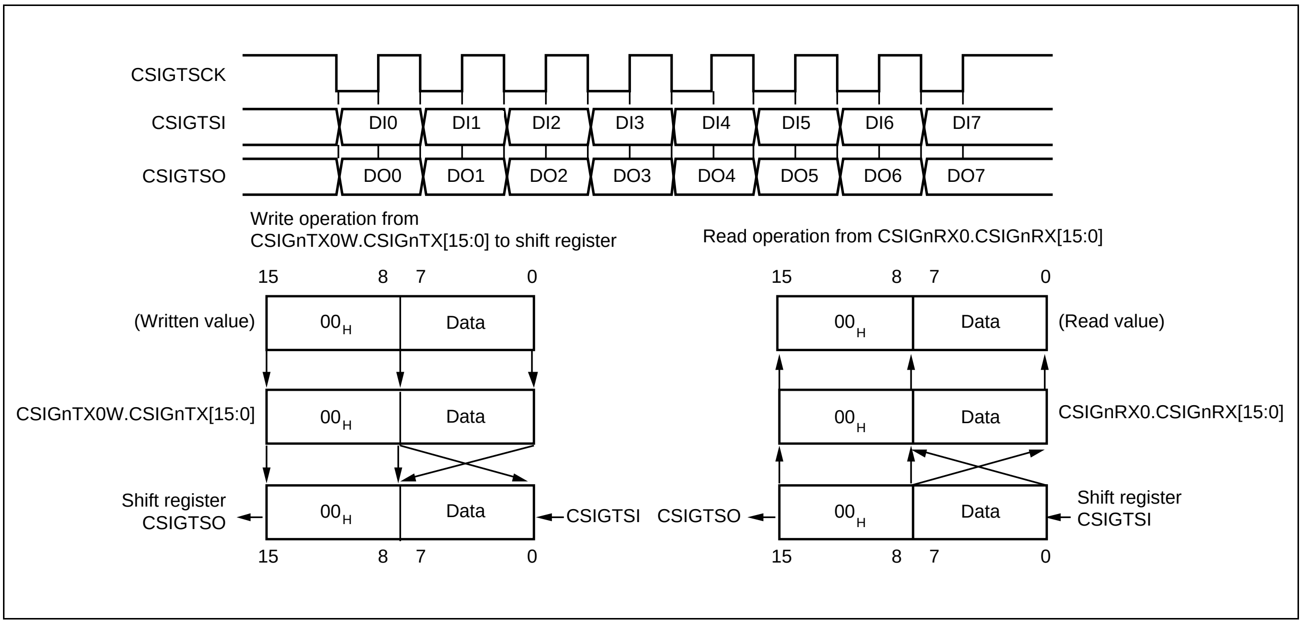


Figure 19.14 Serial Data Direction Select Function — LSB First (CSIGnDIR = 1)

19.5.7 Communication in Slave Mode

The following figure illustrates the communication signals and timing in slave mode.

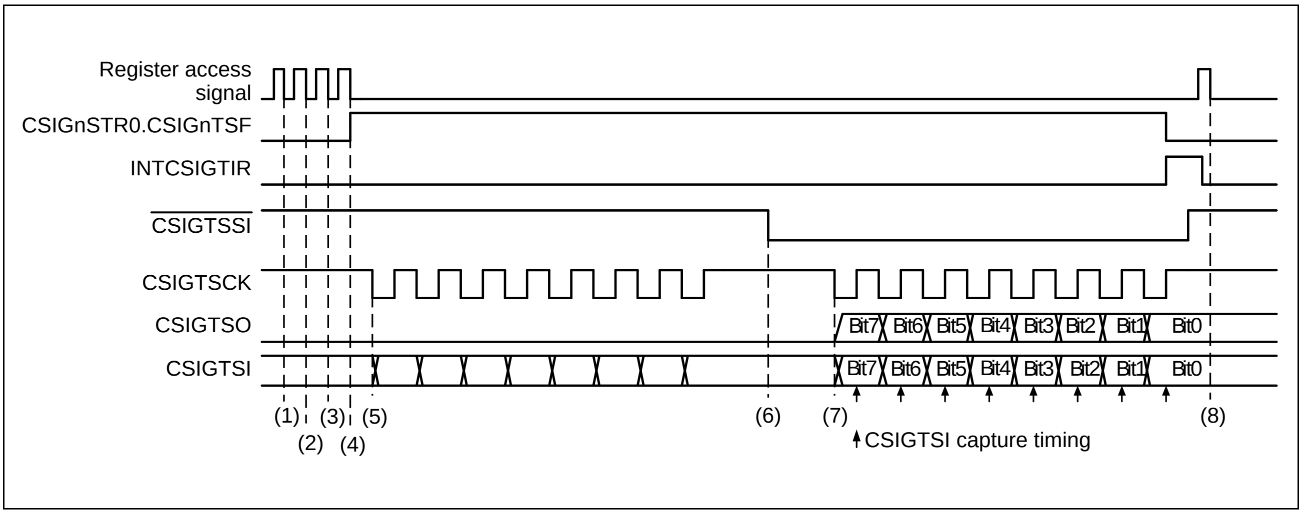


Figure 19.15 Rx/Tx Communication Timing in Slave Mode

1. Slave mode is selected (CSIGnCTL2.CSIGnPRS[2:0] = 111_{B}), the $\overline{\text{CSIGTSSI}}$ signal is enabled (CSIGnCTL1.CSIGnSSE = 1) and the clock phase is set to the high level (CSIGnCTL1.CSIGnCKR = 0).
2. Data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_{B}).
   Data direction is set to MSB first (CSIGnCFG0.CSIGnDIR = 0).
3. CSIG is set to transmit/receive mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, CSIGnCTL0.CSIGnRXE = 1).
4. When transfer data is written to the transmission register CSIGnTX0H, the transfer status flag CSIGnSTR0.CSIGnTSF is automatically set and the CSIG waits until signal $\overline{\text{CSIGTSSI}}$ goes low.
5. While signal $\overline{\text{CSIGTSSI}}$ is high, transmission/reception is not started even if the serial clock is input. CSIGTSO retains the values and input at CSIGTSI is ignored.
6. As soon as $\overline{\text{CSIGTSSI}}$ falls to low level, CSIGTSO is enabled.
7. If the serial clock is input to the CSIG while $\overline{\text{CSIGTSSI}}$ is low, transfer data is sent to CSIGTSO in synchronization with the serial clock, and simultaneously data is received from CSIGTSI.
8. The register CSIGnRX0 is read.

19.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIGnCTL1.CSIGnHSE bit. For handshake, the signals CSIGTRYI and CSIGTRYO are used.

The busy timing depends on the setting of the data phase selection bit CSIGnCFG0.CSIGnDAP.

19.5.8.1 Slave Mode

If CSIGnCTL1.CSIGnHSE = 1, a low-level CSIGTRYO signal is output when the slave becomes busy. This happens when previously received data is still in the CSIGnRX0 register, and the new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

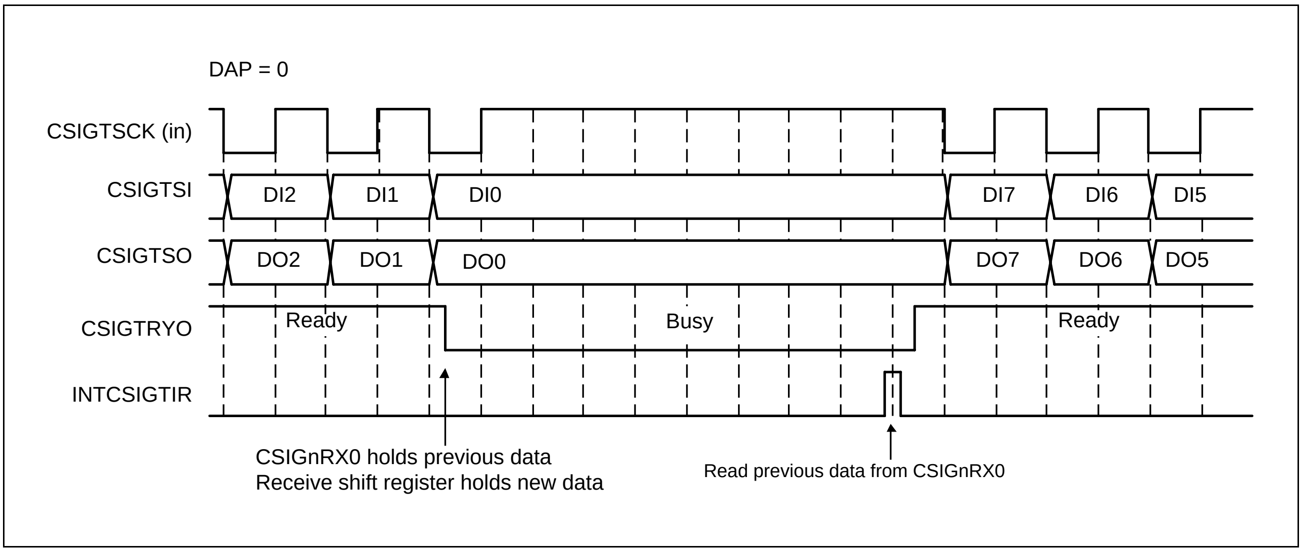


Figure 19.16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

While the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGTRYO to high (“ready”) as soon as the read from the reception register CSIGnRX0 completes.

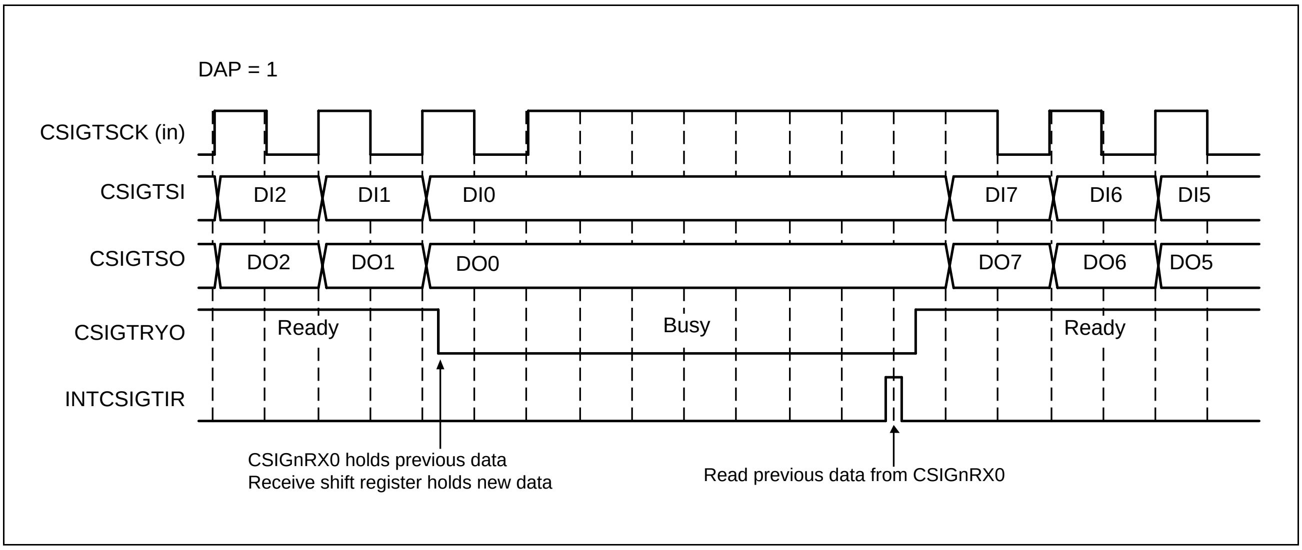


Figure 19.17 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

19.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI while CSIGnCTL1.CSIGnHSE = 1, subsequent transfers are put on hold, the master goes into wait state and suspends the CSIGTSCK clock.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

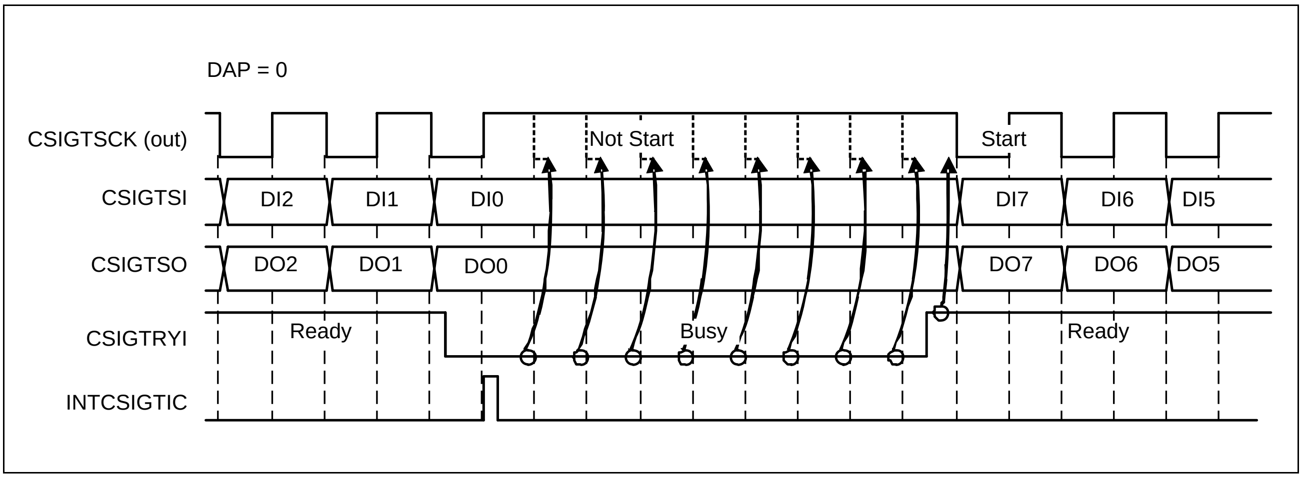


Figure 19.18 Master’s Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGTRYI low signal from the slave is received while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGTRYI becomes high (slave is "ready").

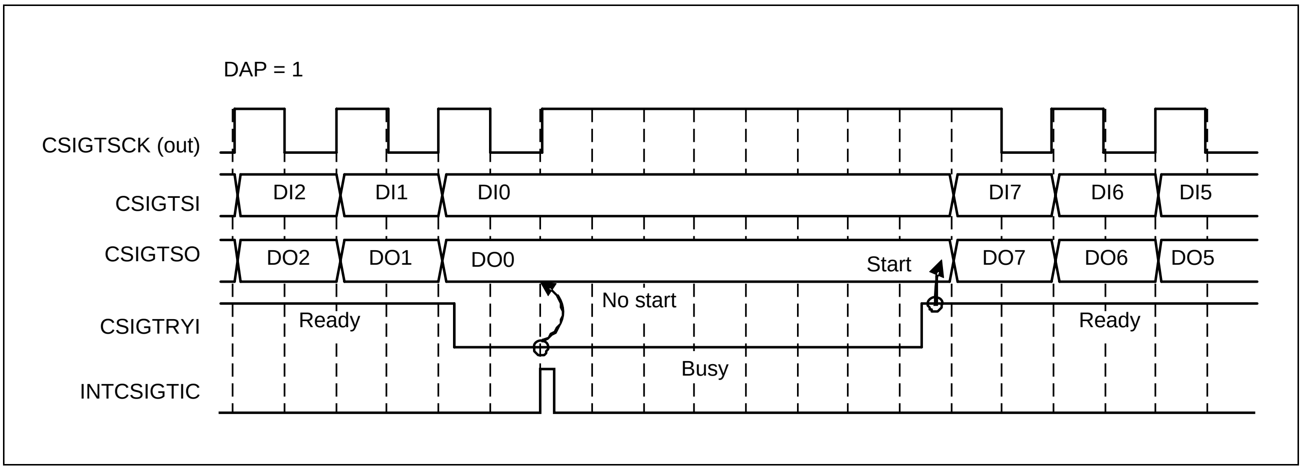


Figure 19.19 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 1)

CAUTION

If multiple slaves are connected, the master must only detect the CSIGTRYI signal of the slave it has selected for communication.

CSIGTRYI of the master must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will continue until it completes.

19.5.9 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIGnCTL1.CSIGnLBM = 1), the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIGTSCK, CSIGTSO, and CSIGTSI are disconnected from the ports. In addition, the CSIGTSO output level is fixed to low, and CSIGTSCK is set to reset level (High). The rest of CSIG works as in normal operation.

In order to test the CSIG, set the loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

Table 19.38 Output Level of Pins

Pin	Output Level
CSIGTSCK(out)	High level
CSIGTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIGTRYO	Normal function (Low level)

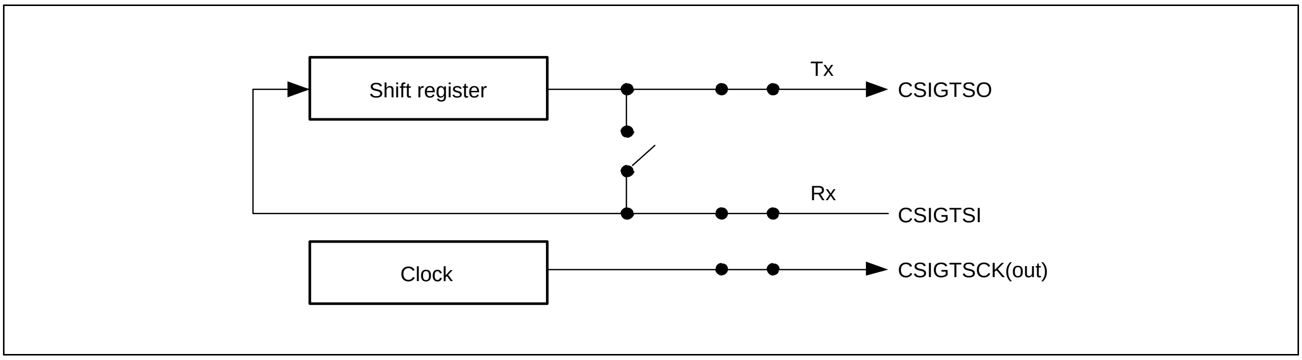


Figure 19.20 Normal Operation

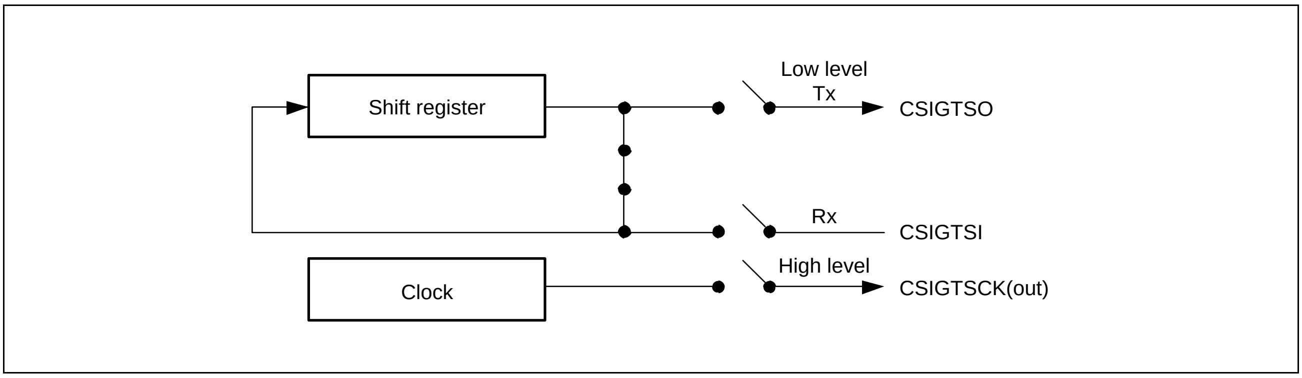


Figure 19.21 Operation in Loop-Back Mode

19.5.10 Error Detection

CSIG can detect three error types:

- Data consistency check error (transmission data)
- Parity error (received data)
- Overrun error (received data)

Data consistency check error and parity error check functions can be individually enabled or disabled.

If one of these errors is detected, the interrupt INTCSIGTIRE is generated.

19.5.10.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as an output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIGnCTL1.CSIGnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn_m = 1 for CSIGTSO). It will not be enabled if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is enabled, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIGTSO are captured and their logical interpretation is written to the corresponding shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency check error and:

- Interrupt INTCSIGTIRE is generated.
- The CSIGnSTR0.CSIGnDCE bit is set.

The data consistency check function is illustrated in the following block diagram.

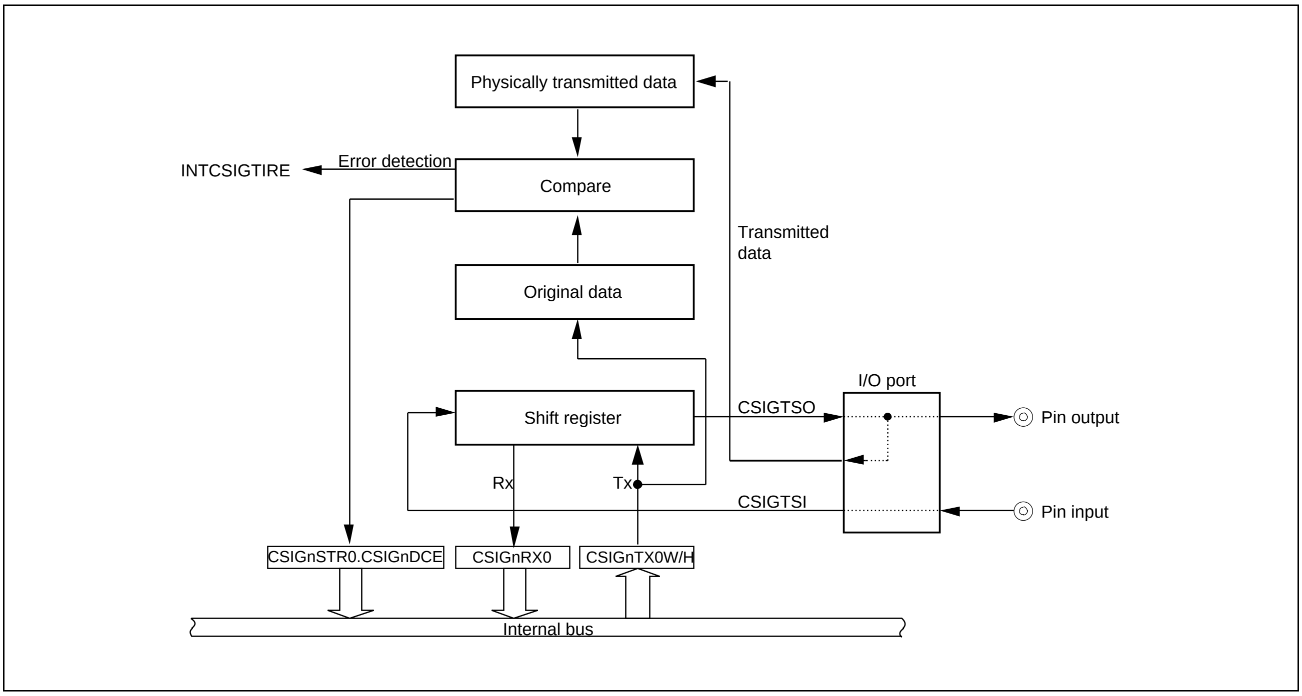


Figure 19.22 Functional Block Diagram of the Data Consistency Check

19.5.10.2 Parity Check

Parity is a common mean to detect a single bit error during data transmission. CSIG can append a parity bit after the last data bit (even if extended data length is used).

The use and type of parity is specified by CSIGnCFG0.CSIGnPS[1:0].

Parity check is enabled if CSIGnCFG0.CSIGnPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt INTCSIGTIRE is generated.
- The CSIGnSTR0.CSIGnPE bit is set.

The following figure shows an example.

Data length is 8 bits. The data transmitted is 05_H and 35_H. Parity type is odd.

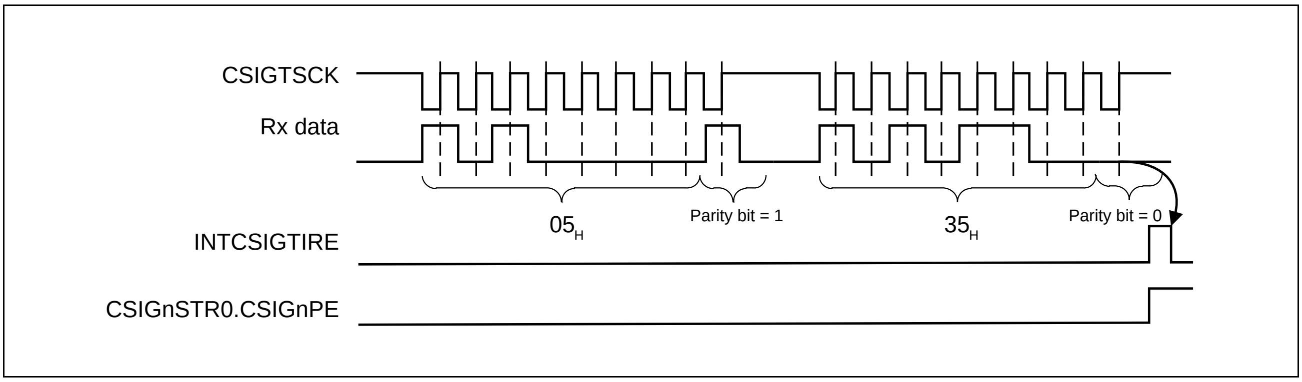


Figure 19.23 Parity Check Example

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last data bit.

19.5.10.3 Overrun Error

This error occurs when previously received data still resides in the reception register CSIGnRX0 because it has not been read, and new data is received.

The overrun error is not generated if data reception is disabled (CSIGnCTL0.CSIGnRXE = 0).

If overrun occurs:

- Interrupt INTCSIGTIRE is generated
- The CSIGnSTR0.CSIGnOVE bit is set
- Data in the CSIGnRX0 register is overwritten and communication continues.

The following figure illustrates the overrun error detection function.

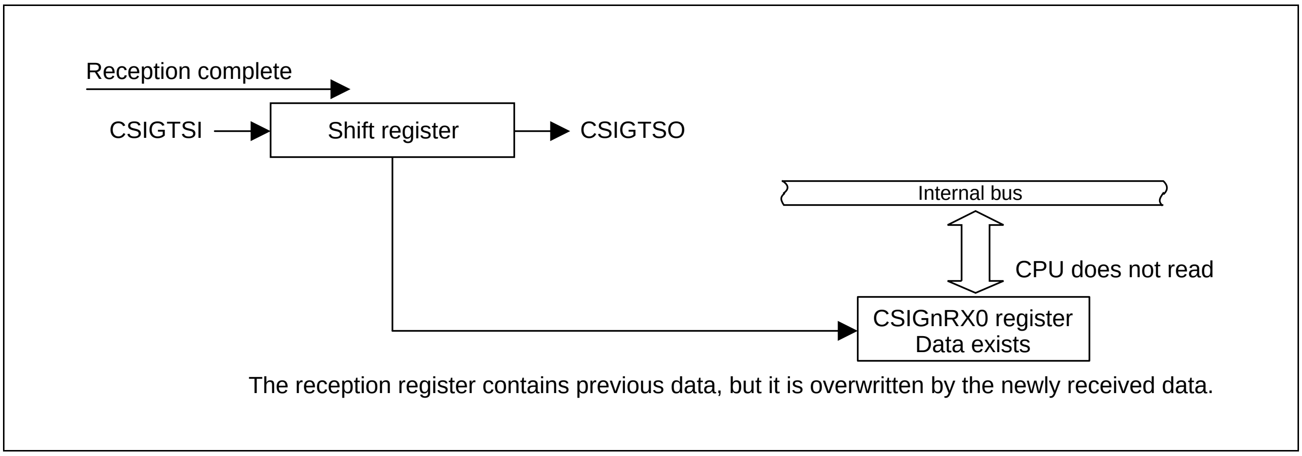


Figure 19.24 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 is not read
- Rx data 4 is received, and data is overwritten.

Thus an overrun error occurs.

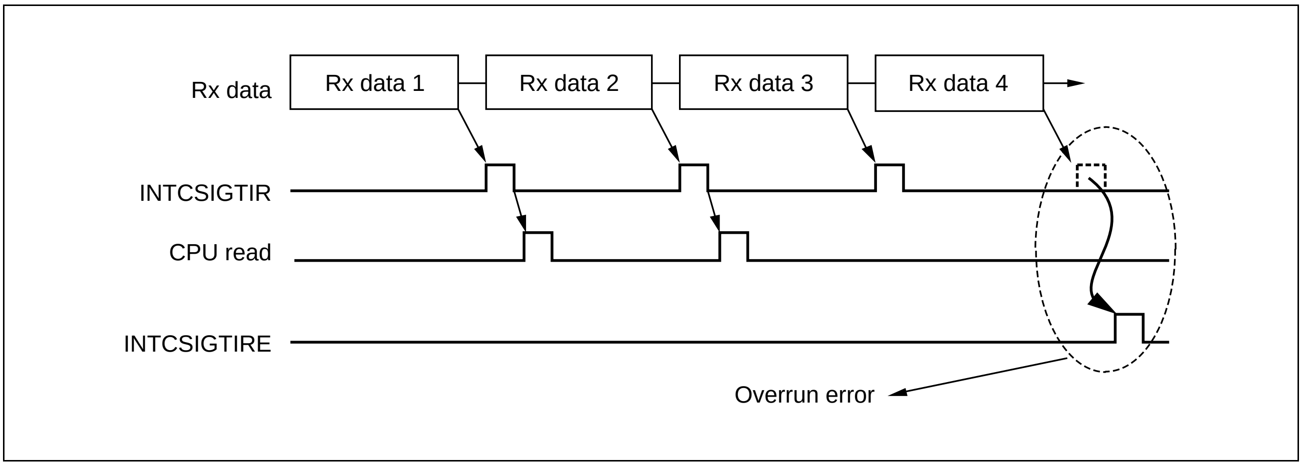


Figure 19.25 Overrun Error Detection — Example

NOTE

An overrun error can be avoided by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

For details see **Section 19.5.8, Handshake Function**.

19.6 Operating Procedures

19.6.1 Master Mode Transmission/Reception by DMA

This section describes an example of performing the transmission/reception in master mode in combination with a DMA.

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- MSB is transmitted first (CSIGnCFG0.CSIGnDIR = 0)
- INTCSIGTIC interrupt is generated at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- The number of data is 10 (0 to 9)

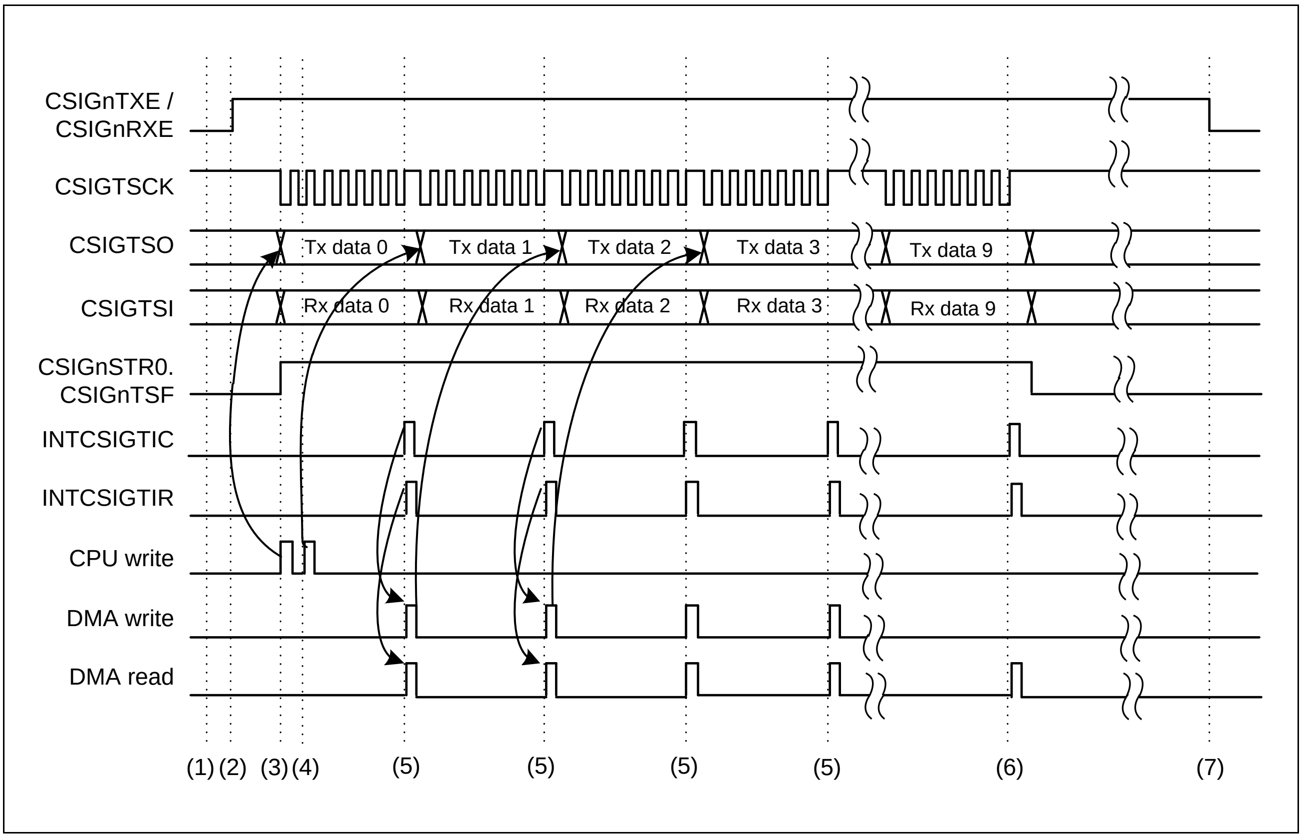


Figure 19.26 Communication in Master Mode

Procedure:

1. Configure the communication protocol in register CSIGnCFG0. Specify the interrupt timing and operation mode by setting the corresponding bits of the CSIGnCTL1 register and CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set CSIGnPWR = 1 (enable clock), CSIGnTXE = 1 (enable transmission), and CSIGnRXE = 1 (enable reception).
3. Write the first data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIGnTX0H. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After the transmission or reception of each data, INTCSIGTIC or INTCSIGTIR interrupt is generated. INTCSIGTIC indicates that the next data can be written to CSIGnTX0H. INTCSIGTIR indicates that the reception register CSIGnRX0 must be read.
   In this example, CPU write and DMA write are assumed as equivalent.
6. No more write action is required after the transmission of data 8 completes. Data 9 (the last data) has been written after the transmission of data 7. However, the reception register CSIGnRX0 must be read after the reception of data 8 and 9 completes.
7. Finally, to disable the transmit/receive operation, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE. When no communication is taking place, set CSIGnCTL0.CSIGnPWR to “0” to minimize the power consumption of the CSIGn.

Section 20 Clocked Serial Interface H (CSIH)

This section contains a generic description of the Clocked Serial Interface H (CSIH).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of CSIH.

20.1 Features of RH850/F1KH, RH850/F1KM CSIH

20.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has single channel interface.

Table 20.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of units	5	5	5
Name	CSIHn (n = 0 to 4)	CSIHn (n = 0 to 4)	CSIHn (n = 0 to 4)

Table 20.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of units	4	4	4	4	4
Name	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)

Table 20.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of units	1	1	3	4
Name	CSIHn (n = 0)	CSIHn (n = 0)	CSIHn (n = 0 to 2)	CSIHn (n = 0 to 3)

Table 20.4 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual CSIH units are identified by the index “n”: for example, CSIHnCTL0 (n = 0 to 4) is the CSIHn control register 0.
x	CSIHn has a maximum of 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index “x”: that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index “y”: for example, CSIHnBRSy is the baud rate setting register of CSIHn.

Table 20.5 Indices (RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual CSIH units are identified by the index “n”: for example, CSIHnCTL0 (n = 0 to 3) is the CSIHn control register 0.
x	CSIHn has a maximum of 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index “x”: that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index “y”: for example, CSIHnBRSy is the baud rate setting register of CSIHn.

The following table shows values indicated by the indices of each product.

Table 20.6 Indices of Products (RH850/F1KH-D8)

Indices of Each Product		
176 Pins	233 Pins	324 Pins
For the value of x, see **Table 20.9, Number of Chip Select Signals (RH850/F1KH-D8)**.		
y = 0 to 3		

Table 20.7 Indices of Products (RH850/F1KM-S4)

Indices of Each Product				
100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
For the value of x, see **Table 20.10, Number of Chip Select Signals (RH850/F1KM-S4)**.				
y = 0 to 3				

Table 20.8 Indices of Products (RH850/F1KM-S1)

Indices of Each Product			
48 Pins	64 Pins	80 Pins	100 Pins
For the value of x, see **Table 20.11, Number of Chip Select Signals (RH850/F1KM-S1)**.			
y = 0 to 3			

The numbers of chip select signals for each of the CSIH units are listed in the following table.

Table 20.9 Number of Chip Select Signals (RH850/F1KH-D8)

	Chip Select Index		
Unit Name	176 Pins	233 Pins	324 Pins
CSIH0	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)
CSIH1	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH2	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH3	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)
CSIH4	CSx (x = 0,1)	CSx (x = 0,1)	CSx (x = 0,1)

Table 20.10 Number of Chip Select Signals (RH850/F1KM-S4)

	Chip Select Index				
Unit Name	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
CSIH0	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)
CSIH1	CSx (x = 0 to 2, 4, 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH2	CSx (x = 0 to 3)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH3	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)

Table 20.11 Number of Chip Select Signals (RH850/F1KM-S1)

	Chip Select Index			
Unit Name	48 Pins	64 Pins	80 Pins	100 Pins
CSIH0	CSx (x = 0, 1)	CSx (x = 0 to 3)	CSx (x = 0 to 7)	CSx (x = 0 to 7)
CSIH1	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 5)
CSIH2	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 3)
CSIH3	—	—	—	CSx (x = 0 to 3)

20.1.2 Register Base Addresses

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses.

Table 20.12 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<CSIH0_base>	FFD8 0000_H
<CSIH1_base>	FFD8 2000_H
<CSIH2_base>	FFD8 4000_H
<CSIH3_base>	FFD8 6000_H
<CSIH4_base>	FFD9 0000_H

Table 20.13 Register Base Addresses (RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<CSIH0_base>	FFD8 0000_H
<CSIH1_base>	FFD8 2000_H
<CSIH2_base>	FFD8 4000_H
<CSIH3_base>	FFD8 6000_H

20.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

Table 20.14 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
CSIHn	PCLK	CKSCLK_ICSI	Communication clock
	Register access clock	CPUCLK_L, CKSCLK_ICSI	Bus clock

20.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 20.15 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
CSIH0			
INTCSIHTIC	Communication status interrupt	29	70
INTCSIHTIR	Receive status interrupt	30	71
INTCSIHTIRE	Communication error interrupt	31	—
INTCSIHTIJC	Job completion interrupt	20, 32	72
CSIH1			
INTCSIHTIC	Communication status interrupt	16, 116	28
INTCSIHTIR	Receive status interrupt	17, 117	29
INTCSIHTIRE	Communication error interrupt	27, 118	—
INTCSIHTIJC	Job completion interrupt	28, 119	30
CSIH2			
INTCSIHTIC	Communication status interrupt	8, 132	89
INTCSIHTIR	Receive status interrupt	10, 133	90
INTCSIHTIRE	Communication error interrupt	11, 134	—
INTCSIHTIJC	Job completion interrupt	12, 135	91
CSIH3			
INTCSIHTIC	Communication status interrupt	9, 158	41
INTCSIHTIR	Receive status interrupt	13, 159	42
INTCSIHTIRE	Communication error interrupt	14, 160	—
INTCSIHTIJC	Job completion interrupt	15, 161	43
CSIH4			
INTCSIHTIC	Communication status interrupt	124	58
INTCSIHTIR	Receive status interrupt	125	59
INTCSIHTIRE	Communication error interrupt	126	—
INTCSIHTIJC	Job completion interrupt	127	76

Table 20.16 Interrupt Requests (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
CSIH0			
INTCSIHTIC	Communication status interrupt	29	70
INTCSIHTIR	Receive status interrupt	30	71
INTCSIHTIRE	Communication error interrupt	31	—
INTCSIHTIJC	Job completion interrupt	20, 32	72
CSIH1			
INTCSIHTIC	Communication status interrupt	16, 116	28
INTCSIHTIR	Receive status interrupt	17, 117	29
INTCSIHTIRE	Communication error interrupt	27, 118	—
INTCSIHTIJC	Job completion interrupt	28, 119	30
CSIH2			
INTCSIHTIC	Communication status interrupt	8, 132	89
INTCSIHTIR	Receive status interrupt	10, 133	90
INTCSIHTIRE	Communication error interrupt	11, 134	—
INTCSIHTIJC	Job completion interrupt	12, 135	91
CSIH3			
INTCSIHTIC	Communication status interrupt	9, 158	41
INTCSIHTIR	Receive status interrupt	13, 159	42
INTCSIHTIRE	Communication error interrupt	14, 160	—
INTCSIHTIJC	Job completion interrupt	15, 161	43

20.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

Table 20.17 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
CSIHn	All reset sources (ISORES)

20.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 20.18 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
CSIH0		
CSIHTSCK	Serial clock signal	CSIH0SC*2
CSIHTSI	Serial data input signal	CSIH0SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH0SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH0RYI
CSIHTSO	Serial data output signal	CSIH0SO*2
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS[7:0]*1	Chip select signal	CSIH0CSS[7:0]*1
CSIH1		
CSIHTSCK	Serial clock signal	CSIH1SC*2
CSIHTSI	Serial data input signal	CSIH1SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH1SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH1RYI
CSIHTSO	Serial data output signal	CSIH1SO*2
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH1CSS[5:0]*1
CSIH2		
CSIHTSCK	Serial clock signal	CSIH2SC*2
CSIHTSI	Serial data input signal	CSIH2SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH2SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH2RYI
CSIHTSO	Serial data output signal	CSIH2SO*2
CSIHTRYO	Ready/busy output signal	CSIH2RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH2CSS[5:0]*1
CSIH3		
CSIHTSCK	Serial clock signal	CSIH3SC*2
CSIHTSI	Serial data input signal	CSIH3SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH3SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH3RYI
CSIHTSO	Serial data output signal	CSIH3SO*2
CSIHTRYO	Ready/busy output signal	CSIH3RYO
CSIHTCSS[3:0]*1	Chip select signal	CSIH3CSS[3:0]*1
CSIH4		
CSIHTSCK	Serial clock signal	CSIH4SC*2
CSIHTSI	Serial data input signal	CSIH4SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH4SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH4RYI
CSIHTSO	Serial data output signal	CSIH4SO*2
CSIHTRYO	Ready/busy output signal	CSIH4RYO
CSIHTCSS[1:0]*1	Chip select signal	CSIH4CSS[1:0]*1

Note 1. For the number of chip select signals, see **Table 20.9, Number of Chip Select Signals (RH850/F1KH-D8)**.

Note 2. For the port pins that are used as CSIHnSO and CSIHnSC, set the output driver strength to high (PDSCn_m = 1).

Table 20.19 External Input/Output Signals (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
CSIH0		
CSIHTSCK	Serial clock signal	CSIH0SC*2
CSIHTSI	Serial data input signal	CSIH0SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH0SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH0RYI
CSIHTSO	Serial data output signal	CSIH0SO*2
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS[7:0]*1	Chip select signal	CSIH0CSS[7:0]*1
CSIH1		
CSIHTSCK	Serial clock signal	CSIH1SC*2
CSIHTSI	Serial data input signal	CSIH1SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH1SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH1RYI
CSIHTSO	Serial data output signal	CSIH1SO*2
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH1CSS[5:0]*1
CSIH2		
CSIHTSCK	Serial clock signal	CSIH2SC*2
CSIHTSI	Serial data input signal	CSIH2SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH2SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH2RYI
CSIHTSO	Serial data output signal	CSIH2SO*2
CSIHTRYO	Ready/busy output signal	CSIH2RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH2CSS[5:0]*1
CSIH3		
CSIHTSCK	Serial clock signal	CSIH3SC*2
CSIHTSI	Serial data input signal	CSIH3SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH3SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH3RYI
CSIHTSO	Serial data output signal	CSIH3SO*2
CSIHTRYO	Ready/busy output signal	CSIH3RYO
CSIHTCSS[3:0]*1	Chip select signal	CSIH3CSS[3:0]*1

Note 1. For the number of chip select signals, see **Table 20.10, Number of Chip Select Signals (RH850/F1KM-S4)**, **Table 20.11, Number of Chip Select Signals (RH850/F1KM-S1)**.

Note 2. For the port pins that are used as CSIHnSO and CSIHnSC, set the output driver strength to high (PDSCn_m = 1).

CAUTION

When port P8_6 is used as CSIH0CSS4, port P8_6 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**, **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

20.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSIHnSO (CSIHTSO) output are shown in the following table. See **Section 20.5.12, Error Detection** for details on data consistency checking.

Table 20.20 Port Pins for Data Consistency Checking (RH850/F1KH-D8)

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P0_3	ALT_OUT4
CSIH1		
CSIHTSO	P0_5	ALT_OUT3
	P10_2	ALT_OUT5
CSIH2		
CSIHTSO	P11_2	ALT_OUT1
CSIH3		
CSIHTSO	P11_6	ALT_OUT3
CSIH4		
CSIHTSO	P2_4	ALT_OUT3
	P23_1	ALT_OUT1

Table 20.21 Port Pins for Data Consistency Checking (RH850/F1KM-S4)

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P0_3	ALT_OUT4
CSIH1		
CSIHTSO	P0_5	ALT_OUT3
	P10_2	ALT_OUT5
CSIH2		
CSIHTSO	P11_2	ALT_OUT1
CSIH3		
CSIHTSO	P11_6	ALT_OUT3

Table 20.22 Port Pins for Data Consistency Checking (RH850/F1KM-S1)

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P0_3	ALT_OUT4
CSIH1		
CSIHTSO	P0_5	ALT_OUT3
	P10_2	ALT_OUT5
CSIH2		
CSIHTSO	P11_2	ALT_OUT1
CSIH3		
CSIHTSO	P11_6	ALT_OUT3

20.2 Overview

20.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable
- Multiple slaves configuration and RCB (Recessive Configuration for Broadcasting) are possible since there are to eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is usable
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6) for all products
  - Slave mode: 8.0 MHz (however, it must be equal to or lower than PCLK/6) for CSIH0 in RH850/F1KM-S1 48- and 64-pin products, CSIH2 for other products
- Clock phases and data phases are selectable
- Data transfer with MSB first or LSB first is selectable
- Transfer data length is selectable from 2 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, time-out, overflow, and overrun)
- Support of job concept
- 128 words I/O buffer memory
- Direct access mode or memory mode (FIFO, dual buffer, and transmit-only buffer) is selectable
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- Built-in LBM (Loop Back Mode) function for self-test
- CPU-controlled high-priority communication function
- Enforced chip select idle setting
- Built-in RCB (Recessive Configuration for Broadcasting) bit
- Built-in JOB enable control bit for AUTOSAR

20.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIHTSCK (output in master mode, input in slave mode)
- Data output signal CSIHTSO
- Data input signal CSIHTSI

Additional signals are available for external control and monitoring:

- $\overline{\text{CSIHTSSI}}$: Slave select input signal
- CSIHTRYO: Ready/busy output signal (handshake signal)
- CSIHTRYI: Ready/busy input signal (handshake signal)
- CSIHTCSS[7:0]: Chip select signals

Data transmission is bit-wise and serial, and performed synchronously with the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Table 20.23 Main Registers of CSIH

Register	Function
CSIHnCTL0	Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables (bypasses) buffering.
CSIHnCTL1	Controls options such as interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIHnBRSy	Specifies the transfer clock frequency for each chip select signal.
CSIHnMCTL0	Selects memory mode and specifies the time-out value.
CSIHnMCTL1	Controls the memory in FIFO mode.
CSIHnMCTL2	Controls the memory in dual buffer mode.
CSIHnCFGx	Configures the communication protocol for each chip select signal.

20.2.3 Block Diagram

The following block diagram shows the main components of the CSIH.

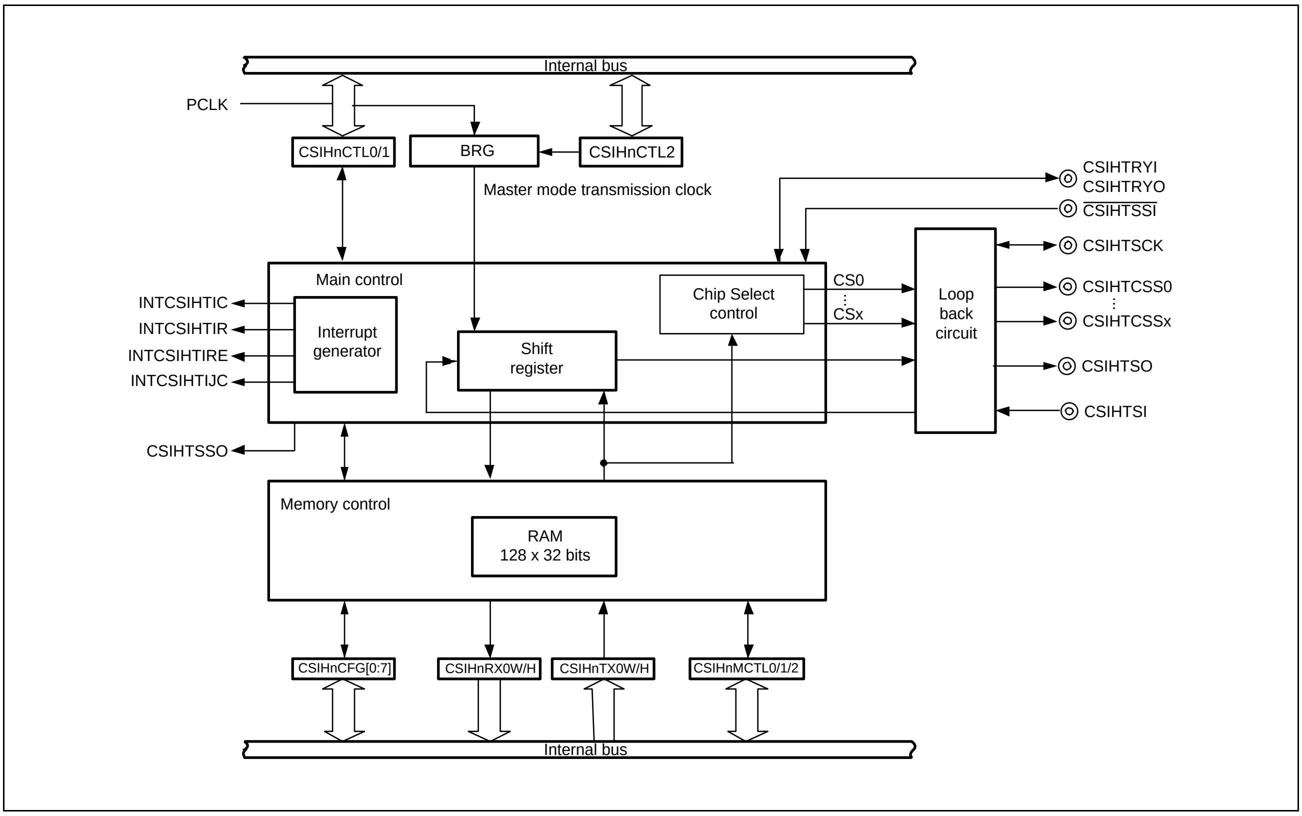


Figure 20.1 CSIH Block Diagram

In master mode, the transmission clock CSIHTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self-test.

NOTE

This section describes the following modes:

- The “operating mode” is either master mode or slave mode. The master can control and communicate with several slaves only in master mode (for details, see **Section 20.5.1, Operating Modes (Master/Slave)).**
- The “job mode” is related to the AUTOSAR job concept (for details, see **Section 20.5.3.3, Job Concept**).
- The “memory mode” accords with the various configurations of the associated buffer memory (for details, see **Section 20.5.6, CSIH Buffer Memory**).
- The “data transfer mode” specifies the type of communication – transmit-only, receive-only, or transmit/receive (for details, see **Section 20.5.7, Data Transfer Modes**).

20.3 Registers

20.3.1 List of Registers

CSIH registers are listed in the following table.

For details about <CSIHn_base>, see **Section 20.1.2, Register Base Addresses**.

Table 20.24 List of Registers

Module Name	Register Name	Symbol	Address
CSIHn	CSIHn control register 0	CSIHnCTL0	<CSIHn_base> + 0000_H
	CSIHn control register 1	CSIHnCTL1	<CSIHn_base> + 0010_H
	CSIHn control register 2	CSIHnCTL2	<CSIHn_base> + 0014_H
	CSIHn status register 0	CSIHnSTR0	<CSIHn_base> + 0004_H
	CSIHn status clear register 0	CSIHnSTCR0	<CSIHn_base> + 0008_H
	CSIHn memory control register 0	CSIHnMCTL0	<CSIHn_base> + 1040_H
	CSIHn memory control register 1	CSIHnMCTL1	<CSIHn_base> + 1000_H
	CSIHn memory control register 2	CSIHnMCTL2	<CSIHn_base> + 1004_H
	CSIHn memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base> + 1018_H
	CSIHn configuration register 0	CSIHnCFG0	<CSIHn_base> + 1044_H
	CSIHn configuration register 1	CSIHnCFG1	<CSIHn_base> + 1048_H
	CSIHn configuration register 2	CSIHnCFG2	<CSIHn_base> + $104C_H$
	CSIHn configuration register 3	CSIHnCFG3	<CSIHn_base> + 1050_H
	CSIHn configuration register 4	CSIHnCFG4	<CSIHn_base> + 1054_H
	CSIHn configuration register 5	CSIHnCFG5	<CSIHn_base> + 1058_H
	CSIHn configuration register 6	CSIHnCFG6	<CSIHn_base> + $105C_H$
	CSIHn configuration register 7	CSIHnCFG7	<CSIHn_base> + 1060_H
	CSIHn transmit data register 0 for word access	CSIHnTX0W	<CSIHn_base> + 1008_H
	CSIHn transmit data register 0 for half word access	CSIHnTX0H	<CSIHn_base> + $100C_H$
	CSIHn receive data register 0 for word access	CSIHnRX0W	<CSIHn_base> + 1010_H
	CSIHn receive data register 0 for half word access	CSIHnRX0H	<CSIHn_base> + 1014_H
	CSIHn emulation register	CSIHnEMU	<CSIHn_base> + 0018_H
	CSIHn baud rate setting register 0	CSIHnBRS0	<CSIHn_base> + 1068_H
	CSIHn baud rate setting register 1	CSIHnBRS1	<CSIHn_base> + $106C_H$
	CSIHn baud rate setting register 2	CSIHnBRS2	<CSIHn_base> + 1070_H
	CSIHn baud rate setting register 3	CSIHnBRS3	<CSIHn_base> + 1074_H

20.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock, enables/disables transmission/reception, and enables/disables the memory allocated for transmission and/or reception. It forces the stop of communication at the end of the current job.

Access: This register can be read or written in 8-bit units.

Address: <CSIHn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnPWR	CSIHnTXE	CSIHnRXE	—	—	—	CSIHnJOBE	CSIHnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

Table 20.25 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. Clock supply to internal circuits stops. If CSIHnPWR is cleared (to 0) during communication, ongoing communication is immediately aborted. In this case, the communication setting must be reconfigured.
6	CSIHnTXE	Enables/disables transmission. 0: Disables transmission. 1: Enables transmission.
5	CSIHnRXE	Enables/disables reception. 0: Disables reception. 1: Enables reception.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIHnJOBE	Stops communication at the end of the current job (communication ends if data is written to the transmission buffer when CSIHnTX0W.CSIHnEOJ = 1 (job completion)). 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. Even if this bit is set to 1, the read value is always 0. In FIFO mode, the pointer must be cleared by setting CSIHnSTCR0.CSIHnPCT = 1 before the next communication is started
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data. 1: Direct access mode CSIH memory is bypassed.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CSIHn SLRS	—	—	—	—	—	CSIHn PHE	CSIHn CKR	CSIHn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn CSL7	CSIHn CSL6	CSIHn CSL5	CSIHn CSL4	CSIHn CSL3	CSIHn CSL2	CSIHn CSL1	CSIHn CSL0	CSIHn EDLE	CSIHn JE	CSIHn DCS	CSIHn CSRI	CSIHn LBM	CSIHn SIT	CSIHn HSE	CSIHn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.26 CSIHnCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CSIHnSLRS	Sets the internal synchronization timing for receive data input. 0: Rising edge of PCLK 1: Falling edge of PCLK For differences by the setting, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	CSIHnPHE	Sets the CPU-controlled priority-based communication function. 0: The CPU-controlled high-priority communication function is disabled. 1: The CPU-controlled high-priority communication function is enabled. To enable the CPU-controlled high-priority communication function, set this bit to 1 and set CSIHnJE = 1. This bit can only be set in transmit-only buffer mode.
17	CSIHnCKR	CSIHTSCK Clock Inversion Function 0: The default level of CSIHTSCK is high 1: The default level of CSIHTSCK is low For details, see **Section 20.3.11, CSIHnCFGx — CSIHn Configuration Register x (x = 0 to 7)**.
16	CSIHnSLIT	Selects the timing of interrupt INTCSIHTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated (this function is activated only in direct access mode/transmit-only buffer mode). For details, see **Section 20.4.3, INTCSIHTIC (Communication Status Interrupt)**.
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHTCSSx). 0: Chip select is active low. 1: Chip select is active high. For details, see **Section 20.5.3, Chip Selection (CS) Features**.

Table 20.26 CSIHnCTL1 Register Contents

Bit Position	Bit Name	Function
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see **Section 20.5.8.2, Data Length Greater than 16 Bits**.
6	CSIHnJE	Enables/disables job mode. 0: Disables job mode. 1: Enables job mode. For details, see **Section 20.5.3.3, Job Concept**. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W. CSIHnCIRE bits are enabled only when CSIHnJE = 1. Setting this bit in slave mode is prohibited. In addition, to enable the CPU-controlled high-priority communication function, set this bit to 1 as well as CSIHnPHE = 1.
5	CSIHnDCS	Enables/disables data consistency check. 0: Disables data consistency check. 1: Enables data consistency check. For details, see **Section 20.5.12.1, Data Consistency Check**.
4	CSIHnCSRI	Defines chip select signal behavior after last data transfer. 0: Chip select signal retains the active level. 1: Chip select signal returns to the inactive level. The last data is determined at the interrupt timing in direct access mode or FIFO mode. When CSIHnCTL1.CSIHnSLIT = 1, the last data is determined in direct access mode.
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Deactivates loop-back mode. 1: Activates loop-back mode. For details, see **Section 20.5.13, Loop-Back Mode**.
2	CSIHnSIT	Selects interrupt delay mode. 0: No delay is generated. 1: Half clock delay is generated for all interrupts. This bit is only valid in master mode. In slave mode, no delay is generated. For details, see **Section 20.4.2, Interrupt Delay**.
1	CSIHnHSE	Enables/disables the handshake function. 0: Disables the handshake function. 1: Enables the handshake function. For details see **Section 20.5.11, Handshake Function**.
0	CSIHnSSE	Enables/disables the slave select function. 0: Input signal $\overline{\text{CSIHTSSI}}$ is disabled. 1: Input signal $\overline{\text{CSIHTSSI}}$ is recognized. If the slave select function is not used, this bit must be set to 0 (see also **Section 20.5.2, Master/Slave Connections**).

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 20.27 Operation of the Slave Select Function during Reception

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Receive Operation
0	—	—	Reception is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 20.28 Operation of the Slave Select Function during Transmission

CSIHnCTL0.CSIHnTXE	CSIHnCTL1.CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Transmit Operation
0	—	—	Transmission is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the reference clock value, and specifies the transfer clock frequency.

For details see **Section 20.5.5, Transmission Clock Selection**.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + 0014$_H$

Value after reset: E000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnPRS[2:0]			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.29 CSIHnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIHnPRS[2:0]	These bits select the operation mode and the reference clock value.
12 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	Selection of Reference Clock (PRSOUT)
0	0	0	PCLK (Master mode)
0	0	1	PCLK/2 (Master mode)
0	1	0	PCLK/4 (Master mode)
0	1	1	PCLK/8 (Master mode)
1	0	0	PCLK/16 (Master mode)
1	0	1	PCLK/32 (Master mode)
1	1	0	PCLK/64 (Master mode)
1	1	1	External clock via CSIHTSCK(in) (Slave mode)

In master mode, the following bits are used to set the transfer clock frequency:

CSIHnCTL2.CSIHnPRS[2:0], CSIHnCFGx.CSIHnBRSS[1:0], CSIHnBRSy.CSIHnBRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSIHnBRSy.CSIHnBRS[11:0] bits is selected for each chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSIHnCFGx.CSIHnBRSS[1:0] bits.

The following table shows the relationship between CSIHnCFGx.CSIHnBRSS[1:0] and CSIHnBRSy.CSIHnBRS[11:0].

CSIHnCFGx.CSIHnBRSS[1:0]	Transfer Clock Frequency Setting Bit to be Selected
00	CSIHnBRS0.CSIHnBRS[11:0]
01	CSIHnBRS1.CSIHnBRS[11:0]
10	CSIHnBRS2.CSIHnBRS[11:0]
11	CSIHnBRS3.CSIHnBRS[11:0]

The following table shows the relationship between the transfer clock frequency and the transfer clock frequency setting (CSIHnBRSy[11:0]) selected by the CSIHnBRSS[1:0] bits when the bit value of the CSIHnPRS[2:0] bits is α.

CSIHnBRSy[11:0]	Transfer Clock Frequency
0	BRG stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
...	…
4095	PCLK / (2^{α} × 4095× 2)

When a time-out error is used in slave mode, the clock selected by this setting is used. In slave mode, the CSIHnPRS[2:0] bits are set to 111_B. In this case, the prescaler has the same setting as when the CSIHnPRS[2:0] bits are set to 000_B. When using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000_H.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIHn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnSRP[7:0]								CSIHnSPF[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOE	CSIHn OFE	—	—	—	—	—	CSIHn HPST	CSIHn TSF	—	CSIHn FLF	CSIHn EMF	CSIHn DCE	—	CSIHn PE	CSIHn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.30 CSIHnSTR0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 24</td><td>CSIHnSRP[7:0]</td><td>Indicates the number of received data in FIFO mode.
<table>
<tr><th>CSIHnSRP[7:0]</th><th>Description</th></tr>
<tr><td>00_H</td><td rowspan="3">Number of received data (0 to 128)</td></tr>
<tr><td>…</td></tr>
<tr><td>80_H</td></tr>
<tr><td>Other than the above</td><td>Undefined</td></tr>
</table>
These bits are cleared by CSIHnSTCR0.CSIHnPCT.
In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H.
In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].</td></tr>
<tr><td>23 to 16</td><td>CSIHnSPF[7:0]</td><td>Indicates the number of unsent data in FIFO mode.
(The number of data written by the CPU is the number of sent data.)
<table>
<tr><th>CSIHnSRP[7:0]</th><th>Description</th></tr>
<tr><td>00_H</td><td rowspan="3">Number of unsent data (0 to 128)</td></tr>
<tr><td>…</td></tr>
<tr><td>80_H</td></tr>
<tr><td>Other than the above</td><td>Undefined</td></tr>
</table>
These bits are cleared by CSIHnSTCR0.CSIHnPCT.
In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H.
In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].</td></tr>
</table>

Table 20.30 CSIHnSTR0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOE</td><td>Time-out Error Flag in FIFO Mode
Indicates whether a time-out error was detected in FIFO mode.
0: No time-out error was detected.
1: A time-out error was detected.
For details, see **Section 20.5.12.3, Time-Out Error**.
This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.
When setting to 1 by time-out error detection and clearing to 0 by CSIHnSTCR0.CSIHnTMOEC occur simultaneously, setting to 1 takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.</td></tr>
<tr><td>14</td><td>CSIHnOFE</td><td>Overflow Error Flag in FIFO mode
Indicates whether an overflow error was detected in FIFO mode.
0: No overflow error was detected.
1: An overflow error was detected.
For details, see **Section 20.5.12.4, Overflow Error**.
This bit is cleared by CSIHnSTCR0.CSIHnOFEC.
When setting to 1 by overflow error detection and clearing to 0 by CSIHnSTCR0.CSIHnOFEC occur simultaneously, setting to 1 takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.</td></tr>
<tr><td>13 to 9</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>8</td><td>CSIHnHPST</td><td>Communication Priority Indication Flag
0: Indicates low-priority communication is in progress.
1: Indicates high-priority communication is in progress.
This bit always reads 0 if CPU-controlled high-priority communication is disabled (CSIHnCTL1.CSIHnPHE = 0).</td></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer Status Flag
0: Idle state
1: Communication is in progress or being prepared.
The timing to set or clear this bit is as follows:
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Timing to Set</th><th rowspan="2">Timing to Clear</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Dual Buffer Mode, Transmit-Only Buffer Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="3">Data is written to a transmit register (CSIHnTX0W/ CSIHnTX0H)</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST bit is set</td><td rowspan="3">Within a half clock cycle the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave Mode</th><th colspan="2">Timing to Set</th><th rowspan="2">Timing to Clear</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Dual Buffer Mode, Transmit-Only Buffer Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="2">Data is written to a transmit register (CSIHnTX0W/ CSIHnTX0H)</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST bit is set</td><td rowspan="3">Within a half clock cycle the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td><td>Input timing of CSIHTSCK</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>5</td><td>CSIHnFLF</td><td>A flag indicating that the buffer is full in FIFO mode.
0: FIFO buffer is not full.
1: FIFO buffer is full.
This bit is cleared by CSIHnSTCR0.CSIHnPCT.
The FIFO buffer might be filled with unsent data or received data.</td></tr>
</table>

Table 20.30 CSIHnSTR0 Register Contents

Bit Position	Bit Name	Function
4	CSIHnEMF	A flag indicating that the buffer is empty in FIFO mode. 0: FIFO buffer is not empty. 1: FIFO buffer is empty. This bit is set to 1 by CSIHnSTCR0.CSIHnPCT. This bit is set to 1 when CSIHnSTR0.CSIHnSRP[7:0] + CSIHnSTR0.CSIHnSPF[7:0] = 00_H. The FIFO buffer might be filled with unsent data or received data.
3	CSIHnDCE	Data Consistency Check Error Flag 0: No data consistency check error is detected. 1: Data consistency check error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. When setting to 1 by data consistency check error detection and clearing to 0 by CSIHnSTCR0.CSIHnDCEC occur simultaneously, setting to 1 takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.
2	Reserved	When read, the value after reset is returned.
1	CSIHnPE	Parity Error Flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. When setting to 1 due to parity error detection and clearing to 0 by CSIHnSTCR0.CSIHnPEC occur simultaneously, setting to 1 by parity error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
0	CSIHnOVE	Overrun Error Flag (Fixed to 0 in dual buffer mode) 0: No overrun error is detected. 1: Overrun error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIHnSTCR0.CSIHnOVEC occur simultaneously, setting to 1 by overrun error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Table 20.31 Behaviors in Various Memory Modes

Bit Name	Bit Position	Direct Access Mode	FIFO Mode	Transmit-Only Buffer Mode	Dual Buffer Mode
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received data	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of untransmitted data packets	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Communication is in progress or being prepared			
CSIHnFLF	5	Fixed to 0	0: FIFO is not full 1: FIFO is full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: FIFO is not empty 1: FIFO is empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error is detected. 1: An error is detected.			
CSIHnPE	1	0: No error is detected. 1: An error is detected.			
CSIHnOVE	0	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	Fixed to 0

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access:	This register can be read or written in 16-bit units. When read, the value 0000$_H$ is always returned.
Address:	<CSIHn_base> + 0008$_H$
Value after reset:	0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOEC	CSIHn OFEC	—	—	—	—	—	CSIHn PCT	—	—	—	—	CSIHn DCEC	—	CSIHn PEC	CSIHn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Table 20.32 CSIHnSTCR0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOEC</td><td>Controls the time-out error flag clear command.
0: No operation. The read value is always 0.
1: Clears the time-out error flag (CSIHnSTR0.CSIHnTMOE).</td></tr>
<tr><td>14</td><td>CSIHnOFEC</td><td>Controls the overflow error flag clear command.
0: No operation. The read value is always 0.
1: Clears the overflow error flag (CSIHnSTR0.CSIHnOFE).</td></tr>
<tr><td>13 to 9</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>8</td><td>CSIHnPCT</td><td>Controls the FIFO pointer clear command.
0: No operation. The read value is always 0.
1: Clears the following FIFO buffer pointers (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and status bits.
<table>
<tr><th>FIFO Buffer Pointer</th><th>Status Bit</th></tr>
<tr><td>CSIHnMRWP0.CSIHnTRWA[6:0]
CSIHnMRWP0.CSIHnRRA[6:0]
CSIHnMCTL2.CSIHnSOP[6:0]</td><td>CSIHnSTR0.CSIHnSPF[7:0]
CSIHnSTR0.CSIHnSRP[7:0]
CSIHnSTR0.CSIHnFLF
CSIHnSTR0.CSIHnTSF</td></tr>
</table>
Additionally, the CSIHnSTR0.CSIHnEMF bit is set to 1 (FIFO empty) (in FIFO mode only).</td></tr>
<tr><td>7 to 4</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>3</td><td>CSIHnDCEC</td><td>Controls the data consistency check error flag clear command.
0: No operation. The read value is always 0.
1: Clears the data consistency check error flag (CSIHnSTR0.CSIHnDCE).</td></tr>
<tr><td>2</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>1</td><td>CSIHnPEC</td><td>Controls the parity error flag clear command.
0: No operation. The read value is always 0.
1: Clears the parity error flag (CSIHnSTR0.CSIHnPE).</td></tr>
<tr><td>0</td><td>CSIHnOVEC</td><td>Controls the overrun error flag clear command.
0: No operation. The read value is always 0.
1: Clears the overrun error flag (CSIHnSTR0.CSIHnOVE).</td></tr>
</table>

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + 1040_H

Value after reset: $001F_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CSIHn MMS[1:0]		—	—	—	CSIHnTO[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.33 CSIHnMCTL0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 10</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>9 to 8</td><td>CSIHnMMS [1:0]</td><td>Selects the memory mode.
<table>
<tr><th>CSIHnMMS1</th><th>CSIHnMMS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>FIFO mode</td></tr>
<tr><td>0</td><td>1</td><td>Dual buffer mode</td></tr>
<tr><td>1</td><td>0</td><td>Transmit-only buffer mode</td></tr>
<tr><td>1</td><td>1</td><td>Prohibited</td></tr>
</table>
After a change of the memory mode, the respective buffer pointers must be cleared by setting the CSIHnSTCR0.CSIHnPCT bit to 1.
In direct access mode, the setting of these bits is ignored.</td></tr>
<tr><td>7 to 5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>4 to 0</td><td>CSIHnTO[4:0]</td><td>Selects the time-out setting in FIFO mode.
<table>
<tr><th>CSIHnTO[4:0]</th><th>Description</th></tr>
<tr><td>00000_B</td><td>No time-out is detected</td></tr>
<tr><td>00001_B</td><td>00001_B Time-out is (1 × 8 × BRG output clocks)</td></tr>
<tr><td>00010_B</td><td>00010_B Time-out is (2 × 8 × BRG output clocks)</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>11111_B</td><td>Time-out is (31 × 8 × BRG output clocks)</td></tr>
</table>
CAUTION: Changing the time-out setting is only permitted when CSIHnCTL0.CSIHnPWR = 0.
Set the CSIHnTO[4:0] bits to 00000_B in direct access mode, dual buffer mode, or transmit-only buffer mode (except FIFO mode).
For details about time-out detection, see also **Section 20.5.12.3, Time-Out Error**.</td></tr>
</table>

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.8 CSIHnMCTL1 — CSIHn Memory Control Register 1

This register selects the conditions to generate the interrupt requests INTCSIHTIC and INTCSIHTIR in FIFO mode.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnFES[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnFFS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.34 CSIHnMCTL1 Register Contents

Bit Position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CSIHnFES[6:0]	Selects the conditions to generate the INTCSIHTIC interrupt (transmit data empty) in FIFO mode. When the number of unsent data to be transmitted in FIFO (checked by the CSIHnSTR0.CSIHnSPF[7:0] bit) and CSIHnMCTL1.CSIHnFES[6:0] match, the FIFO empty flag (CSIHnSTR0.CSIHnEMF bit) is set to 1, and the INTCSIHTIC interrupt request is generated.
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CSIHnFFS[6:0]	Selects the conditions to generate the INTCSIHTIR interrupt (receive data full) in FIFO mode. When the number of received data in FIFO (checked by the CSIHnSTR0.CSIHnSRP[7:0] bit) and (128 - CSIHnMCTL1.CSIHnFFS[6:0]) match, the INTCSIHTIR interrupt request is generated.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.9 CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory in dual buffer or transmit-only buffer mode and triggers to start communication.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHn BTST	—	—	—	—	—	—	—	CSIHnND[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnSOP[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.35 CSIHnMCTL2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31</td><td>CSIHnBTST</td><td>Provides a start trigger for buffer transfer.
0: No operation.
1: Issues the start transfer command.
The read value is always 0.

CAUTION: This bit can only be used in dual buffer mode and transmit-only buffer mode.</td></tr>
<tr><td>30 to 24</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>23 to 16</td><td>CSIHnND[7:0]</td><td>Specifies the number of data for each memory mode.
The read value indicates the number of remaining communication data.
<table>
<tr><th>CSIHnND[7:0]</th><th>Dual Buffer Mode</th><th>Transmit-Only Buffer Mode</th><th>FIFO Mode</th><th>Direct Access Mode</th></tr>
<tr><td>00_H</td><td>Send 0 data</td><td>Send 0 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>01_H</td><td>Send 1 data</td><td>Send 1 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>No influence</td><td>No influence</td></tr>
<tr><td>$3F_H$</td><td>Send 63 data</td><td>Send 63 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>40_H</td><td>Send 64 data</td><td>Send 64 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>No influence</td><td>No influence</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>Send 127 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>80_H</td><td>Prohibited</td><td>Send 128 data</td><td>No influence</td><td>No influence</td></tr>
<tr><td>Other than the above</td><td colspan="4">Setting is prohibited.</td></tr>
</table>
The values are automatically decremented after data transfer (not decremented in direct access mode).</td></tr>
<tr><td>15 to 7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 20.35 CSIHnMCTL2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 0</td><td>CSIHnSOP[6:0]</td><td>Selects the pointer of the data to be sent.
If communication is forced to stop by setting CSIHnCTL0.CSIHnPWR to 0 or CSIHnSTCR0.CSIHnPCT to 1, these bits are cleared by hardware.
In FIFO mode, these bits indicate the send address.
<table>
<tr><th>CSIHnSOP[6:0]</th><th>Dual Buffer Mode</th><th>Transmit-Only Buffer Mode</th><th>FIFO Mode</th><th>Direct Access Mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>0000_H</td><td>No influence</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>0004_H</td><td>No influence</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No influence</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No influence</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>0100_H</td><td>No influence</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No influence</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>$01FC_H$</td><td>No influence</td></tr>
</table>
CAUTION: In direct access mode, these bits are not incremented.</td></tr>
</table>

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.10 CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the dual buffer or transmit-only buffer.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1018$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnRRA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnTRWA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.36 CSIHnMRWP0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 23</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>22 to 16</td><td>CSIHnRRA[6:0]</td><td>Selects the read pointer of the receive buffer.
<table>
<tr><th>CSIHnRRA[6:0]</th><th>Dual Buffer Mode</th><th>Transmit-Only Buffer Mode</th><th>FIFO Mode</th><th>Direct Access Mode</th></tr>
<tr><td>00$_H$</td><td>0000$_H$</td><td>No influence</td><td>0000$_H$</td><td>No influence</td></tr>
<tr><td>01$_H$</td><td>0004$_H$</td><td>No influence</td><td>0004$_H$</td><td>No influence</td></tr>
<tr><td>...</td><td>...</td><td>No influence</td><td>...</td><td>No influence</td></tr>
<tr><td>3F$_H$</td><td>00FC$_H$</td><td>No influence</td><td>00FC$_H$</td><td>No influence</td></tr>
<tr><td>40$_H$</td><td>Prohibited</td><td>No influence</td><td>0100$_H$</td><td>No influence</td></tr>
<tr><td>...</td><td>Prohibited</td><td>No influence</td><td>...</td><td>No influence</td></tr>
<tr><td>7F$_H$</td><td>Prohibited</td><td>No influence</td><td>01FC$_H$</td><td>No influence</td></tr>
</table>
These bits are automatically incremented when received data is read.
If an overrun error is generated while reading the CSIHnRX0W or CSIHnRX0H register, the read pointer is not incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode and transmit-only buffer mode, these bits are not incremented.
To perform write access in transmit-only buffer mode, set 00$_H$ to these bits.
In FIFO mode, these bits indicate the read address of the received data.</td></tr>
<tr><td>15 to 7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 20.36 CSIHnMRWP0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 0</td><td>CSIHnTRWA
[6:0]</td><td>Selects the read/write pointer of the transmit buffer.
<table>
<tr><th>CSIHnTRWA[6:0]</th><th>Dual Buffer Mode</th><th>Transmit-Only Buffer Mode</th><th>FIFO Mode</th><th>Direct Access Mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>0000_H</td><td>No influence</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>0004_H</td><td>No influence</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No influence</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No influence</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>0100_H</td><td>No influence</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No influence</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>$01FC_H$</td><td>No influence</td></tr>
</table>
These bits are automatically incremented when the transmission data is written or read.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode, these bits are not incremented.
In FIFO mode, these bits indicate the read/write address of transmission data.</td></tr>
</table>

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.11 CSIHnCFGx — CSIHn Configuration Register x (x = 0 to 7)

These eight registers configure the prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup for each chip select signal, CSIHTCSSx.

Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

In slave mode, set all bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers to 0.

Access: This register can be read or written in 32-bit units.

Address: CSIHnCFG0: <CSIHn_base> + 1044$_{H}$
CSIHnCFG1: <CSIHn_base> + 1048$_{H}$
CSIHnCFG2: <CSIHn_base> + 104C$_{H}$
CSIHnCFG3: <CSIHn_base> + 1050$_{H}$
CSIHnCFG4: <CSIHn_base> + 1054$_{H}$
CSIHnCFG5: <CSIHn_base> + 1058$_{H}$
CSIHnCFG6: <CSIHn_base> + 105C$_{H}$
CSIHnCFG7: <CSIHn_base> + 1060$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnBRSSx [1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]				—	—	—	—	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

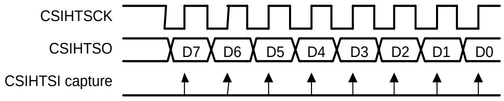
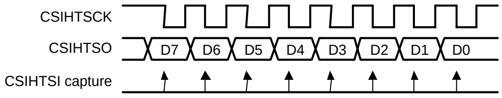
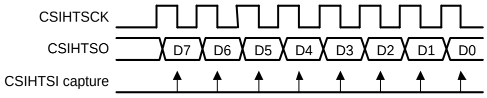
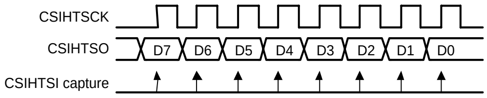
Table 20.37 CSIHnCFGx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31, 30</td><td>CSIHnBRSSx[1:0]</td><td>These bits select the baud rate setting register (CSIHnBRSy).
<table>
<tr><th>CSIHnBRSSx1</th><th>CSIHnBRSSx0</th><th>Baud Rate Setting Register Selection</th></tr>
<tr><td>0</td><td>0</td><td>The transfer clock frequency is set according to the CSIHnBRS0 setting.</td></tr>
<tr><td>0</td><td>1</td><td>The transfer clock frequency is set according to the CSIHnBRS1 setting.</td></tr>
<tr><td>1</td><td>0</td><td>The transfer clock frequency is set according to the CSIHnBRS2 setting.</td></tr>
<tr><td>1</td><td>1</td><td>The transfer clock frequency is set according to the CSIHnBRS3 setting.</td></tr>
</table>
The maximum value for setting the transfer clock frequency, in accordance with the CSIHnCTL2.CSIHnPRS[2:0] setting, must be as follows:
Master mode: PCLK/4
Slave mode: PCLK/6</td></tr>
<tr><td>29, 28</td><td>CSIHnPSx[1:0]</td><td>Selects the parity for transmitting or receiving chip select signal x.
<table>
<tr><th>CSIHnPSx1</th><th>CSIHnPSx0</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Does not transmit any parity bit.</td><td>Does not wait for reception of the parity bit.</td></tr>
<tr><td>0</td><td>1</td><td>Adds a parity bit fixed to 0.</td><td>Waits for reception of the parity bit but does not evaluate it.</td></tr>
<tr><td>1</td><td>0</td><td>Adds the odd parity bit.</td><td>Waits for the odd parity bit.</td></tr>
<tr><td>1</td><td>1</td><td>Adds the even parity bit.</td><td>Waits for the even parity bit.</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIHnDLSx[3:0]</td><td>Selects the data length for chip select signal x.
<table>
<tr><th>CSIHnDLSx[3:0]</th><th>Data Length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
CAUTION: When CSIHnTX0W.CSIHnEDL = 1, the setting of this bit has no effect.
When CSIHnTX0W.CSIHnEDL = 0 (the data length is 16 bits), the setting of this bit is valid. Setting “1 bit” is only enabled if the previous transmit data was 16 bits with CSIHnEDL = 1.</td></tr>
<tr><td>23 to 20</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>19</td><td>CSIHnRCBx</td><td>Selects the recessive configuration for broadcasting for chip select signal x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see **Section 20.5.3.1, Configuration Registers**</td></tr>
<tr><td>18</td><td>CSIHnDIRx</td><td>Selects the serial data direction of chip select signal x.
0: Data is transmitted/received with MSB first.
1: Data is transmitted/received with LSB first.
For details, see **Section 20.5.9, Serial Data Direction Selection**.</td></tr>
</table>

Table 20.37 CSIHnCFGx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17</td><td>CSIHnCKPx</td><td>CSIHnCKPx: Clock phase selection bit</td></tr>
<tr><td>16</td><td>CSIHnDAPx</td><td>CSIHnDAPx: Data phase selection bit

- CSIHnCTL1.CSIHnCKR = 0

<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock Phase and Data Phase Selection</th></tr>
<tr><td>0</td><td>0</td><td>
</td></tr>
<tr><td>0</td><td>1</td><td>
</td></tr>
<tr><td>1</td><td>0</td><td>
</td></tr>
<tr><td>1</td><td>1</td><td>
</td></tr>
</table>

- CSIHnCTL1.CSIHnCKR = 1

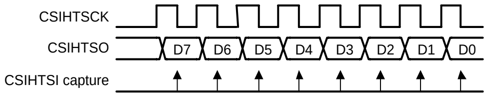
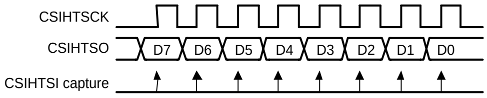
<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock Phase and Data Phase Selection</th></tr>
<tr><td>0</td><td>0</td><td>
</td></tr>
<tr><td>0</td><td>1</td><td>
</td></tr>
<tr><td>1</td><td>—</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the idle enforcement configuration for chip select signal x
0: If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are the same, an idle state is not inserted between two transfers.
1: Regardless of the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers, an idle state are inserted between two transfers.
This bit is only available in master mode.
For details about the enforced idle state, see **Section 20.5.15, Enforced Chip Select Idle Setting**.</td></tr>
</table>

Table 20.37 CSIHnCFGx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>14 to 12</td><td>CSIHnIDx[2:0]</td><td>Selects the idle time for chip select signal x.
<table>
<tr><th>CSIHnIDx[2:0]</th><th>Idle time</th></tr>
<tr><td>000_B</td><td>0.5 transmission clock cycle</td></tr>
<tr><td>001_B</td><td>1.0 transmission clock cycle</td></tr>
<tr><td>010_B</td><td>1.5 transmission clock cycles</td></tr>
<tr><td>011_B</td><td>2.5 transmission clock cycles</td></tr>
<tr><td>100_B</td><td>3.5 transmission clock cycles</td></tr>
<tr><td>101_B</td><td>4.5 transmission clock cycles</td></tr>
<tr><td>110_B</td><td>6.5 transmission clock cycles</td></tr>
<tr><td>111_B</td><td>8.5 transmission clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>11 to 8</td><td>CSIHnHDx[3:0]</td><td>Specifies the hold time for chip select signal x in transmission clock cycles.
<table>
<tr><th>CSIHn HDx[3:0]</th><th>Hold time when CSIHnCTL1.CSIHnSIT is 0</th><th>Hold time when CSIHnCTL1.CSIHnSIT is 1</th></tr>
<tr><td>0000_B</td><td>0.5 transmission clock cycle</td><td>1.0 transmission clock cycle</td></tr>
<tr><td>0001_B</td><td>1.0 transmission clock cycle</td><td>1.5 transmission clock cycles</td></tr>
<tr><td>0010_B</td><td>1.5 transmission clock cycles</td><td>2.0 transmission clock cycles</td></tr>
<tr><td>0011_B</td><td>2.5 transmission clock cycles</td><td>3.0 transmission clock cycles</td></tr>
<tr><td>0100_B</td><td>3.5 transmission clock cycles</td><td>4.0 transmission clock cycles</td></tr>
<tr><td>0101_B</td><td>4.5 transmission clock cycles</td><td>5.0 transmission clock cycles</td></tr>
<tr><td>0110_B</td><td>6.5 transmission clock cycles</td><td>7.0 transmission clock cycles</td></tr>
<tr><td>0111_B</td><td>8.5 transmission clock cycles</td><td>9.0 transmission clock cycles</td></tr>
<tr><td>1000_B</td><td>9.5 transmission clock cycles</td><td>10.0 transmission clock cycles</td></tr>
<tr><td>1001_B</td><td>10.5 transmission clock cycles</td><td>11.0 transmission clock cycles</td></tr>
<tr><td>1010_B</td><td>11.5 transmission clock cycles</td><td>12.0 transmission clock cycles</td></tr>
<tr><td>1011_B</td><td>12.5 transmission clock cycles</td><td>13.0 transmission clock cycles</td></tr>
<tr><td>1100_B</td><td>14.5 transmission clock cycles</td><td>15.0 transmission clock cycles</td></tr>
<tr><td>1101_B</td><td>16.5 transmission clock cycles</td><td>17.0 transmission clock cycles</td></tr>
<tr><td>1110_B</td><td>18.5 transmission clock cycles</td><td>19.0 transmission clock cycles</td></tr>
<tr><td>1111_B</td><td>20.5 transmission clock cycles</td><td>21.0 transmission clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

Table 20.37 CSIHnCFGx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>CSIHnINx[3:0]</td><td>Specifies the inter-data time for chip select signal x in transmission clock cycles.
<table>
<tr><th>CSIHn INx[3:0]</th><th>Hold time when CSIHnCTL1.CSIHnSIT is 0</th><th>Hold time when CSIHnCTL1.CSIHnSIT is 1</th></tr>
<tr><td>0000_B</td><td>0.0 transmission clock cycle</td><td>0.5 transmission clock cycle</td></tr>
<tr><td>0001_B</td><td>0.5 transmission clock cycle</td><td>1.0 transmission clock cycle</td></tr>
<tr><td>0010_B</td><td>1.0 transmission clock cycle</td><td>1.5 transmission clock cycles</td></tr>
<tr><td>0011_B</td><td>2.0 transmission clock cycles</td><td>2.5 transmission clock cycles</td></tr>
<tr><td>0100_B</td><td>3.0 transmission clock cycles</td><td>3.5 transmission clock cycles</td></tr>
<tr><td>0101_B</td><td>4.0 transmission clock cycles</td><td>4.5 transmission clock cycles</td></tr>
<tr><td>0110_B</td><td>6.0 transmission clock cycles</td><td>6.5 transmission clock cycles</td></tr>
<tr><td>0111_B</td><td>8.0 transmission clock cycles</td><td>8.5 transmission clock cycles</td></tr>
<tr><td>1000_B</td><td>9.0 transmission clock cycles</td><td>9.5 transmission clock cycles</td></tr>
<tr><td>1001_B</td><td>10.0 transmission clock cycles</td><td>10.5 transmission clock cycles</td></tr>
<tr><td>1010_B</td><td>11.0 transmission clock cycles</td><td>11.5 transmission clock cycles</td></tr>
<tr><td>1011_B</td><td>12.0 transmission clock cycles</td><td>12.5 transmission clock cycles</td></tr>
<tr><td>1100_B</td><td>14.0 transmission clock cycles</td><td>14.5 transmission clock cycles</td></tr>
<tr><td>1101_B</td><td>16.0 transmission clock cycles</td><td>16.5 transmission clock cycles</td></tr>
<tr><td>1110_B</td><td>18.0 transmission clock cycles</td><td>18.5 transmission clock cycles</td></tr>
<tr><td>1111_B</td><td>20.0 transmission clock cycles</td><td>20.5 transmission clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>3 to 0</td><td>CSIHnSPx[3:0]</td><td>Specifies the setup time for chip select signal x in transmission clock cycles.
<table>
<tr><th>CSIHnSPx[3:0]</th><th>Setup Time</th></tr>
<tr><td>0000_B</td><td>0.5 transmission clock cycle</td></tr>
<tr><td>0001_B</td><td>1.0 transmission clock cycle</td></tr>
<tr><td>0010_B</td><td>1.5 transmission clock cycles</td></tr>
<tr><td>0011_B</td><td>2.5 transmission clock cycles</td></tr>
<tr><td>0100_B</td><td>3.5 transmission clock cycles</td></tr>
<tr><td>0101_B</td><td>4.5 transmission clock cycles</td></tr>
<tr><td>0110_B</td><td>6.5 transmission clock cycles</td></tr>
<tr><td>0111_B</td><td>8.5 transmission clock cycles</td></tr>
<tr><td>1000_B</td><td>9.5 transmission clock cycles</td></tr>
<tr><td>1001_B</td><td>10.5 transmission clock cycles</td></tr>
<tr><td>1010_B</td><td>11.5 transmission clock cycles</td></tr>
<tr><td>1011_B</td><td>12.5 transmission clock cycles</td></tr>
<tr><td>1100_B</td><td>14.5 transmission clock cycles</td></tr>
<tr><td>1101_B</td><td>16.5 transmission clock cycles</td></tr>
<tr><td>1110_B</td><td>18.5 transmission clock cycles</td></tr>
<tr><td>1111_B</td><td>20.5 transmission clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.12 CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1008_H

Value after reset: $XXXX\ XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHn CIRE	CSIHn EOJ	CSIHn EDL	—	—	—	—	—	CSIHnC S7	CSIHnC S6	CSIHnC S5	CSIHnC S4	CSIHnC S3	CSIHnC S2	CSIHnC S1	CSIHnC S0
Value after reset	—	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.38 CSIHnTX0W Register Contents

Bit Position	Bit Name	Function
31	CSIHnCIRE	Enables the communication interrupt request INTCSIHTIC in dual buffer or transmit-only buffer mode, or the job completion interrupt request INTCSIHTIJC in FIFO mode. 0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHTIC or INTCSIHTIJC after transmission. For details, see **Section 20.4.3, INTCSIHTIC (Communication Status Interrupt)** and **Section 20.4.6, INTCSIHTIJC (Job Completion Interrupt)**. **CAUTION:** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
30	CSIHnEOJ	Specifies the end of a job. 0: Indicates that it is not end-of-job data. The job continues. 1: Indicates end-of-job data. **CAUTION:** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the subsequent data must have the same CS selection. If CS is modified for the subsequent data, the correct operation is not assured. **CAUTION:** This bit is only available if CSIHnCTL1.CSIHnEDLE = 1.
28 to 24	Reserved	When read, an undefined value is returned. When writing to these bits, write 0.

Table 20.38 CSIHnTX0W Register Contents

Bit Position	Bit Name	Function
23 to 16	CSIHnCS[7:0]	Activates one or more chip select signals. 0: Activates chip select signals x for the associated transmission. 1: Deactivates chip select signals x for the associated transmission. Setting CSIHnTX0W.CSIHnCS[7:0] = FF$_H$ is prohibited. **CAUTION:** If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value. In slave mode, set the CSIHnCS[7:0] bit to FE$_H$.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.13 CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

The settings specified by the upper 16 bits of CSIHnTX0W are applied to the transmission. Set transmit data to CSIHnTX0W before using this register because the value of CSIHnTX0W is undefined after the reset.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + 100C$_H$

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.39 CSIHnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.14 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIHn_base> + 1010_H

Value after reset: $0XXX\ XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CSIHn RPE	CSIHn TDCE	CSIHn CS7	CSIHn CS6	CSIHn CS5	CSIHn CS4	CSIHn CS3	CSIHn CS2	CSIHn CS1	CSIHn CS0
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.40 CSIHnRX0W Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data.
24	CSIHnTDCE	Indicates whether a transmission data consistency check error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
23 to 16	CSIHnCSx (x = 7 to 0)	Indicates which chip select signal was activated. 0: Chip select x was activated for the associated reception. 1: Chip select x was deactivated for the associated reception.
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

This register stores the received data when an INTCSIHTIR interrupt is generated. Read the received data stored in this register before generation of an INTCSIHTIR interrupt. Otherwise the data is rewritten with the next received data.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.15 CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <CSIHn_base> + 1014_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.41 CSIHnRX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

This register stores the received data when an INTCSIHTIR interrupt is generated. Read the received data stored in this register before generation of an INTCSIHTIR interrupt. Otherwise the data is rewritten with the next received data.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.16 CSIHnEMU — CSIHn Emulation Register

This register controls operation of SVSTOP.

Access: This register can be read or written in 8-bit or 1-bit units.
Perform write operation when (EPC.SVSTOP = 0).

Address: <CSIHn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 20.42 CSIHnEMU Register Contents

Bit Position	Bit Name	Function
7	CSIHnSVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. ● When the EPC.SVSTOP bit is set to 0 Continues transmit/receive operation regardless of the setting of this bit. ● When the EPC.SVSTOP bit is set to 1 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.17 CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3)

This register sets the transfer clock frequency for each chip select signal.

With CSIHnCFG0 to 7.CSIHnBRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency setting, see **Section 20.5.5, Transmission Clock Selection**.

Access:	This register can be read or written in 16-bit units.
Address:	CSIHnBRS0: <CSIHn_base> + 1068_H CSIHnBRS1: <CSIHn_base> + $106C_H$ CSIHnBRS2: <CSIHn_base> + 1070_H CSIHnBRS3: <CSIHn_base> + 1074_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CSIHnBRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.43 CSIHnBRSy Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CSIHnBRS[11:0]	0: BRG stopped 1: PCLK / (2^{α} × 1 × 2) 2: PCLK / (2^{α} × 2 × 2) 3: PCLK / (2^{α} × 3 × 2) 4: PCLK / (2^{α} × 4 × 2) . . . 4095: PCLK / (2^{α} × 4095 × 2) α is the value of CSIHnCTL2.CSIHnPRS[2:0].

CAUTION

When setting this register, see **Table 20.44, Notes on Setting Registers**.

20.3.18 List of Cautions

Table 20.44 Notes on Setting Registers

Register Name	Bit Name	Cautions
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIHnCTL0	CSIHnTXE CSIHnRXE	Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIHnCTL0	CSIHnJOBE	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode.
CSIHnCTL0	CSIHnMBS	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only permitted while CSIHnSTR0.CSIHnTSF = 0. Do not change the mode between FIFO mode and direct access mode while CSIHnCTL0.CSIHnPWR = 1. When the CPU-controlled high-priority communication is enabled, the operation is the same as that in direct access mode regardless of the CSIHnMBS bit setting.
CSIHnCTL1	CSIHnCKR	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. When CS is not used, use this bit instead of CSIHnCFGx.CSIHnCKPx and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode.
CSIHnCTL1	CSIHnSLIT CSIHnCSL[7:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.
CSIHnCTL1	CSIHnPHE CSIHnJE CSIHnLBM	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in slave mode.
CSIHnCTL1	CSIHnSSE	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIHnCTL1	CSIHnSIT	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIHnCTL2	CSIHnPRS[2:0]	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the maximum transfer clock frequency is as follows. ● Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4) ● Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
CSIHnSTR0	CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnHPST CSIHnFLF CSIHnEMF CSIHnTSF	Writing to these bits is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnTMOE CSIHnOFE CSIHnDCE CSIHnPE CSIHnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.
CSIHnSTCR0	CSIHnPCT	If this bit is set to 1 during communication, ongoing communication is aborted.
CSIHnMCTL0	CSIHnMMS[1:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.

Table 20.44 Notes on Setting Registers

Register Name	Bit Name	Cautions
CSIHnMCTL0	CSIHnTO[4:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 in master mode. Set these bits to 0 in direct access, dual buffer, and transmit-only buffer mode.
CSIHnMCTL1	CSIHnFES[6:0] CSIHnFFS[6:0]	Writing to these bits while communication is ongoing is permitted.
CSIHnMCTL2	CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0]	Writing to these bits is prohibited when CSIHnCTL0.CSIHnPWR = 0. Writing to these bits is prohibited when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0. Writing to these bits is prohibited when CSIHnSTR0.CSIHnTSF = 1. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnMRWP0	CSIHnRRA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode. When writing is required, set "00_H" to these bits in transmit-only buffer mode.
CSIHnMRWP0	CSIHnTRWA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnCFGx x = 0 to 7	CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDx[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. These bits must be set to 0 in slave mode.
CSIHnCFGx x = 0 to 7	CSIHnPSx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. In slave mode, the CSIHnCFG0 setting is used for the configuration. Therefore, all the bits in CSIHnCFG1 to CSIHnCFG7 must be set to 0.
CSIHnCFGx x = 0 to 7	CSIHnCKPx	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit must be set to 0 in slave mode as CSIHnCTL1.CSIHnCKR must be used in slave mode. If CS is not used, the CSIHnCTL1.CSIHnCKR bit instead of this bit, and clear this bit to 0.
CSIHnTX0W	CSIHnEOJ CSIHnCIRE	These bits are only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, the values of these bits are ignored even if 1 is read. Set these bits to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	This bit is only valid when CSIHnCTL1.CSIHnEDLE = 1. While CSIHnCTL1.CSIHnEDLE = 0, the value of this bit is ignored even if 1 is read.
CSIHnTX0W	CSIHnCS[7:0]	In master mode, setting this bit to "FF_H" is prohibited. In slave mode, set this bit to "FE_H".
CSIHnTX0W CSIHnTX0H		Reading these bits while communication is ongoing is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, reading and writing to these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is permitted in the mode (Transmit-only buffer, Dual buffer and Direct access modes) except FIFO mode. While CSIHnCTL0.CSIHnPWR = 1, reading these bits is permitted.

Table 20.44 Notes on Setting Registers

Register Name	Bit Name	Cautions
CSIHnRX0H		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 1, reading these bits is permitted in the FIFO mode. In spite of CSIHnCTL0.CSIHnPWR value, reading is permitted of these bits in the mode (Transmit-only buffer, Dual buffer and Direct access modes) except FIFO mode.
CSIHnEMU	CSIHnSVSDIS	Modification of this bit is only permitted while SVSTOP = 0.
CSIHnBRSy y = 0 to 3		Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.

20.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHTIC (communication status interrupt)
- INTCSIHTIR (reception status interrupt)
- INTCSIHTIRE (communication error interrupt)
- INTCSIHTIJC (job completion interrupt)

20.4.1 Overview

The communication error interrupt INTCSIHTIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHTIJC – also the operating mode.

The job completion interrupt INTCSIHTIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

The following table gives an overview.

Table 20.45 Interrupt Generation

Memory Mode	Interrupt	Interrupt Source	
		Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	INTCSIHTIC	Tx data empty*[1]	Tx data empty*[1] except job abort*[4]
	INTCSIHTIR	Rx data full*[2] *and* CSIHnCTL0.CSIHnRXE = 1	Rx data full*[2] *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*[3]	Not applicable	CSIHnTX0W.CSIHnCIRE = 1 (except Tx data empty), or job abort*[4]
Transmit-only buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 *and* (CSIHnCTL0.CSIHnJOBE = 0 *or* CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*[3]	Not applicable	Job abort*[4]
Dual buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 *and* (CSIHnCTL0.CSIHnJOBE = 0 *or* CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	End of communication *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*[3]	Not applicable	Job abort*[4]
Direct access	INTCSIHTIC	One data transfer	One data transfer except the state of job abort*[4]
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*[3]	Not applicable	Job abort*[4]

Note 1. “Tx data empty” refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFES[6:0].

Note 2. “Rx data full” refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFFS[6:0].

Note 3. INTCSIHTIJC is not available in slave mode.

Note 4. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

20.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half a cycle of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting CSIHnCTL1.CSIHnSIT = 1. (The setting of the CSIHnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (clock phase and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

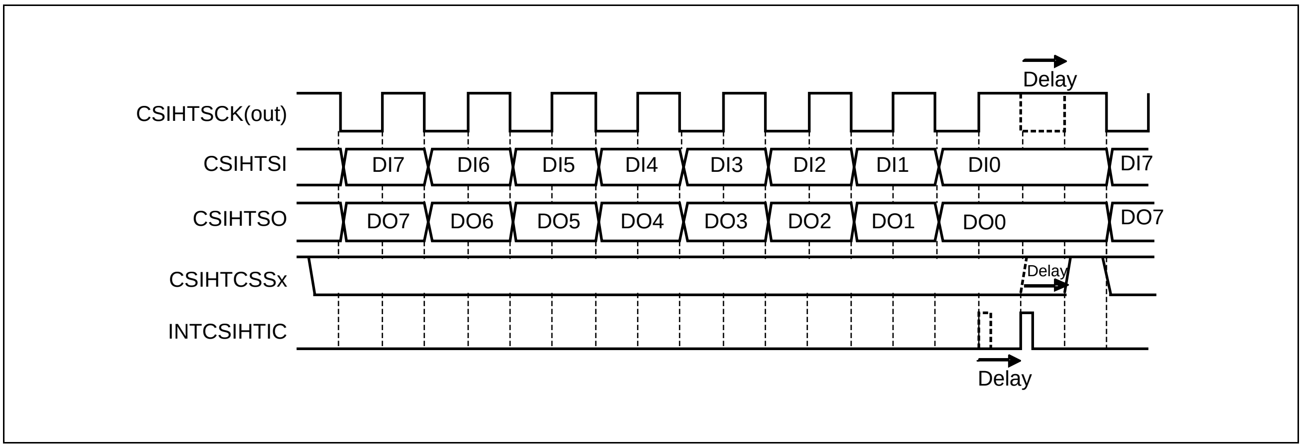


Figure 20.2 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds a half cycle delay to the transmission clock. This also delays the end of the current chip select signal (CSIHTCSSx).

20.4.3 INTCSIHTIC (Communication Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 20.46 INTCSIHTIC Interrupt Generation

Memory Mode	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt is generated just before transmission data in the FIFO runs out, notifying the application that new data should be added. INTCSIHTIC is generated if the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMTCL1.CSIHnFES[6:0].	Similar to “when JE is 0”, an interrupt is generated when the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMCTL1.CSIHnFES[6:0]. However, it is not generated if a job is aborted.
Transmit-only buffer, dual buffer	An interrupt is generated at the end of communication. (specified by the CSIHnMTLC2.CSIHnND[7:0] bit)	Generated when data is transmitted while CSIHnTX0W.CSIHnCIRE = 1. Note that if data and job abort*[1] are transmitted while CSIHnTX0W.CSIHnCIRE = 1, the INTCSIHTIJC interrupt is generated instead of INTCSIHTIC.
Direct access	Generated after every data transfer.	Generated after every data transfer, except when the communication was aborted.

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

20.4.3.1 INTCSIHTIC in Direct Access Mode

The examples below show the INTCSIHTIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase
  (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

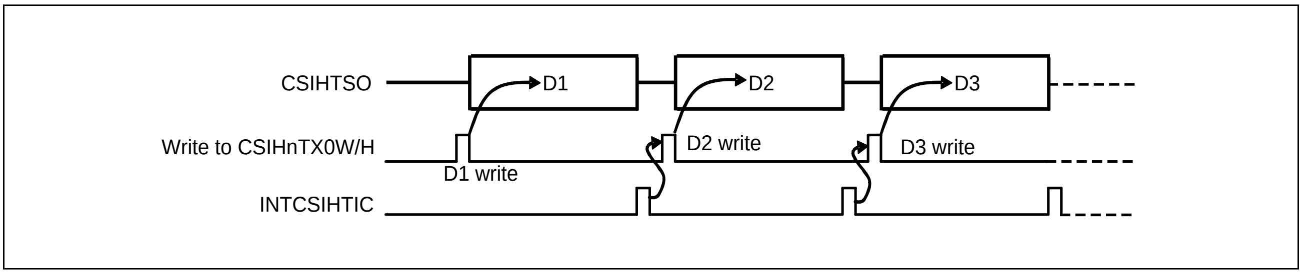


Figure 20.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTXOW.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHTIC is replaced by the job completion interrupt INTCSIHTIJC.

INTCSIHTIC can also be set up to occur as soon as the CSIHnTX0W/H register becomes empty and available for storing the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

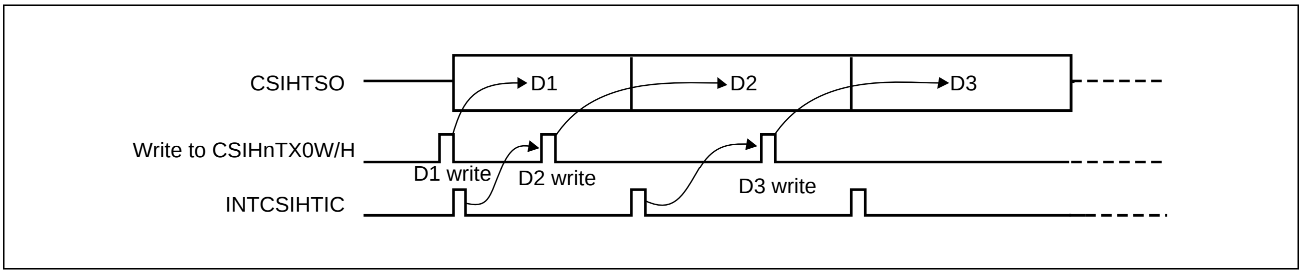


Figure 20.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

During high priority communication in transmit-only buffer mode, the operation is in the same as that in direct access mode.

20.4.3.2 INTCSIHTIC in FIFO Mode

The example below shows the INTCSIHTIC behavior in FIFO mode.

The example assumes:

- Master mode
- FIFO mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

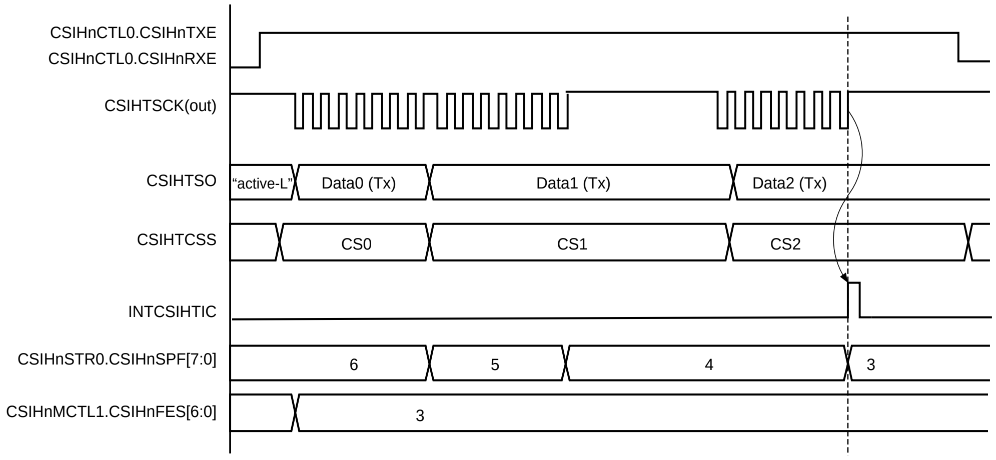


Figure 20.5 Generation of INTCSIHTIC in FIFO Memory Mode

The condition for "FIFO empty" is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, the number of unsent data in FIFO is set to 3.

CSIHnSTR0.CSIHnSPF[7:0] indicates the number of unsent data. When both match, the interrupt INTCSIHTIC occurs.

20.4.3.3 INTCSIHTIC in Job Mode

The example below shows the INTCSIHTIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

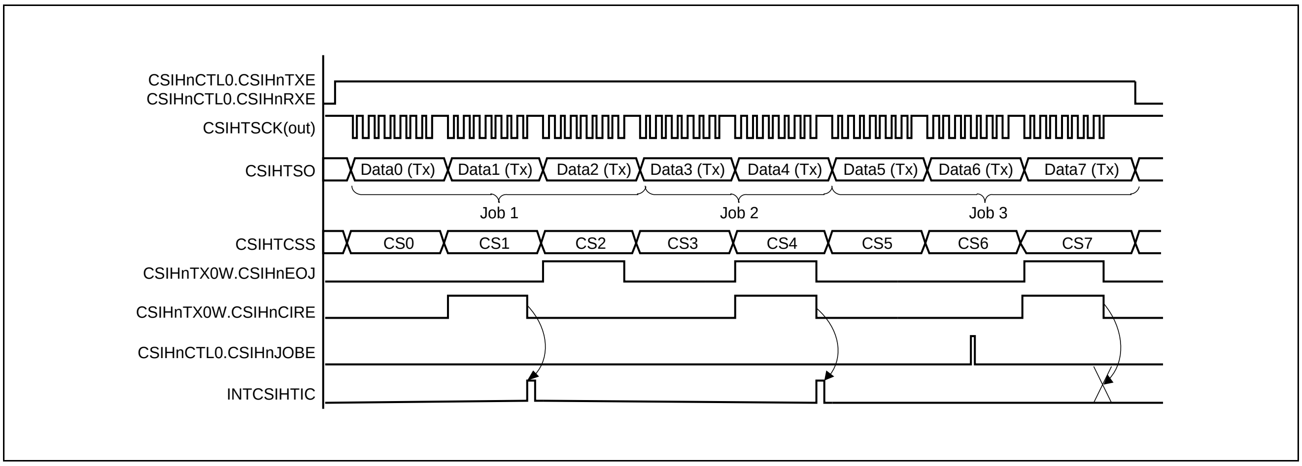


Figure 20.6 Generation of INTCSIHTIC in Job Mode

The rules for generating INTCSIHTIC in job mode are shown in the following table.

Table 20.47 Generation of INTCSIHTIC in Job Mode

CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	INTCSIHTIC
0	0	Not generated
0	1	Generated
1	0	Not generated
1	1	CSIHnCTL0.CSIHnJOBE = 0: Generated
		CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHTIJC

20.4.4 INTCSIHTIR (Reception Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

Table 20.48 INTCSIHTIR Interrupt Generation

	Interrupt Source	
Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt occurs when CSIHnCTL0.CSIHnRXE is 1 and the FIFO buffer is almost full with received data, notifying the application that the FIFO must be emptied. INTCSIHTIR is generated, if the number of received data in the FIFO (CSIHnSTR0.CSIHnSRP[7:0]) equals (128 – CSIHnMCTL1.CSIHnFFS[6:0]).	
Dual buffer	An interrupt is generated when the communication has finished (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits) and CSIHnCTL0.CSIHnRXE = 1.	An interrupt is generated after every data transfer.
Transmit-only buffer, Direct access	An interrupt is generated after every data transfer.	

20.4.4.1 INTCSIHTIR in Direct Access Mode

The example below shows the INTCSIHTIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase
  (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

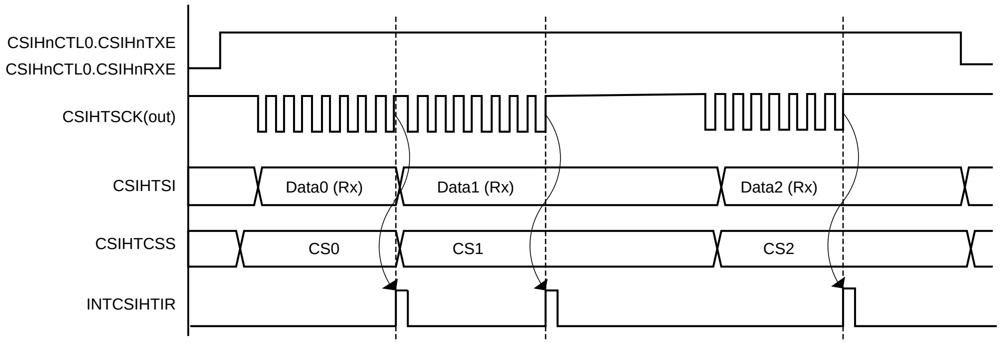


Figure 20.7 Generation of INTCSIHTIR in Direct Access Mode

20.4.4.2 INTCSIHTIR in Dual Buffer Mode

The example below shows the INTCSIHTIR behavior in dual buffer mode.

The example assumes:

- Master mode
- Dual buffer mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Default clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

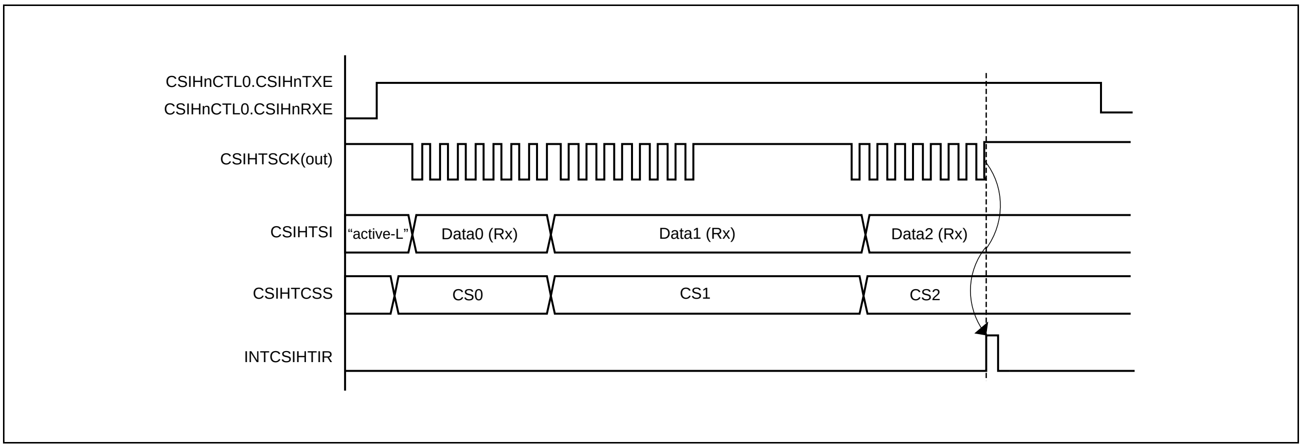


Figure 20.8 Generation of INTCSIHTIR in Dual Buffer Mode

20.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about interrupt generation timing, see **Section 20.5.12, Error Detection**.

Table 20.49 Data Error Types

Error Type	Communication Status after Error Interrupt	Note
FIFO overflow error	Communication continues even if an interrupt is generated.	The data is not written to the FIFO buffer and the data that overflowed is lost, but communications started before the error continue.
Parity error	Communication continues even if an interrupt is generated.	—
Data consistency check error	Communication continues even if an interrupt is generated.	—
Time-out error	Communication continues even if an interrupt is generated.	—
Overrun error	Condition for errors 1: In FIFO mode, when the number of received data is 0 and CPU reads the CSIHnRX0W/H register, an interrupt is generated and communication continues.	—
	Condition for errors 2: In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled): [1] In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data is retained in the CSIHnRX0W/H register, an interrupt is generated, and communication continues. [2] In FIFO mode, when reception by the FIFO buffer is completed and the buffer is in the full state, an interrupt is generated. Communication continues.	In slave mode, when CSIHnCTL1.CSIHnHSE = 1 (handshake function enabled), communication is suspended due to handshake, an overrun error is not generated.

The type of error that caused the generation of INTCSIHTIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency check error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see **Section 20.5.12, Error Detection**.

20.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs. See **Section 20.5.3.3, Job Concept**. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, INTCSIHTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the condition shown in the following table.

Table 20.50 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not applicable	Indicates that the communication stopped at the end of a job after a job abort*[1] was triggered If FIFO empty is not detected, INTCSIHTIJC is generated when CSIHnCIRE is 1.
Transmit-only buffer		Indicates that the communication stopped at the end of a job after a job abort*[1] was triggered.
Dual buffer		
Direct access		

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

20.5 Operation

20.5.1 Operating Modes (Master/Slave)

Whether CSIH operates in the master or slave mode determines the source of the serial clock.

20.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and supplied to the slave(s) by signal CSIHTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to values other than 111_B. In master mode, the BRG frequency can be specified by setting the CSIHnCTL2.CSIHnPRS[2:0] bits in combination with the CSIHnBRSy.CSIHnBRS[11:0] bits.

(1) Chip select signals

In master mode, one or more chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to select one or more slaves. Only the selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see **Section 20.5.3, Chip Selection (CS) Features**.

(2) Clock defaults

The default level of CSIHTSCK depends on the clock phase inversion function bit of the CSIHTSCK, and is high when CSIHnCTL1.CSIHnCKR = 0 and is low when CSIHnCTL1.CSIHnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

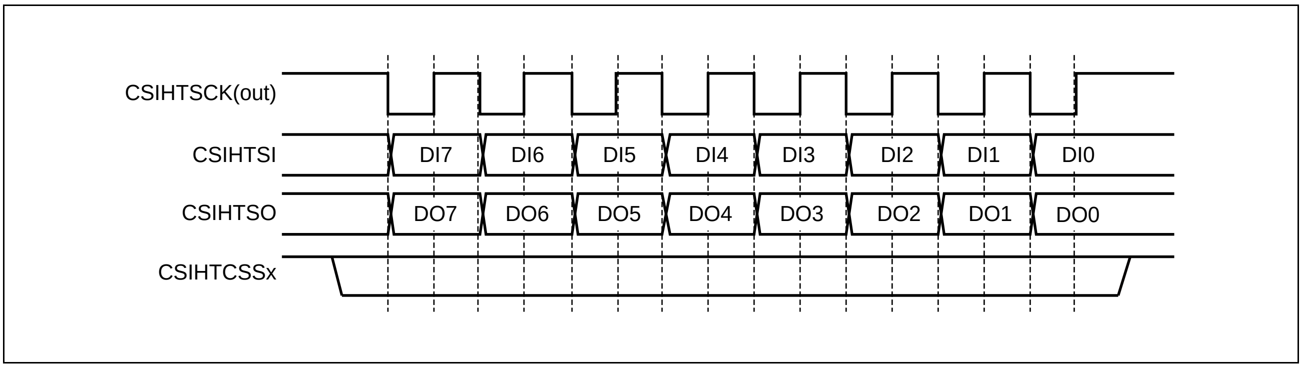


Figure 20.9 Transmission/Reception in Master Mode

20.5.1.2 Slave Mode

In slave mode, another device is the communication master and supplies the transmission clock. Normal transmit/receive operation is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bits to 111_B.

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is enabled (settings of the CSIHnCFG1 - CSIHnCFG7 registers are disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the CSIHnBRSy.CSIHnBRS[11:0] bits to 000_H. However, if you are using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000_H.

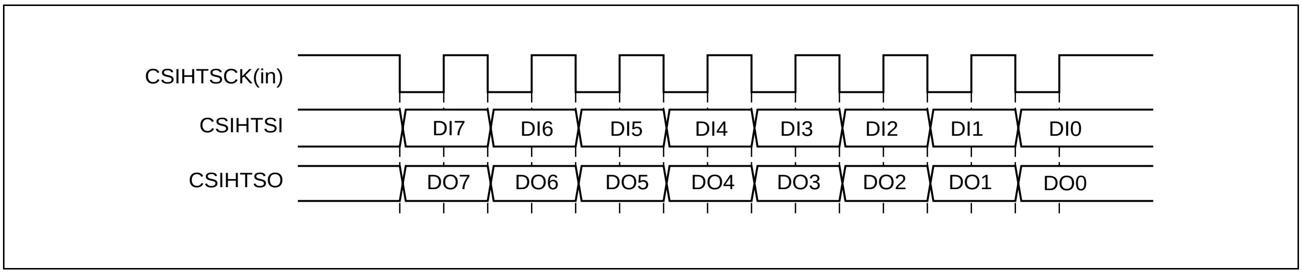


Figure 20.10 Transmission/Reception in Slave Mode

20.5.2 Master/Slave Connections

20.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

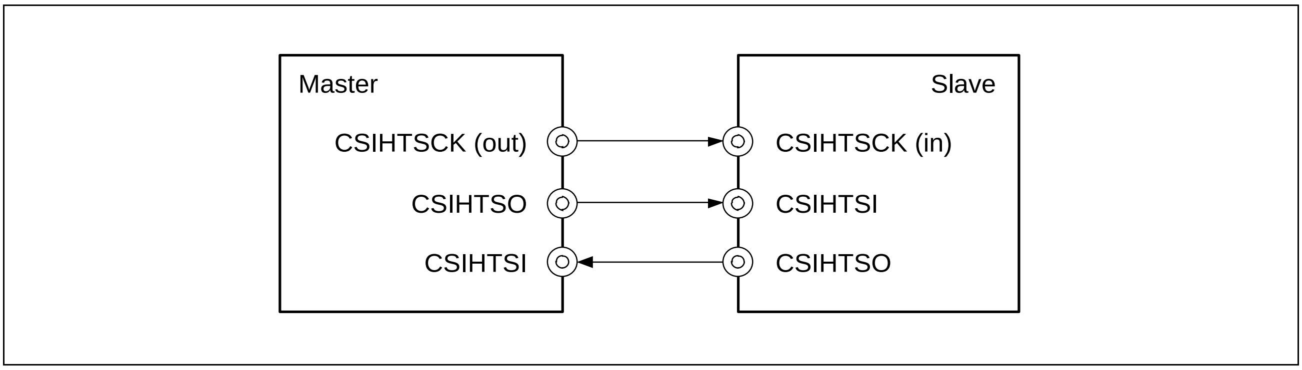


Figure 20.11 Direct Master/Slave Connection

20.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master supplies one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIHTSSI}}$ of the slave.

The $\overline{\text{CSIHTSSI}}$ signal can be enabled/disabled by using the CSIHnCTL1.CSIHnSSE bit.

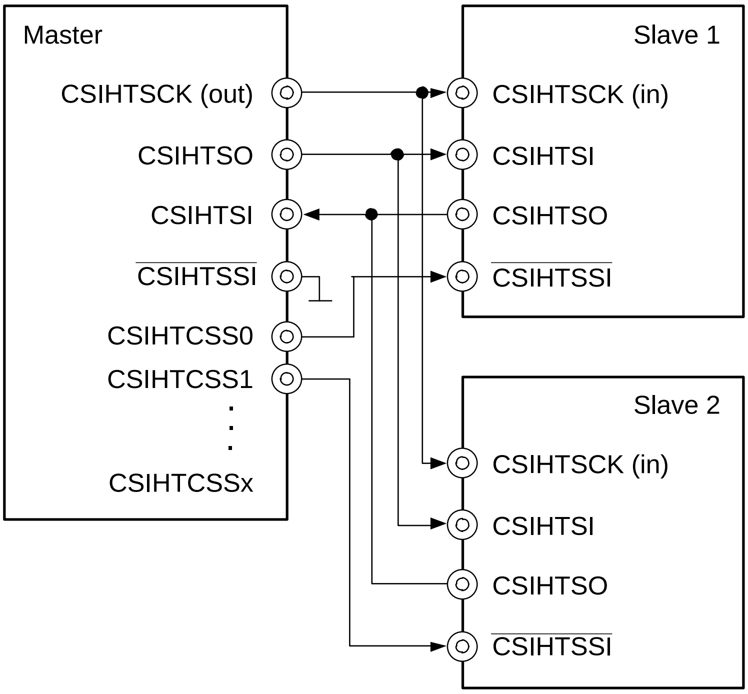


Figure 20.12 Connections between One Master and Multiple Slaves

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSIH slave when its $\overline{\text{CSIHTSSI}}$ signal is low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1), the CSIHTSO output of the slaves that are not selected is disabled and set to input mode in order to avoid interference with the output of the selected slave.

20.5.3 Chip Selection (CS) Features

The chip select signal, CSIHTCSSx can be used by the master to select one or more slaves for communication.

20.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.
  (CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first.
  (CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none.
  (CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase.
  (CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that are only available in master mode are:

- Individual selection of the baud rate generator prescaler for each chip select signal
  (CSIHnCFGx.CSIHnBRSSx[1:0])
- Chip select priority: Categorizes chip select signals into “dominant” and “recessive”. The priority applies if two or more chip selects signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration specified for dominant chip select signals is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

When specifying multiple chip select signals as dominant, be sure to configure the same settings for all dominant signals.

- Chip select timing:
  - Setup time Tsetup: The time from when the CS signal becomes active to the start of data output.
    (CSIHnCFGx.CSIHnSPx[3:0])
  - Inter-data time Tinter: The time between one data and the next data while the same CS signal is active.
    (CSIHnCFGx.CSIHnINx[3:0])
  - Hold time Thold: The time during which the CS signal remains active until CS is switched.
    (CSIHnCFGx.CSIHnHDx[3:0])
  - Idle time Tidle: Inactive time after terminating a CS signal or after every data transfer to the same CSx.
    (CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When CSIHnCFGx.CSIHnIDLx bit is set to 1, IDLE time is inserted for every transfer regardless of CS signal.

Figure 20.13, Chip Select Timings provides an example of when the default active low setting is specified for the CSIHTCSS1 and CSIHTCSS2 signals (CSIHnCTL1.CSIHnCSL1 bit = 0, CSIHnCTL1.CSIHnCSL2 bit = 0). The active level can be specified individually for each CS.

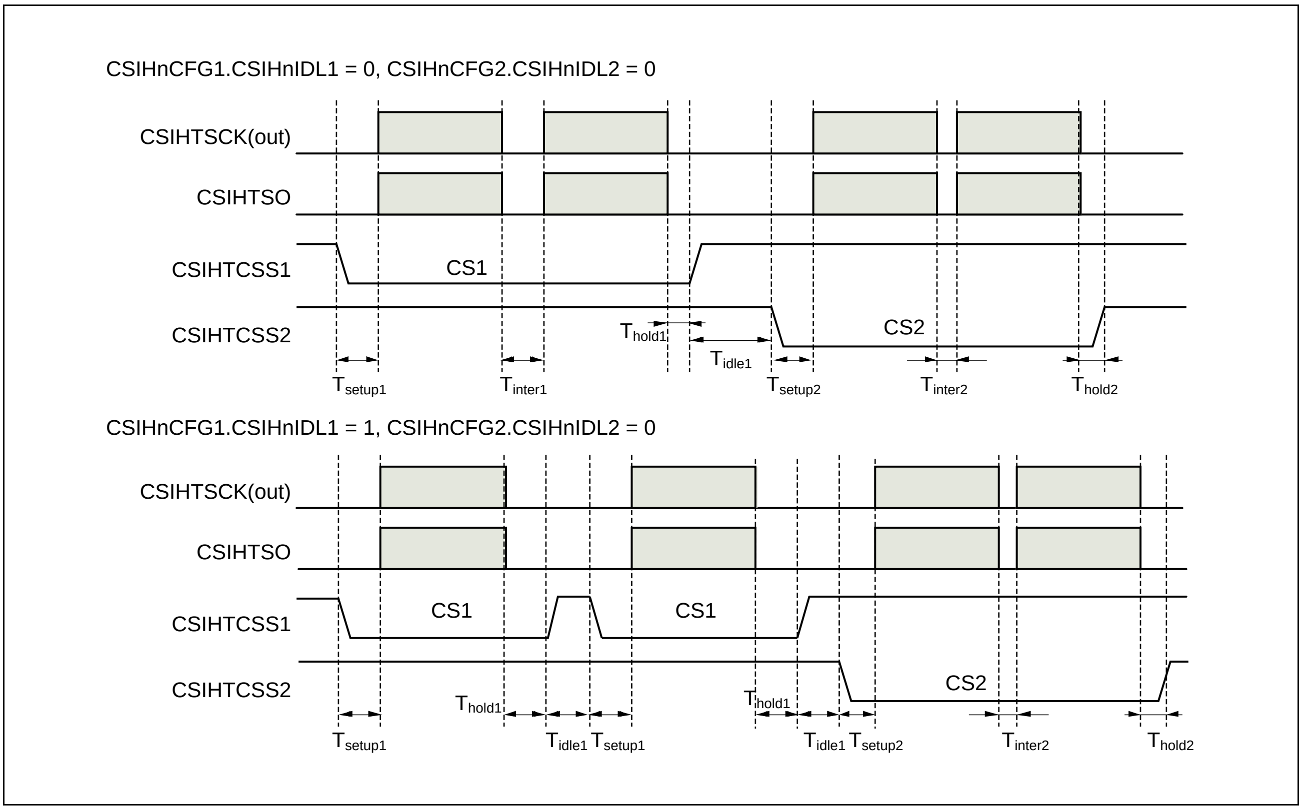


Figure 20.13 Chip Select Timings

Note that each CS signal can have a different value for the setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIHnTX0W.CSIHnCSx.

CSIHnRX0W.CSIHnCSx in the reception register indicates the chip select signal associated with the received data.

CAUTION

When high priority communication function by CPU control is enabled (CSIHnCTL1.CSIHnPHE = 1), IDLE state is inserted regardless of IDLn bit settings when priority communication mode is changed from low to high and from high to low.

20.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with a single slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority” and the priority of CS1 to “dominant: high priority”. Consequently, the second communication is conducted using the CS1 settings, which are set as dominant.

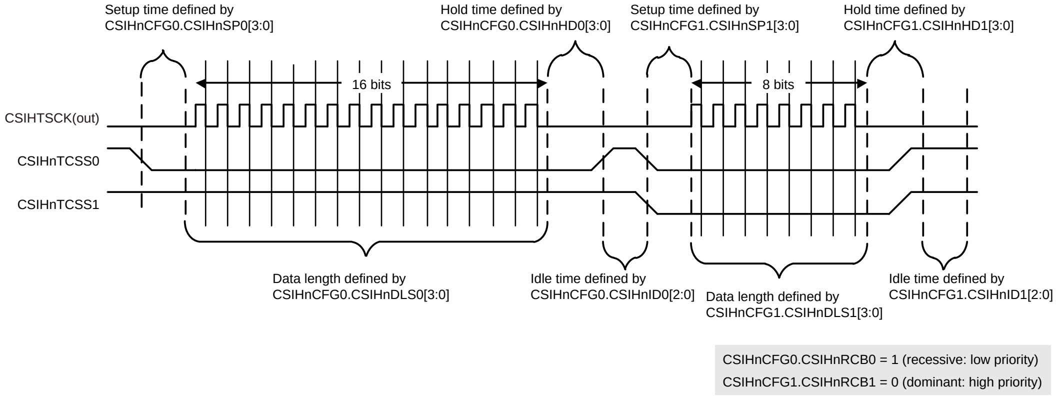


Figure 20.14 Chip Select and RCB Example

20.5.3.3 Job Concept

In CSIH, a job consists of the number of data targeted for transfer.

Job mode enable

The job mode can only be enabled in master mode. The job mode is enabled or disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

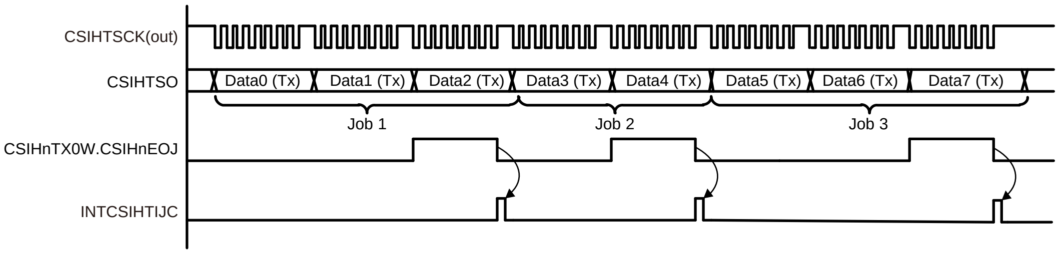


Figure 20.15 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

Communication can be specified to stop when a job is finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until the data for which the CSIHnEOJ bit is set is transmitted. After this data is sent, the communication is stopped and the job completion interrupt INTCSIHTIJC is generated.

20.5.4 Details of Chip Select Timing

20.5.4.1 Changing the Clock Phase

The serial clock level specified by CSIHnCFGx.CSIHnCKPx can be changed while communication is stopped. The minimum value of an idle time is one transmission clock (CSIHTSCK(out)) cycle.

If the idle time is set to 0.5 transmission clock cycles (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data is sent with different CSIHnCFGx.CSIHnCKPx configurations, the idle time is automatically extended to one CSIHTSCK(out) cycle.

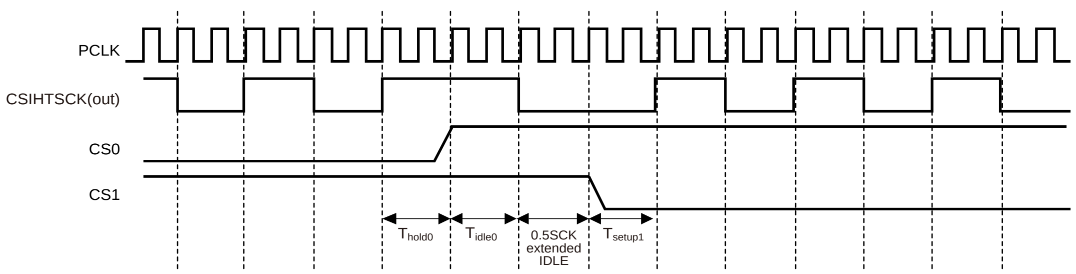


Figure 20.16 Clock Phase Timing with PCLK/4, $T_{hold}0$ = T_{setup1} = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

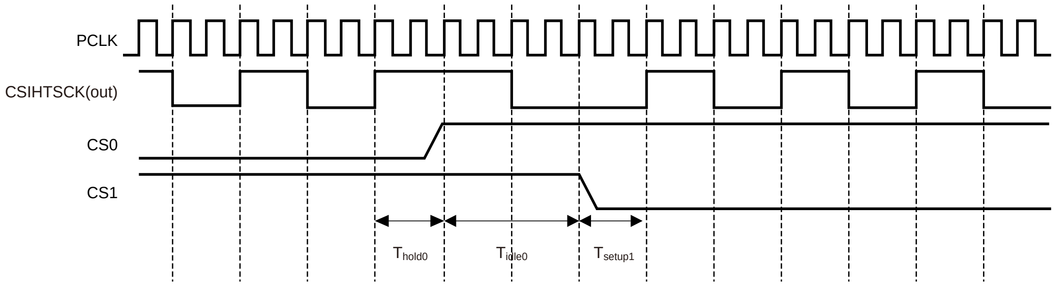


Figure 20.17 Clock Phase Timing with PCLK/4, $T_{hold}0$ = T_{setup1} = 0.5CSIHTSCK, T_{idle0} = 1CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

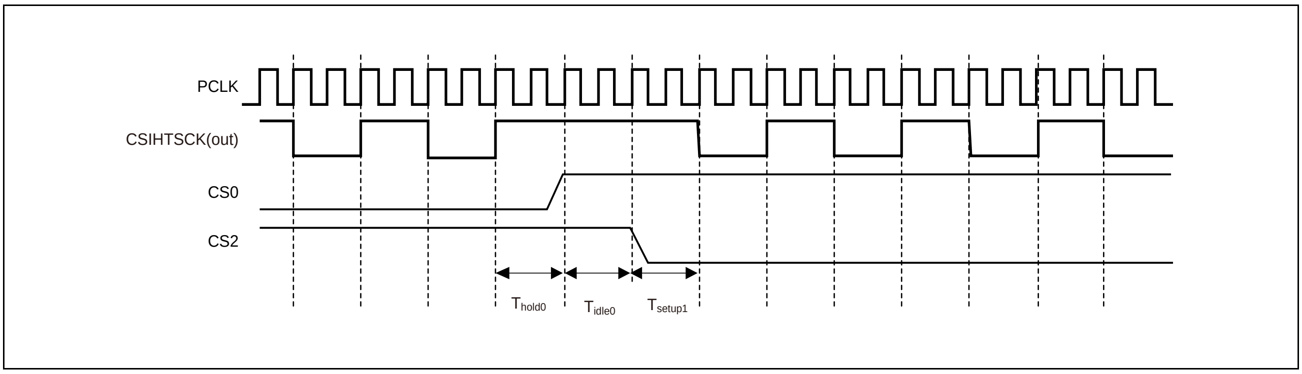


Figure 20.18 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

20.5.4.2 Changing the Data Phase

The CSIHnCFGx.CSIHnDAPx bit defines the phase of the data bits relative to the clock.

The relation between the setting of the CSIHnCFGx.CSIHnDAPx bit and the hold and setup times is as follows:

Hold time is the period from the last edge of the serial clock (CSIHTSCK) until the signals on CSIHTCSS[7:0] change to the inactive level.

Setup time is the period from when the signals on CSIHTCSS[7:0] change to the active level until to when the transmission data (CSIHTSO) is output.

Therefore, there is a gap of 0.5 CSIHTSCK cycles until the edge of the serial clock signal (CSIHTSCK) is output according to the CSIHnCFGx.CSIHnDAPx setting.

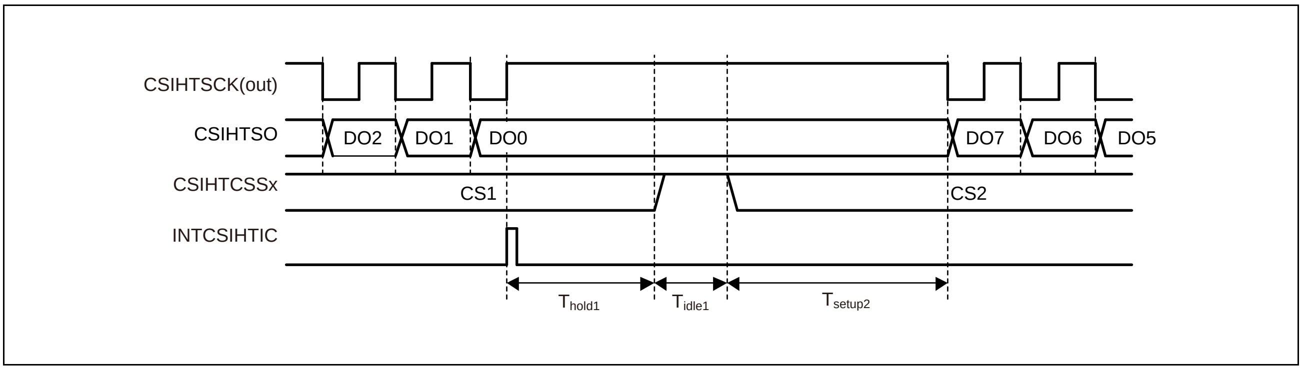


Figure 20.19 Data Phase Timing with
CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and
CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

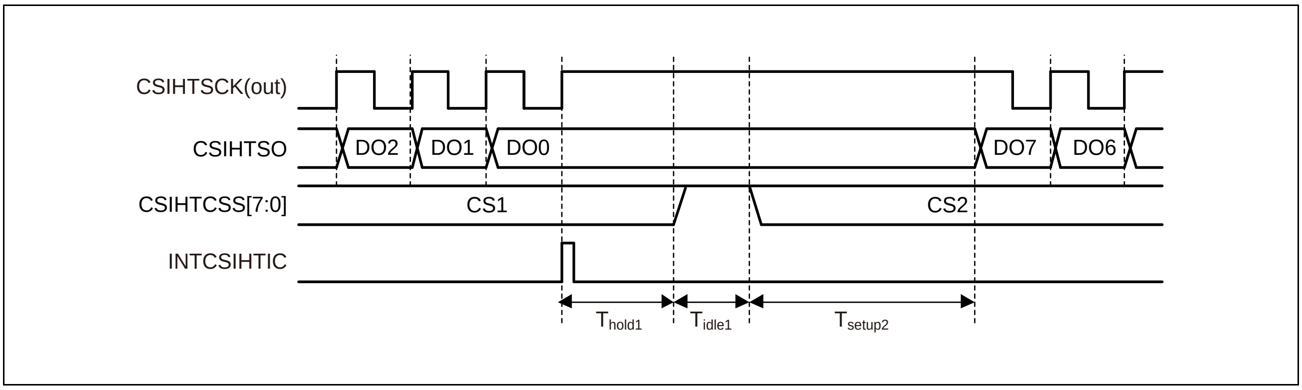


Figure 20.20 Data Phase Timing with
CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and
CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

20.5.5 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnBRSy.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnBRSSx[1:0]

The transfer clock frequency of transmission clock CSIHTSCK is determined by the setting of the CSIHnCTL2.CSIHnPRS[2:0] bits and the setting of the CSIHnBRSy.CSIHnBRS[11:0] bits, but any one of CSIHnBRS3 to CSIHnBRS0 can be selected for each chip select signal by using the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.

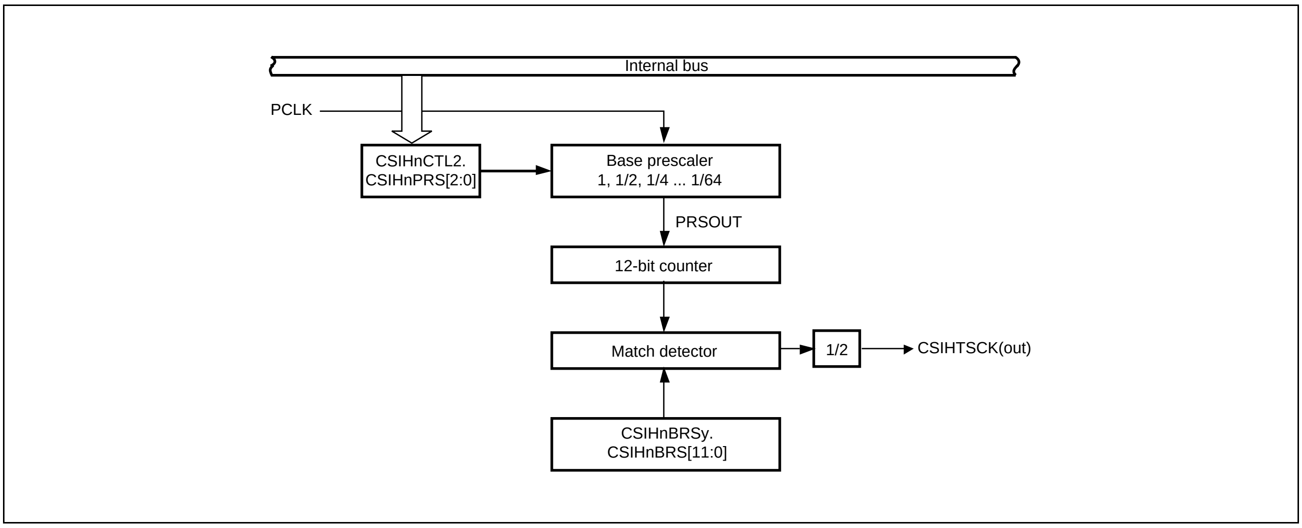


Figure 20.21 Baud Rate Generator Block Diagram

By setting CSIHnBRSy.CSIHnBRS[11:0] to 000_H, the baud rate generator is disabled and CSIHTSCK of the corresponding channels are stopped.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIHTSCK) = PCLK / (division ratio of PCLK) = PCLK / ($2^\alpha \times k \times 2$),

where:

α = CSIHnCTL2.CSIHnPRS[2:0] = 0 to 6

k = CSIHnBRS0.CSIHnBRS0[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 0)
CSIHnBRS1.CSIHnBRS1[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 1)
CSIHnBRS2.CSIHnBRS2[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 2)
CSIHnBRS3.CSIHnBRS3[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 3)

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the following:

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

20.5.6 CSIH Buffer Memory

The CSIH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word is comprised of 32-bit data and 7-bit ECC.

The following configurations are available:

Table 20.51 Configurations of CSIH buffer

Mode	CSIHnCTL0.CSIHnMBS	CSIHnMCTL0.CSIHnMMS[1:0]
FIFO mode	0	00_B
Dual buffer mode		01_B
Transmit-only buffer mode		10_B
Direct access mode	1	X

20.5.6.1 FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided the FIFO is not full.

Data to be transmitted is stored in the FIFO memory. Transmission and reception occur simultaneously – one data is received as one data is transmitted. That means, received data overwrites the transmitted data in the FIFO.

The CSIH automatically updates the respective FIFO memory pointers when data is written to or read from the FIFO memory, or data is transmitted to or received from the FIFO memory.

Table 20.52 FIFO Mode

Pointer Description	Pointer*[1]	Range
Number of untransmitted words	CSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of words received and stored in the FIFO	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address for write/read of transmit data	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
Address for read of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $01FC_H$
Address to be sent	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. The values are automatically updated after each read/write or data transmit/receive operation.

The CSIH status register contains two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. By doing this, CSIHnSTR0.CSIHnEMF is not reset, but set.

All FIFO pointers and FIFO flags except CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

20.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size; 64 words are allocated to transmission data and 64 words to received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

Table 20.53 Dual Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $00FC_H$
Address of data read from receive buffer	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $00FC_H$
The number of transmit data remaining in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 64
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $00FC_H$

Note 1. Pointers are automatically incremented after each read/write.

20.5.6.3 Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer indicates the values shown in the following table.

Table 20.54 Transmit-Only Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 128
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. Pointers are automatically incremented after each read/write.

20.5.6.4 Direct Access Mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

20.5.7 Data Transfer Modes

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 20.55 Start of Data Transfer

Memory and Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
FIFO, direct access	Master	Writing to the CSIHnTX0W register or the CSIHnTX0H register	Writing to the CSIHnTX0W register or the CSIHnTX0H register
	Slave	Clock reception from master	Incoming clock from the master
Transmit-only buffer, dual buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	CSIHnMCTL2.CSIHnBTST = 1
	Slave	Clock reception from master	Clock reception from master

20.5.7.1 Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

20.5.7.2 Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in transmit/receive mode.

Start of communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when the CSIHnMCTL2.CSIHnBTST bit is set.

20.5.7.3 Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in receive-only mode.

In master mode, start of reception depends on the memory mode:

- In case of FIFO or direct access mode, reception starts when dummy data is written to the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock is received from the master. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

- In case of dual buffer mode or transmit-only buffer mode, reception starts when the CSIHnMCTL2.CSIHnBTST bit is set.

20.5.8 Data Length Selection

20.5.8.1 Data Length from 2 to 16 Bits

The length of a data packet is selectable for each chip select signal from 2 to 16 bits using CSIHnCFGx.CSIHnDLSx[3:0]. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000_B):

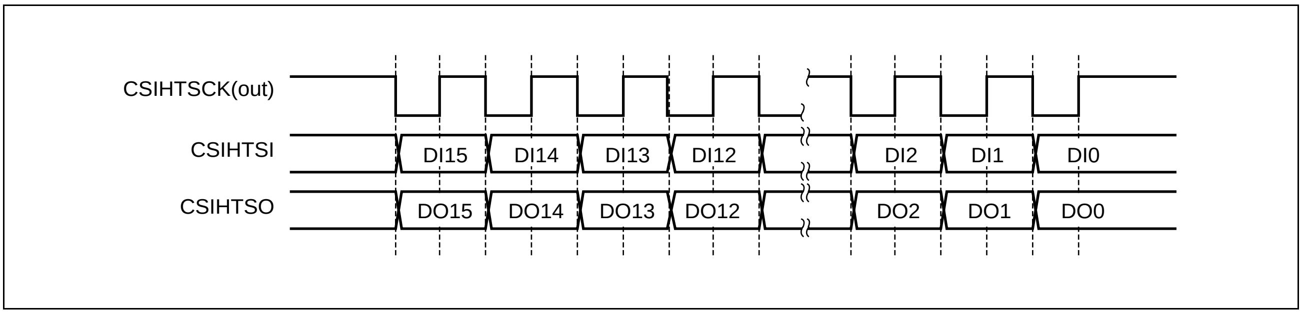


Figure 20.22 16 Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B):

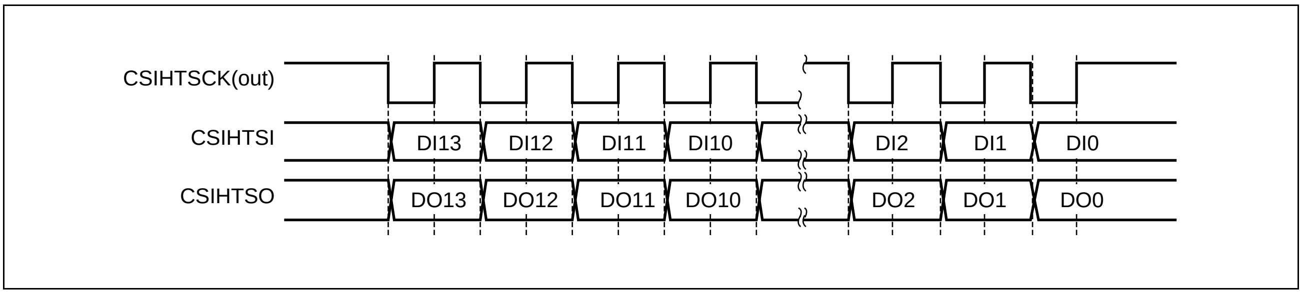


Figure 20.23 14 Bit Data Length, MSB First

20.5.8.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) function can be used.

EDL function is enabled by setting the CSIHnCTL1.CSIHnEDLE bit to 1.

EDL function works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been sent.

Example

Example for sending 40-bit data (123456789A$_H$) to CS0:

40 bits are split into two 16-bit blocks plus 8 bits.

- Initialize CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To transmit 123456789A$_H$ with MSB first, write the following sequence to CSIHnTX0W:
  - 20FE 1234$_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - 20FE 5678$_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - 00FE 009A$_H$ (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

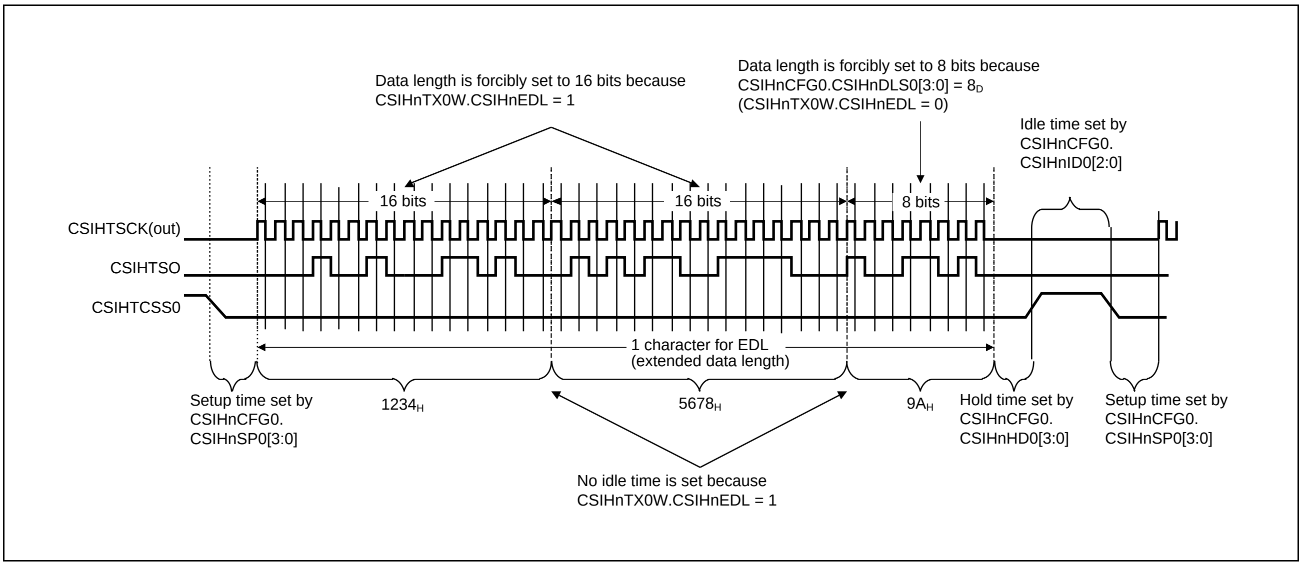


Figure 20.24 EDL Timing Diagram

NOTES

1. 1-bit data length is allowed only when using the EDL mode.
2. If parity is enabled, the parity bit is added after the last bit.
3. When data is sent using extended data length (EDL) function, use the same chip select signal.
4. Example for configuring the data direction:
   - Data to be sent: 123456_H
   - MSB first:
     Set CSIHnCFGx.CSIHnDIRx = 0
     Write CSIHnTX0W = 20FE 1234_H (EDL bit = 1)
     Write CSIHnTX0W = 00FE 0056_H (EDL bit = 0)
   - LSB first:
     Set CSIHnCFGx.CSIHnDIRx = 1
     Write CSIHnTX0W = 20FE 3456_H (EDL bit = 1)
     Write CSIHnTX0W = 00FE 0012_H (EDL bit = 0)
5. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to “1” while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
6. EDL mode cannot be used in receive-only mode of slave mode.
   (CSIHnCTL2.CSIHnPRS[2:0] = 111_B, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)

20.5.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).

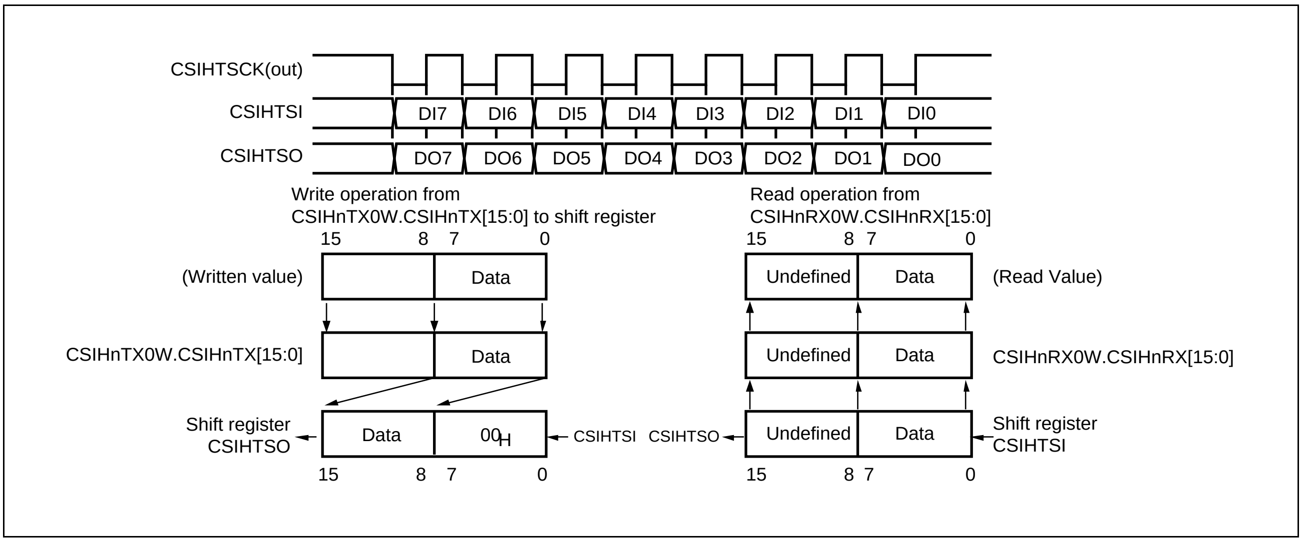


Figure 20.25 Serial Data Direction Select Function — MSB First (CSIHnDIRx = 0)

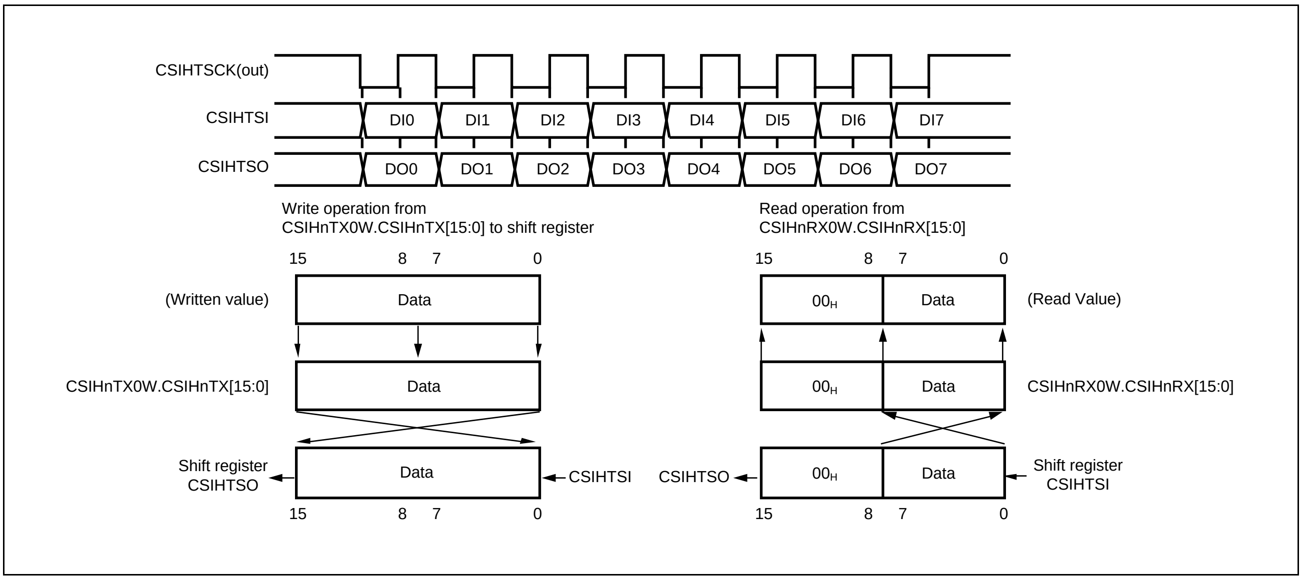


Figure 20.26 Serial Data Direction Select Function — LSB First (CSIHnDIRx = 1)

20.5.10 Slave Select (SS) Function

The Slave Select (SS) function enables communication between one master and multiple slaves (SPI communications).

In master mode, the master device outputs the slave select signal (CSIHTCSSx) to a slave. Communication by a device in slave mode is enabled when the slave input select signal (CSIHTSSI) is at the low level.

See **Section 20.5.2, Master/Slave Connections**, for examples of connections using the SS function.

20.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal and timings using the SS function.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

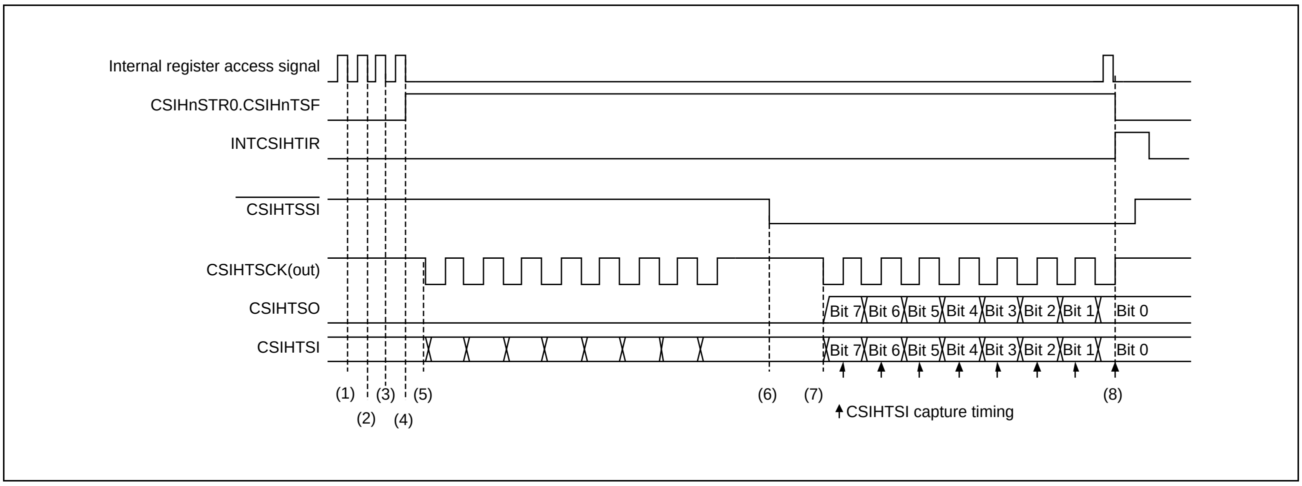


Figure 20.27 Tx/Rx Timing of Communication Using SS Function

(1) CSIH enters slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] = 111_B. CSIHnCFG0.CSIHnCKP0 and CSIHnCFG0.CSIHnDAP0 are 0.

(2) The data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B). The data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).

(3) The transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1, CSIHnCTL0.CSIHnRXE = 1, and CSIHnCTL0.CSIHnPWR = 1). Communication start is permitted.

(4) The transfer status flag CSIHnSTR0.CSIHnTSF is automatically set when transfer data is written to the CSIHnTX0W or CSIHnTX0H transmission register during direct access mode or FIFO mode.

(5) While the signal $\overline{\text{CSIHTSSI}}$ is at high level, transmission/reception is not started, even if an external transmission clock CSIHTSCK is input. Input to CSIHTSI is ignored.

(6) $\overline{\text{CSIHTSSI}}$ falling to low level indicates that CSIHTSO is enabled and ready for transmission.

(7) As soon as the external clock signal CSIHTSCK is detected, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI.

(8) Interrupt INTCSIHTIR indicates that the reception is complete. The CSIHnRX0W/H register can be read.

20.5.10.2 CSIHTSSO Operation

CSIHnPWR	CSIHnTXE	CSIHnRXE	CSIHnSSE	CSIHTSSO
0	—	—	—	H
1	—	—	0	H
	0		1	H
	1		1	Inverse value of $\overline{\text{CSIHTSSI}}$ level

The CSIHTSSO pin is a signal to control the I/O function of the chip’s SO pin when using the SS function.

The CSIHTSO pin is enabled when the CSIHTSSO pin is “High” (the chip’s SO pin is being driven).

The CSIHTSO pin is disabled when the CSIHTSSO pin is “Low” (the chip’s SO pin is not being driven).

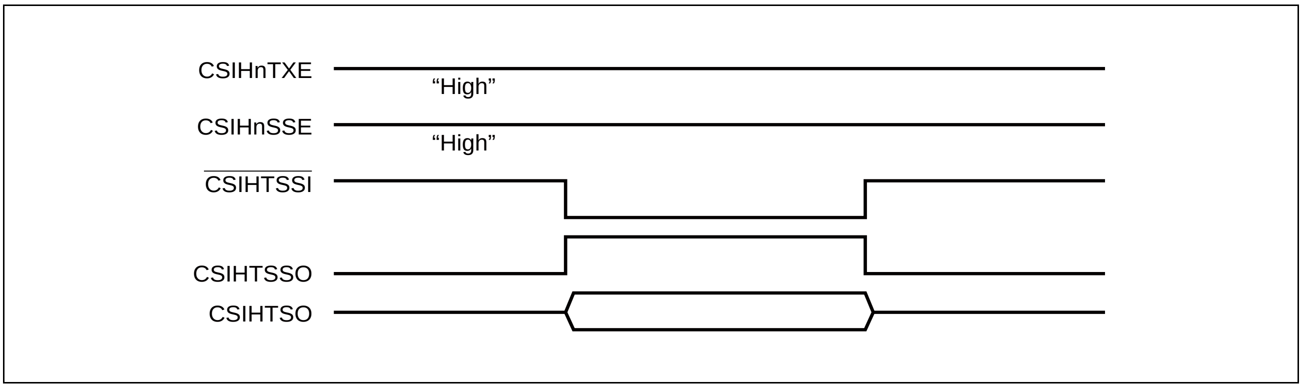


Figure 20.28 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not guaranteed.

20.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIHnCTL1.CSIHnHSE bit. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

20.5.11.1 Slave Mode

If CSIHnCTL1.CSIHnHSE = 1, a low-level CSIHTRYO signal is output when the slave becomes busy. This can happen in two cases:

1. When the next data to be sent is not ready:
   When the slave is in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1) and is in any of the states listed below, the CSIHTRYO outputs the busy state (low level).

Table 20.56 Memory Mode and Slave Transfer State

Memory Mode	Slave Transfer State
Direct access mode	When there is no more data to be sent
FIFO mode	When there is no more data to be sent (CSIHnSTR0.CSIHnEMF = 1)
Dual buffer mode	When CSIHnMCTL2.CSIHnBTST is not set to 1
Transmit-only buffer mode	

The following examples assume an eight-bit data length.

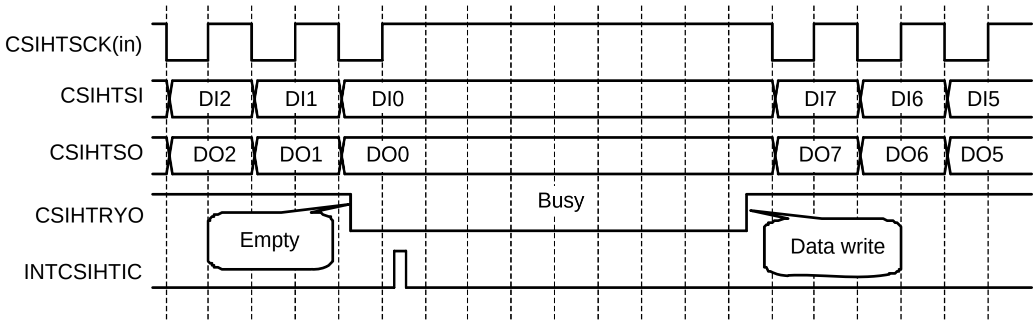


Figure 20.29 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)

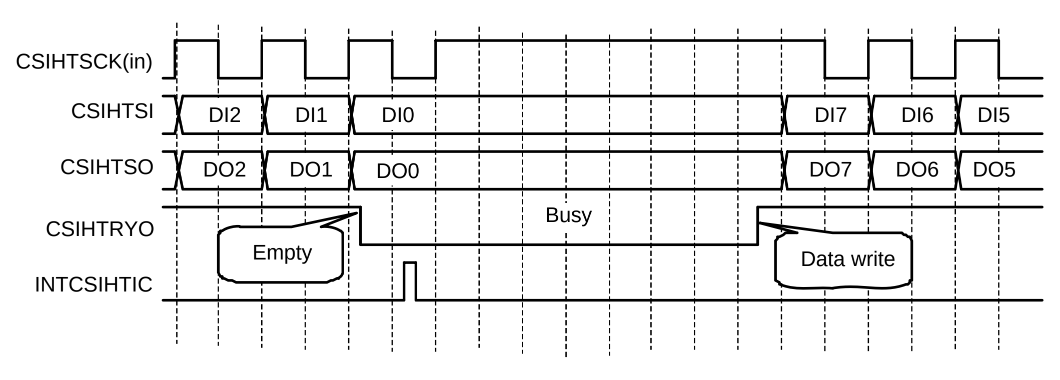


Figure 20.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)

2. When receive register is full:
   When slave is configured in receive-only mode or transmit/receive mode (CSIHnCTL0.CSIHnRXE = 1), and new data cannot be copied from a shift register to CSIHnRX0W/H because the previously received data is still in the CSIHnRX0W/H register (CSIHnRX0W/H is full).
   When CSIHnCTL0.CSIHnRXE is 1 and is in any of the following states, CSIHTRYO outputs busy state (low level).

Table 20.57 Memory Mode and Slave Reception State

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full
FIFO mode	When received data is remaining in buffer (CSIHnSTR0.CSIHnFLF = 1)
Dual buffer mode	No applicable case
Transmit-only buffer mode	When CSIHnRX0W or CSIHnRX0H is full

The following examples assume an eight-bit data length.

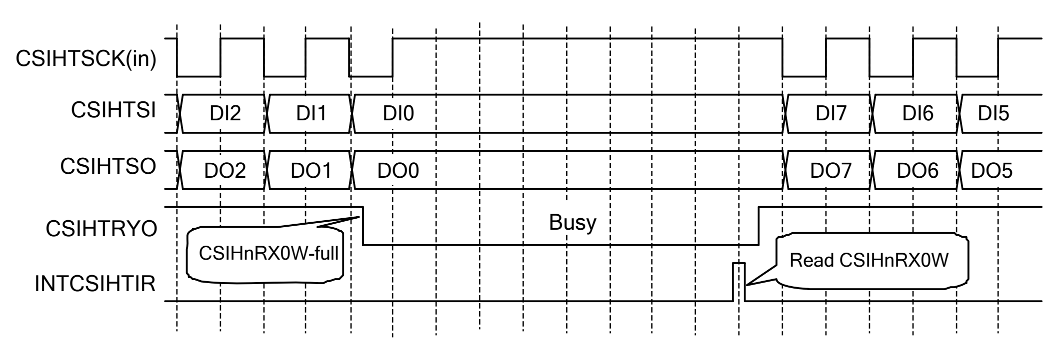


Figure 20.31 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

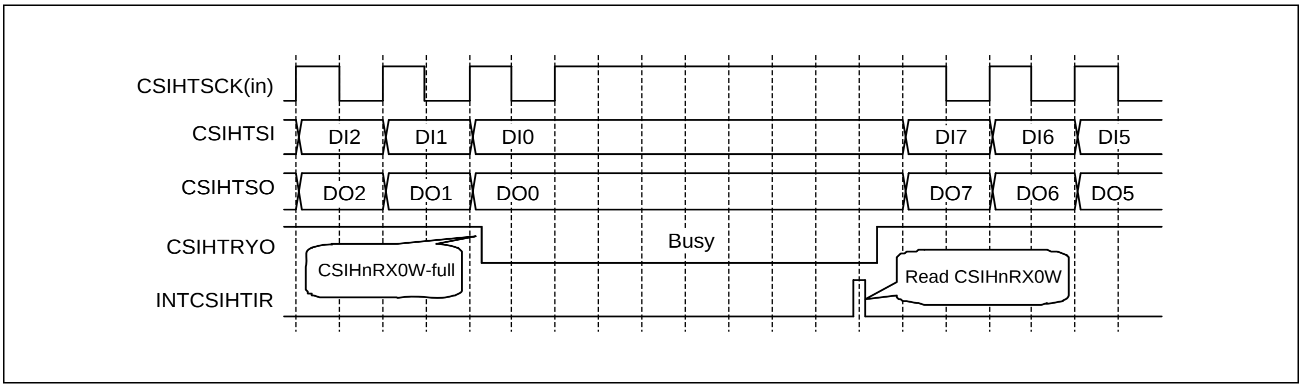


Figure 20.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1)

20.5.11.2 Master Mode

When the master detects CSIHTRYI = 0 while CSIHnCTL1.CSIHnHSE = 1, the subsequent transfers are put on hold, the master goes into wait state and suspends the CSIHTSCK clock.

The CSIHTRYI level is checked at each half clock cycle of CSIHTSCK.

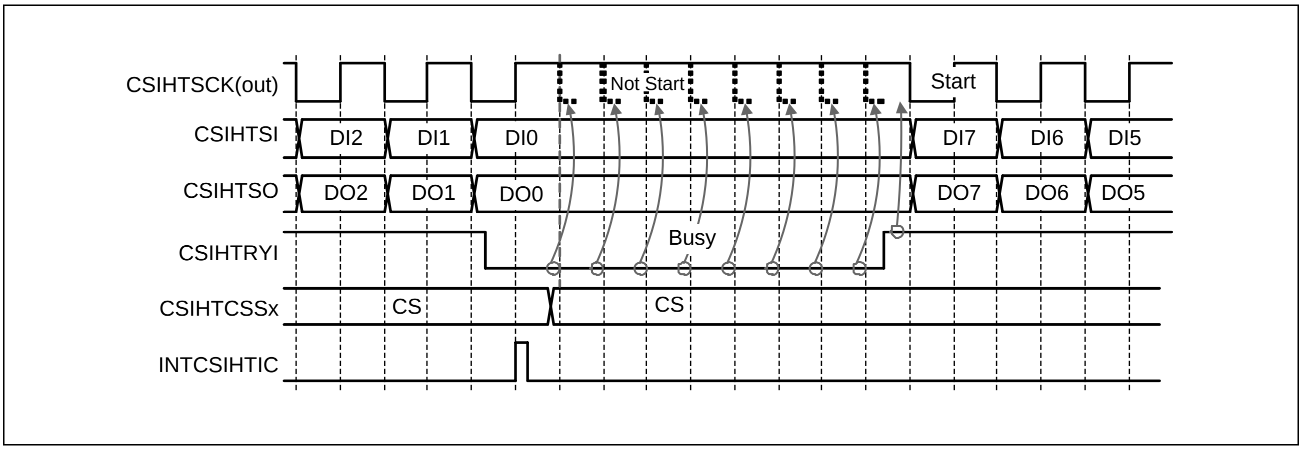


Figure 20.33 Master’s Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 0)

The CSIHTRYI signal must be pulled down by the slave before the next transfer starts. If this is done while data transfer is in progress, the serial clock from the master is suspended after the transfer is complete.

The master resumes the communication as soon as CSIHTRYI becomes high (the slave is “ready”).

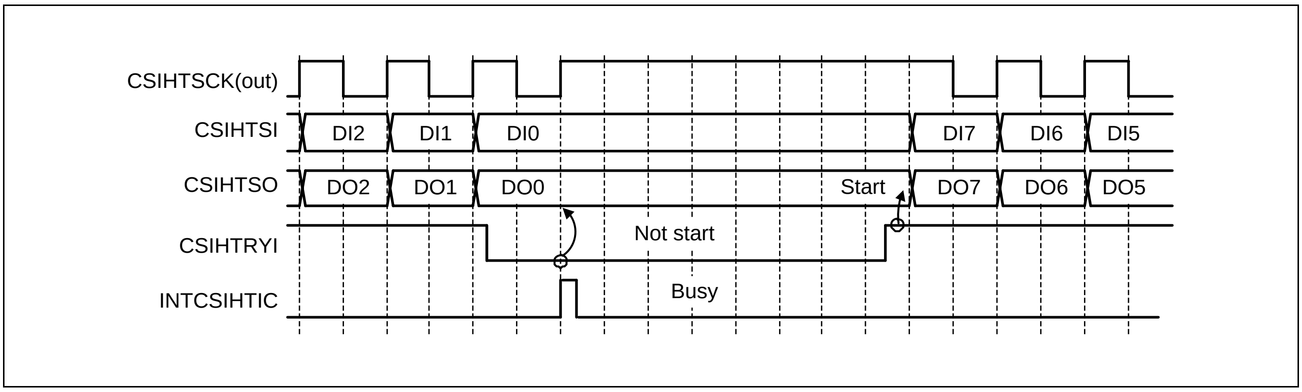


Figure 20.34 Master’s Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 1)

CAUTIONS

1. If multiple slaves are connected, the master must only detect the CSIHTRYI signal of the slave it has selected for communication.
2. Even when the CSIHTRYI pin of the master detects a CSIHTRYO signal from the slave during data transfer, the communication is not put on hold but continues until the data transfer is completed.

20.5.12 Error Detection

CSIH can detect five error types:

- Data consistency check error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and time-out errors can be enabled/disabled individually.

If any of these errors is detected, the interrupt request INTCSIHTIRE is generated and the corresponding flags are set.

20.5.12.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn_m = 1). It will not be enabled if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIHTSO are read back via the CSIHTDCS signal into the corresponding shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency check error and:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

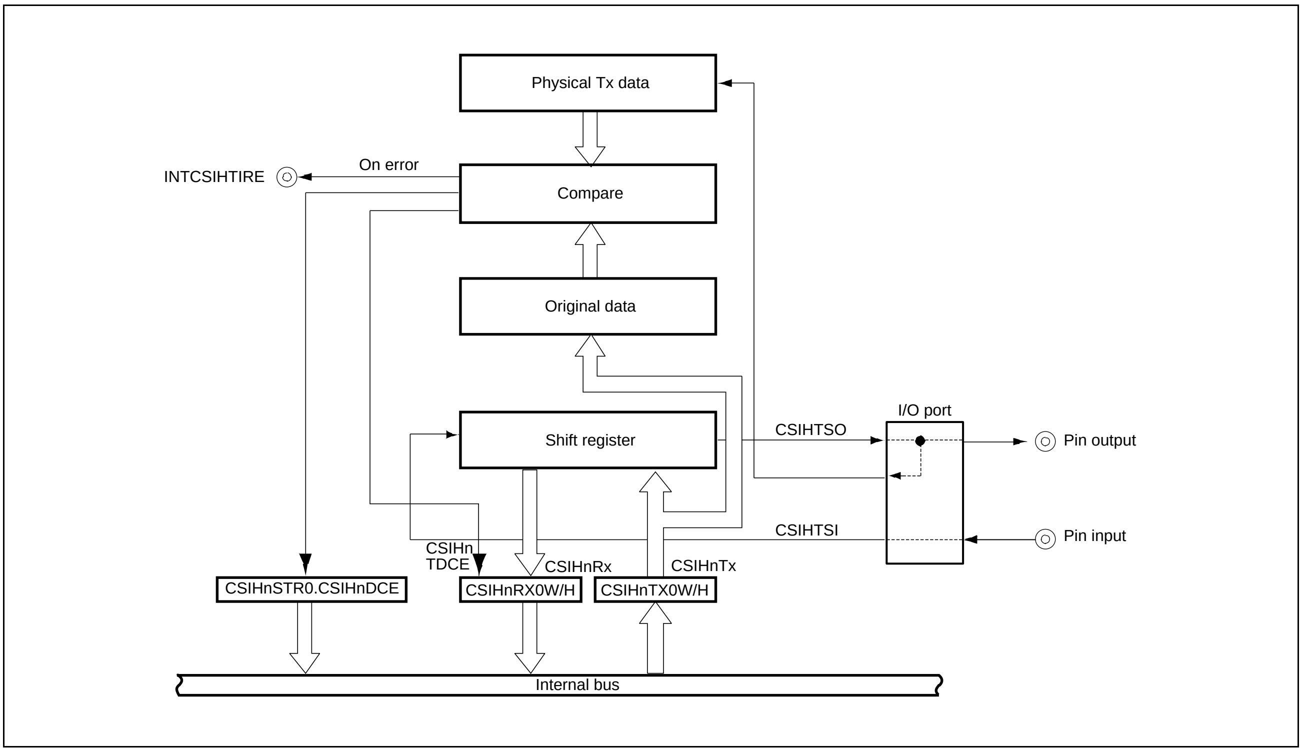


Figure 20.35 Data Consistency Check Functional Block Diagram

20.5.12.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

The parity bit is checked after a reception is complete. In case of parity error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnPE bit is set.

Additionally, CSIHnRX0W.CSIHnRPE of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is 05_H and 35_H.
- Data direction is LSB first.
- Parity type is odd.

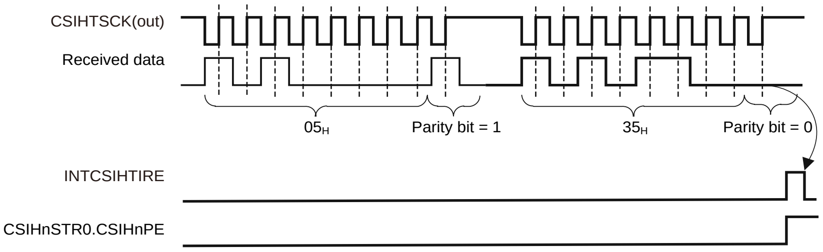


Figure 20.36 Parity Check Example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

20.5.12.3 Time-Out Error

Time-out errors can be checked only in slave FIFO mode.

This error occurs if neither of the following occurred within a certain period of time:

- Received data in FIFO is read
- FIFO receives data from CSIHTSI

The time-out time is defined in CSIHnMCTL0.CSIHnTO[4:0] in units of “8 x transmission clock CSIHTSCK”. A time-out error occurs when the specified time is exceeded (the time-out time is not detected when CSIHnMCTL0.CSIHnTO[4:0] = 00000_B).

The dedicated time-out counter is set by the CSIHnCTL2.CSIHnPRS[2:0] and CSIHnBRSy.CSIHnBRS[11:0] bits.

If the value of the CSIHnBRSy.CSIHnBRS[11:0] bits is left as 000_H, the dedicated time-out counter does not operate.

The dedicated time-out counter measures the time between the last and the next read operation.

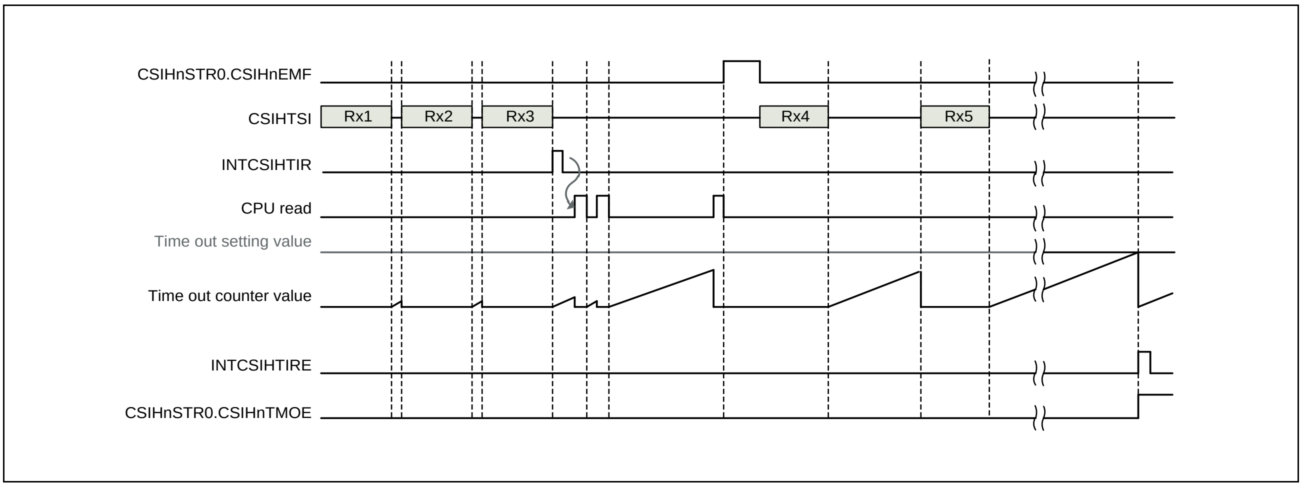


Figure 20.37 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed
- When data read from the CPU is completed
  (The counter does not start if the buffer is empty.)
- When a time-out error is detected

After a time-out error is detected, if data is still available in FIFO, the time-out counter restarts.

If the value set by bits CSIHnMCTL0.CSIHnTO[4:0] is reached again, the INTCSIHTIRE interrupt is output again.

The timeout counter continues to count until received data is read. To stop the counter, read all received data or set CSIHnSTCR0.CSIHnPCT to 1. Note that the pointer is cleared if you perform the latter.

The counter is reset at the following timing:

- Data is read once.
- A new data item is received.
- A timeout error is detected.
- The CSIHnSTCR0.CSIHnPCT bit is set to 1.

If a timeout error occurs, the following occur:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnTMOE bit is set.

20.5.12.4 Overflow Error

An overflow error can occur in FIFO mode. It occurs when transmission data is written to the CSIHnTX0W register while the FIFO buffer is filled with received data.

Example

100 data packets have been transmitted. That means, the FIFO contains 100 received data packets. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data packets to the FIFO. However, only 10 received data packets have been read up to now, and 90 are still in the FIFO.

In this case, only 38 buffers are available for new transmission data. When the CPU tries to write the 39th data, an overflow error occurs.

This is illustrated in the following figure.

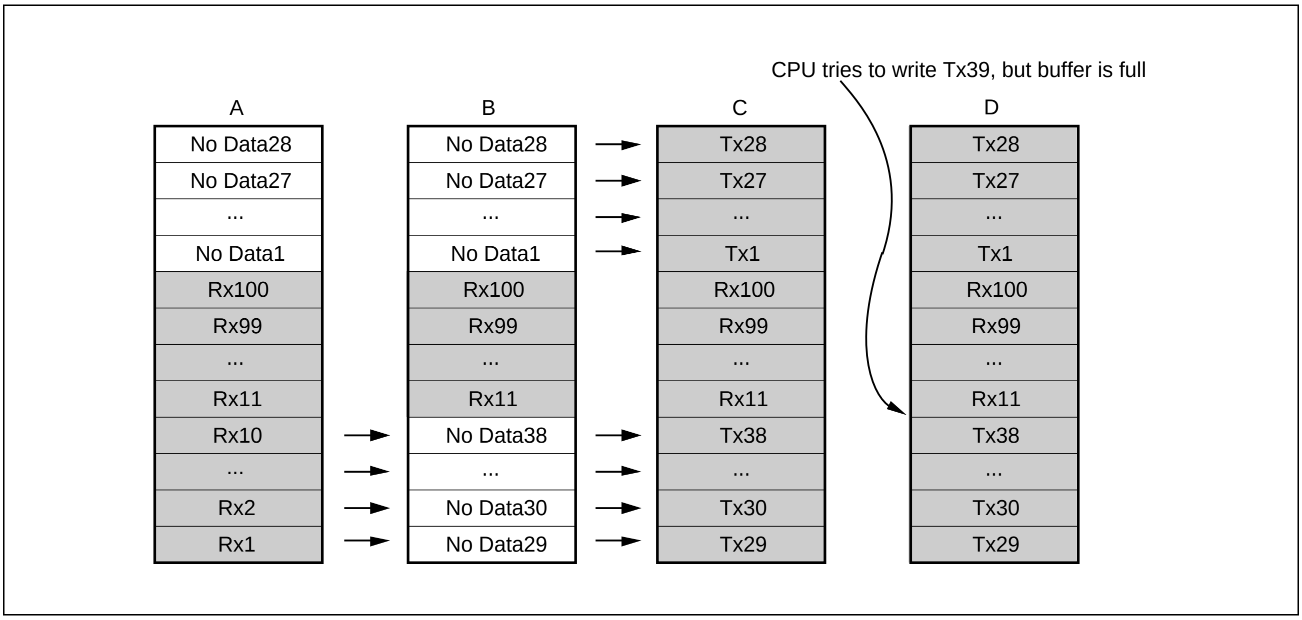


Figure 20.38 FIFO Overview

The 39th and subsequent data packets are discarded. The figure below shows the overflow timing.

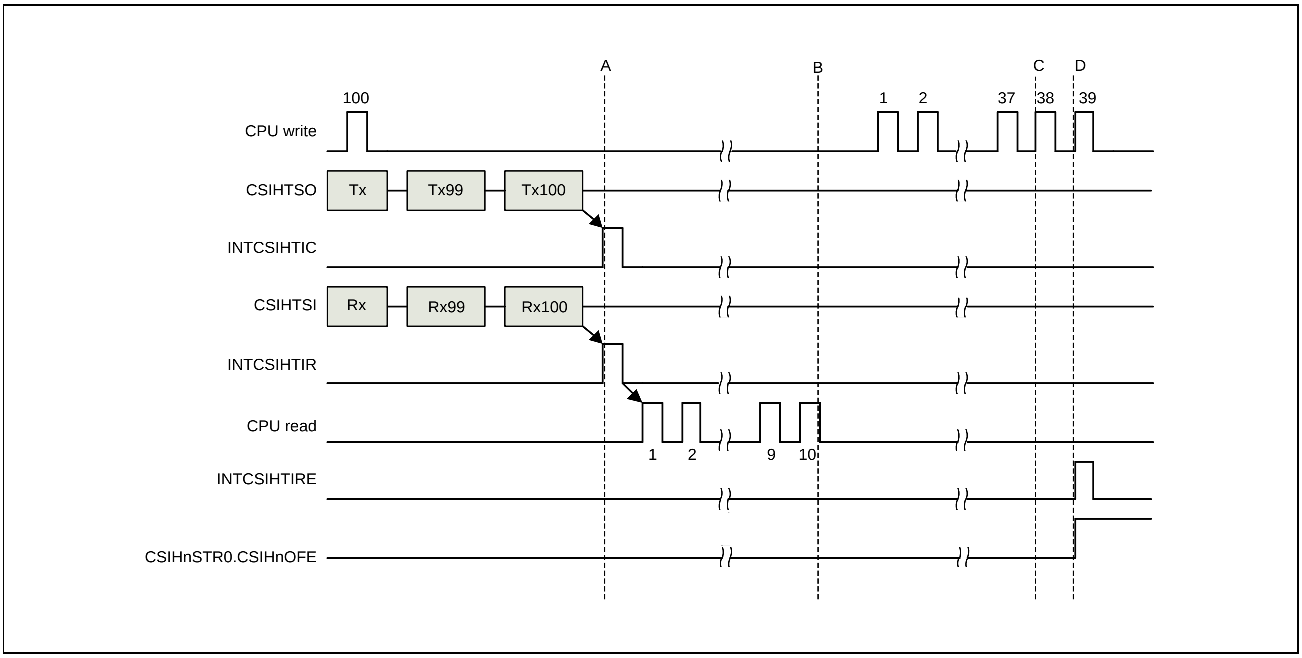


Figure 20.39 FIFO Overflow Timing

In case of overflow error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOFE bit is set.

20.5.12.5 Overrun Error

An overrun error can occur in direct access, transmit-only buffer, and FIFO modes. It does not occur in dual buffer mode. The overrun error does not occur if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are two conditions for overrun errors.

Condition for errors 1

- In FIFO mode, if the CPU reads the CSIHnRX0W/H register when the number of received data is 0.

Condition for errors 2

- In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled):
  - In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data remains in the CSIHnRX0W/H register.
  - In FIFO mode, when reception is completed while the FIFO buffer is still full of receive data.

(1) Direct access/transmit-only buffer

In direct access and transmit-only buffer modes, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W/H. This happens when CSIHnRX0W/H was not read and therefore contains previously received data.

The following figure illustrates the overrun error detection function.

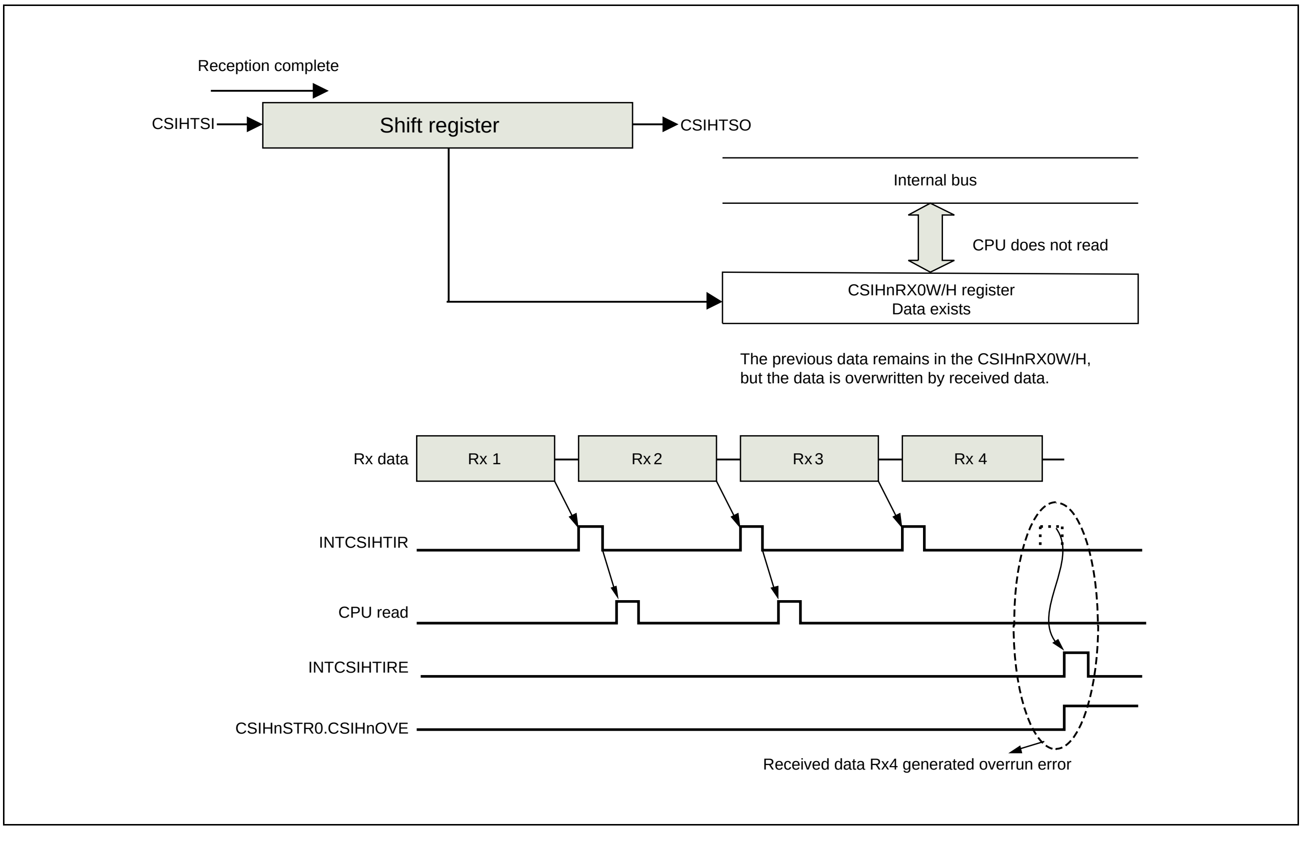


Figure 20.40 Overrun Error Detection in Direct Access and Transmit-Only Buffer Modes

NOTE

An overrun error can be avoided in slave mode by using handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

(2) FIFO mode

In FIFO mode, this error occurs if:

1. Newly received data cannot be transferred from the shift register to the FIFO because the FIFO is full.

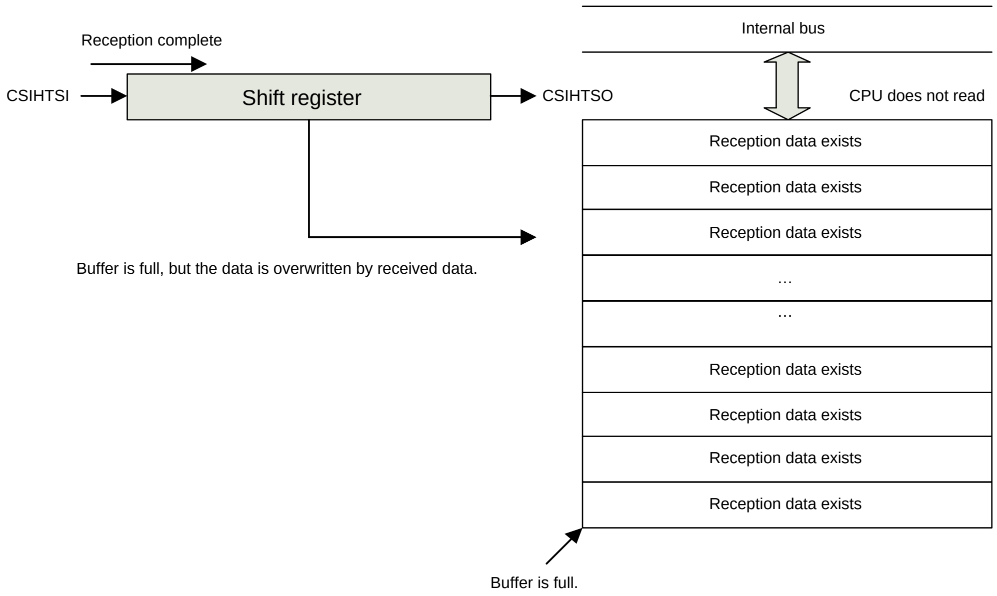


Figure 20.41 Overrun Error Detection in FIFO Mode (FIFO Full)

NOTE

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

2. The CPU attempts to read non-existent receive data.

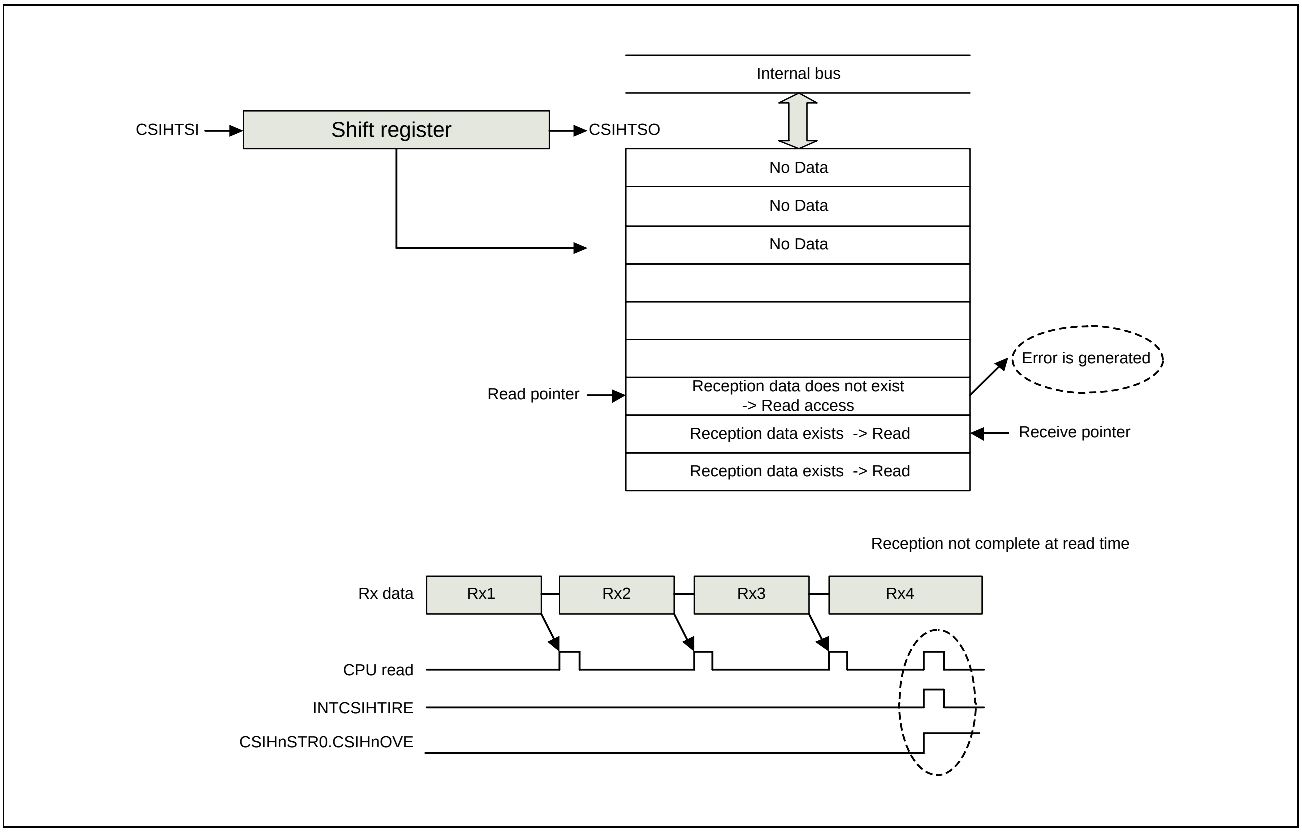


Figure 20.42 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOVE bit is set.
- Received data is overwritten and the communication continues.
  (When the CPU tries to read non-existent data, it waits until reception is completed and then resumes reading.)

For details see **Section 20.5.11, Handshake Function**.

20.5.13 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHTCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIHnCSLx value). The transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHTSCK, CSIHTSO, CSIHTSI, and CSIHTCSSx are disconnected from the ports. In addition, the CSIHTSO output level is fixed to low, and CSIHTSCK is set to reset level (High) regardless of the value of the CSIHnCFGx.CSIHnCKPx. The rest of CSIH works as in normal operation.

In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. Any connected device remains unaffected by the loop-back test.

Table 20.58 Pin Output Level in Loop-Back Mode

Pin Name	Output Level
CSIHTSCK(out)	High level
CSIHTCSS[7:0]	Inactive level
CSIHTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIHTRYO	Normal function (Low level)

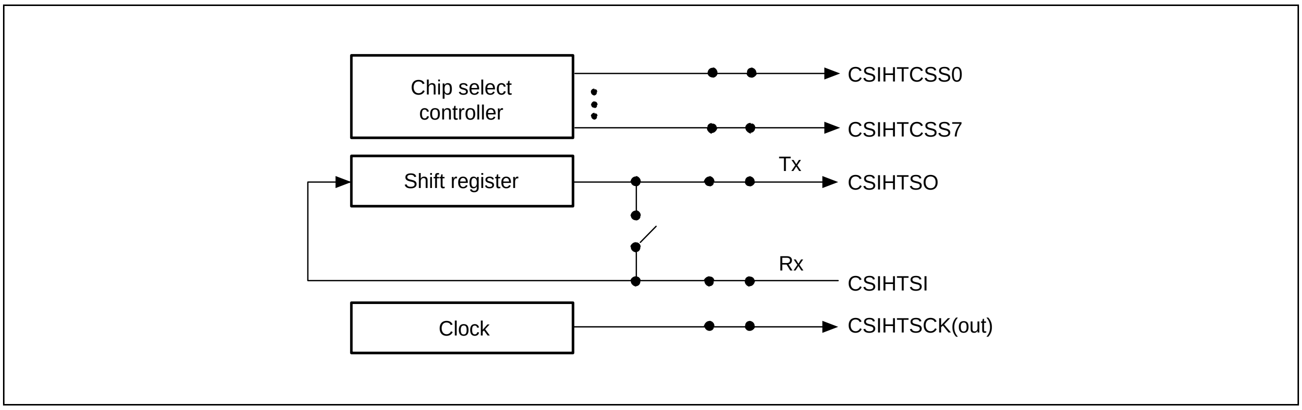


Figure 20.43 Normal Operation

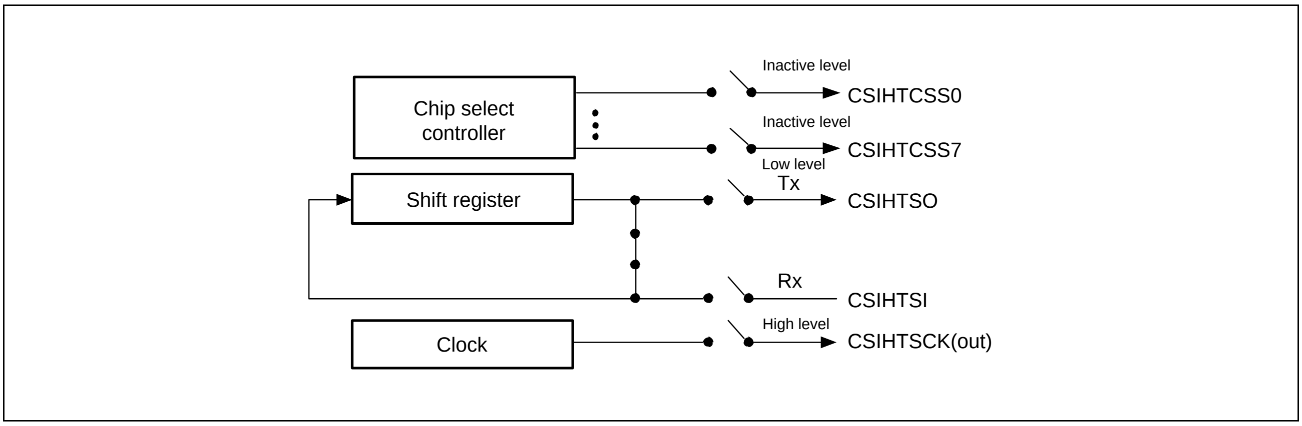


Figure 20.44 Loop-Back Mode Operation

20.5.14 CPU-Controlled High Priority Communication Function

CSIH has a function to abort low priority communication to perform high priority communication if it receives a high-priority communication request from the CPU while low-priority communication is being used. This function supports transmit-only buffer mode as low priority communication and only direct access mode as high-priority communication. To enable this function, CSIHnCTL1.CSIHnPHE and CSIHnCTL1.CSIHnJE must be set to 1.

The following figure illustrates CPU-controlled high-priority communication.

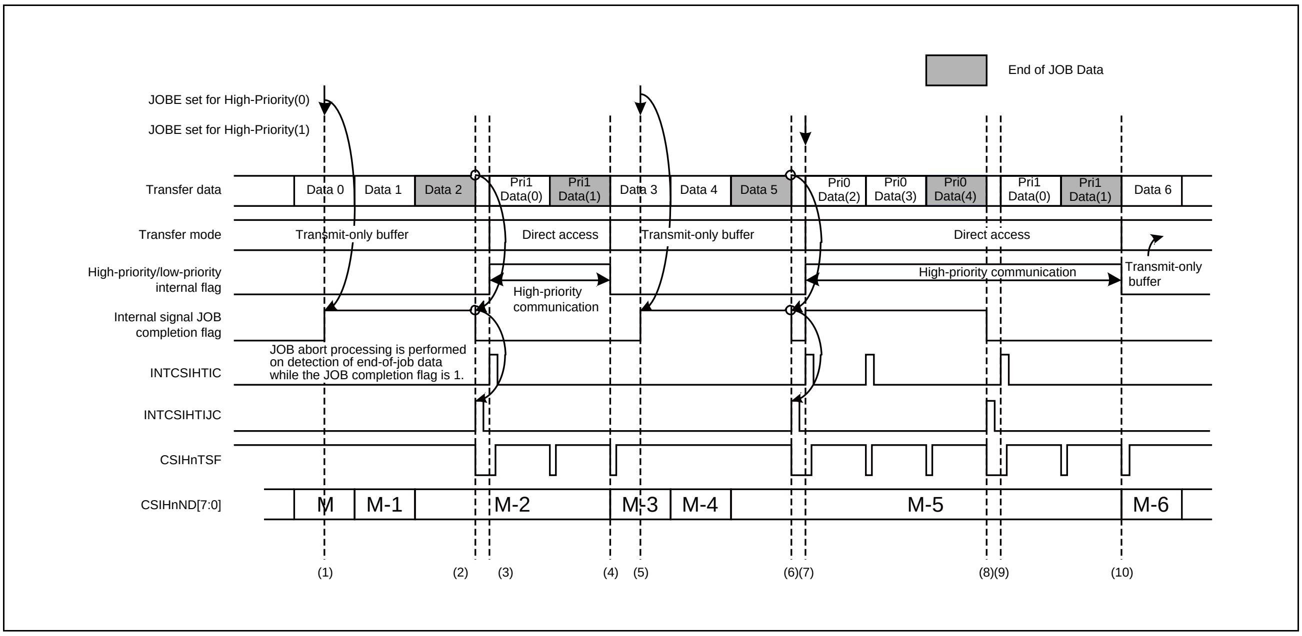


Figure 20.45 Example of CPU-Controlled High-Priority Communication

(1) By setting CSIHnCTL0.CSIHnJOBE = 1 during low-priority communication, start of high-priority communication following end-of-job data is notified, and the internal signal flag is set.

(2) When end-of-job data is detected, the current low-priority communication is aborted and the INTCSIHTIJC interrupt occurs. The internal signal job completion flag is cleared due to the communication abort, and memory mode is automatically switched to direct access mode for the subsequent high-priority communication.

(3) The CPU detects the interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H.

(4) When end-of-job data is detected, communication is aborted. At this time, because the internal signal end-of-job flag is set to 0, the CSIH determines that the next communication is low-priority, automatically switches memory mode to transmit-only buffer mode, and then resumes the aborted low-priority communication.

(5) Same as (1) above.

(6) Same as (2) above.

(7) The CPU detects an interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H. The CPU sets CSIHnCTL0.CSIHnJOBE = 1 again to notify that the next communication is high-priority.

(8) When end-of-job data is detected, communication is aborted and the INTCSIHTIJC interrupt is generated. At this time, the CPU determines that the subsequent communication is also high-priority because the internal signal JOB completion flag is 1, and waits for communication to start.

(9) Same as (3) above.

(10) Same as (4) above.

CAUTION

Memory mode is switched automatically when communication is changed from low priority to high priority (from transmit-only buffer mode to direct access mode) and from high priority to low priority (from direct access mode to transmit-only buffer mode).

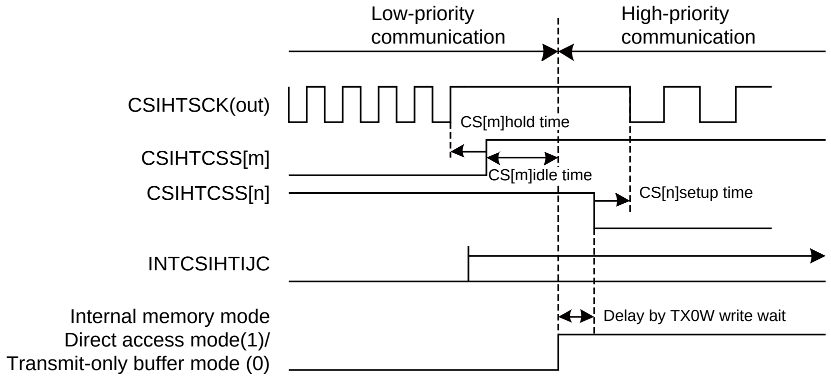


Figure 20.46 Transition from Low-Priority Mode to High-Priority Mode

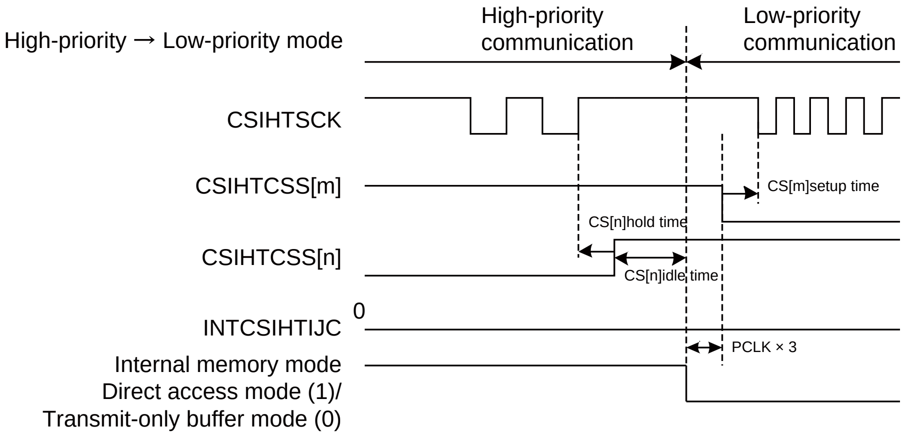


Figure 20.47 Transition from High-Priority Mode to Low-Priority Mode

Do not conduct write operation of communication data or CSIHnCTL0.CSIHnJOBE bit operation during setting inhibited period to switch low and high priority communication mode correctly.

CSIHnTX0W register write inhibited period:

- Period from when CSIHnJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.
- Period from when the last data of high priority communication (End of JOB data) is written to when the CSIHnHPST state = 0 is detected.

CSIHnJOBE register write inhibited period:

- Period from when CSIHnJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.

During high communication mode period, there is no setting inhibited period for CSIHnJOBE bit. It is possible to set CSIHnJOBE bit before writing communication data. For example, to communicate multiple JOB data in high priority mode, it is possible to set CSIHnJOBE bit before writing the first communication data.

CAUTION

When CSIHnJOBE bit is set right before the last communication of high priority communication ends, different operations are required depending on the timing at which setting of CSIHnJOBE bit is internally detected.

If the setting of the CSIHnJOBE bit is detected before the transfer of the last bit is completed, high priority communication mode continues.

When setting of the CSIHnJOBE bit is detected after the transfer of the last bit is completed, the mode temporarily returns to low priority communication. After detection of End of JOB data in low priority communication, the mode changes back to high priority communication.

20.5.15 Enforced Chip Select Idle Setting

It is possible to insert an idle state between two consecutive data transfers by setting CSIHnCFGx.CSIHnIDLx.

1. When CSIHnCFGx.CSIHnIDLx = 0
   If the next CSIHTCSSx is the same as the previous one, an idle state is not inserted, but an inter-data time is inserted instead.
   If the next CSIHTCSSx is different from the previous one, an idle state is inserted.
2. When CSIHnCFGx.CSIHnIDLx = 1
   An idle state is always inserted even if a next CSIHTCSSx is not different from the previous one.

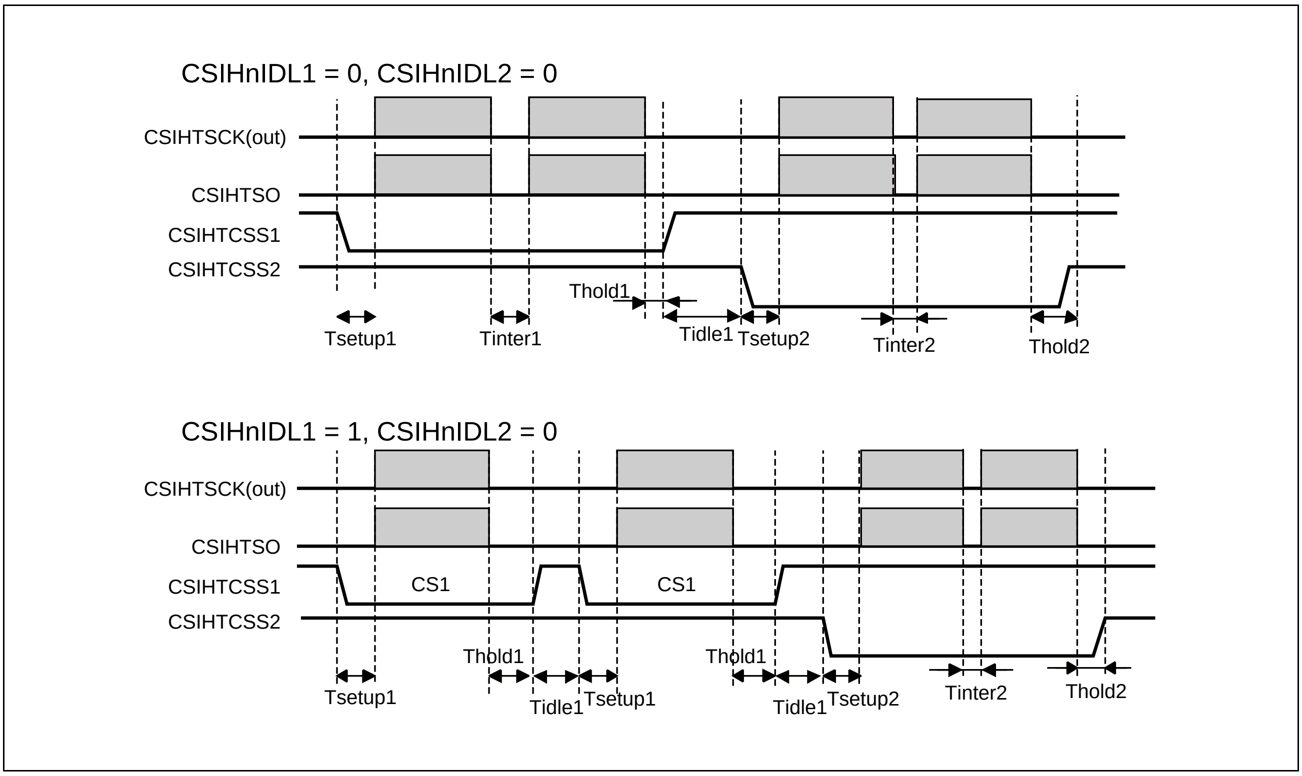


Figure 20.48 Enforced Chip Select Idle Setting Example

CAUTION

If the CPU-controlled high priority communication function is enabled

(CSIHnCTL1.CSIHnPHE = 1), when the mode is switched from low priority communication to high priority communication or from high priority communication to low priority communication, an IDLE state is inserted regardless of the setting of CSIHnCFGx.CSIHnIDLx bit.

20.6 Operating Procedures

The examples and procedures below are described according to the memory mode in the following order:

- Direct access mode
- Transmit-only buffer mode
- Dual buffer mode
- FIFO mode

20.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other with job mode enabled.

20.6.1.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below assumes the following conditions:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

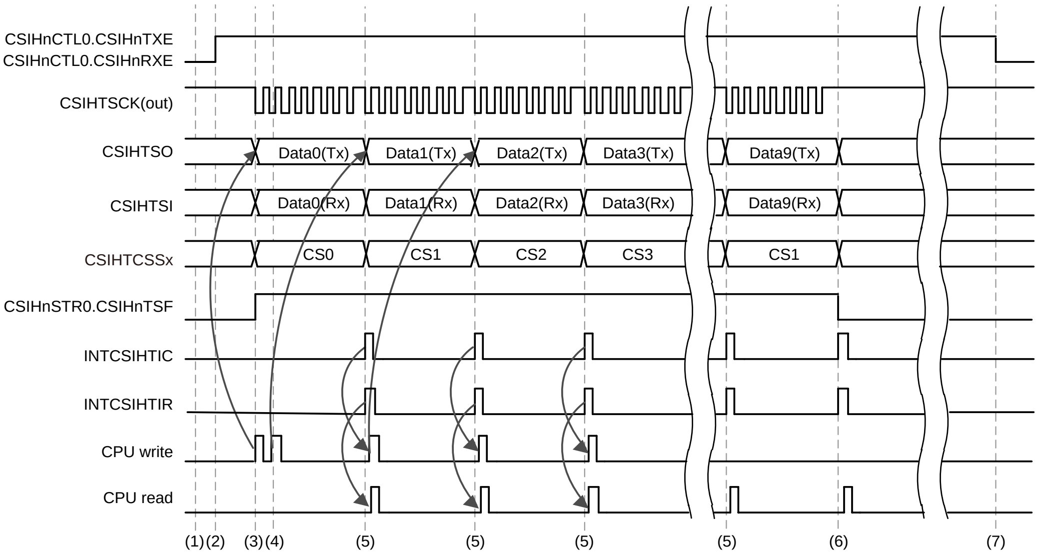


Figure 20.49 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register, CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated:
   - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   - INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. No more write action is required after completion of data 8. Data 9 (the last data) has been written in advance. However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.1.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs, each transmitting three data.

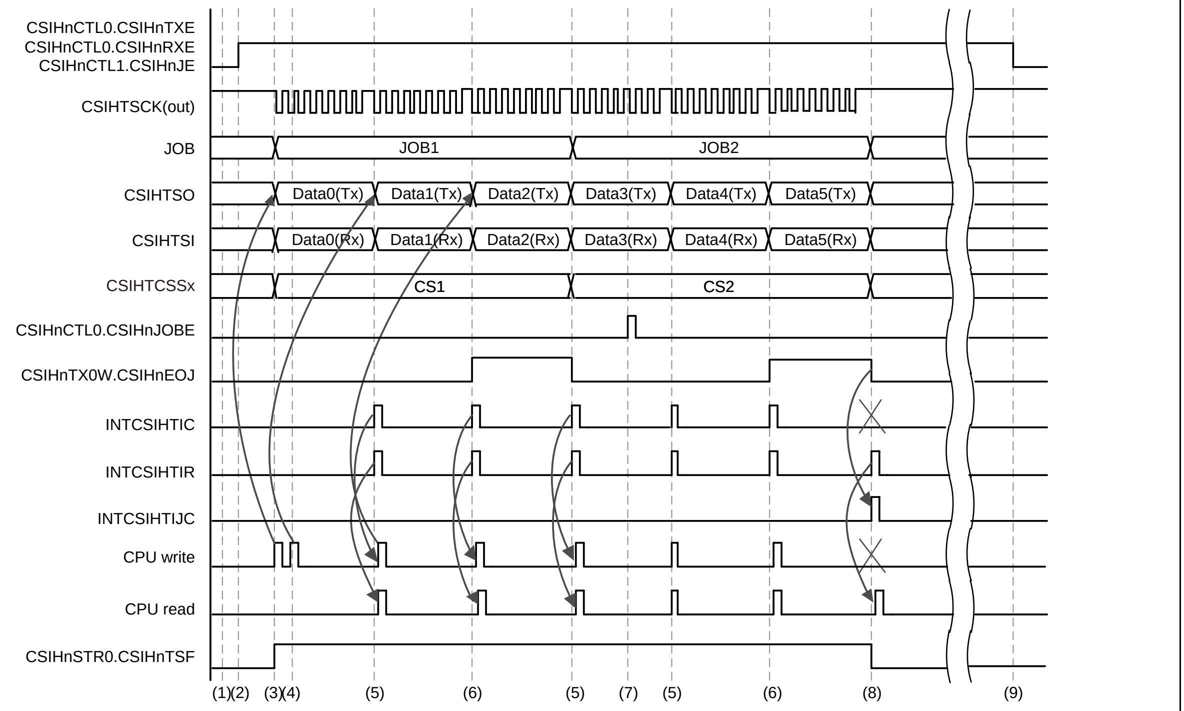


Figure 20.50 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 and CS2. Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits the reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   The CSIHnSTR0.CSIHnTSF flag indicates that communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   - INTCSIHTIR indicates that the reception register, CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of the current job (JOB2).
8. After the forced stop of communication, the interrupt request, INTCSIHTIC is replaced by INTCSIHTIJC. INTCSIHTIR is generated as usual.
   The interrupt request, INTCSIHTIJC indicates that the communication was forcibly stopped at the end of the current job. The interrupt request, INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.
   To start another transmission without stopping communication, perform steps 3 and later.

20.6.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master is provided, one with job mode disabled, and the other one with job mode enabled.

20.6.2.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

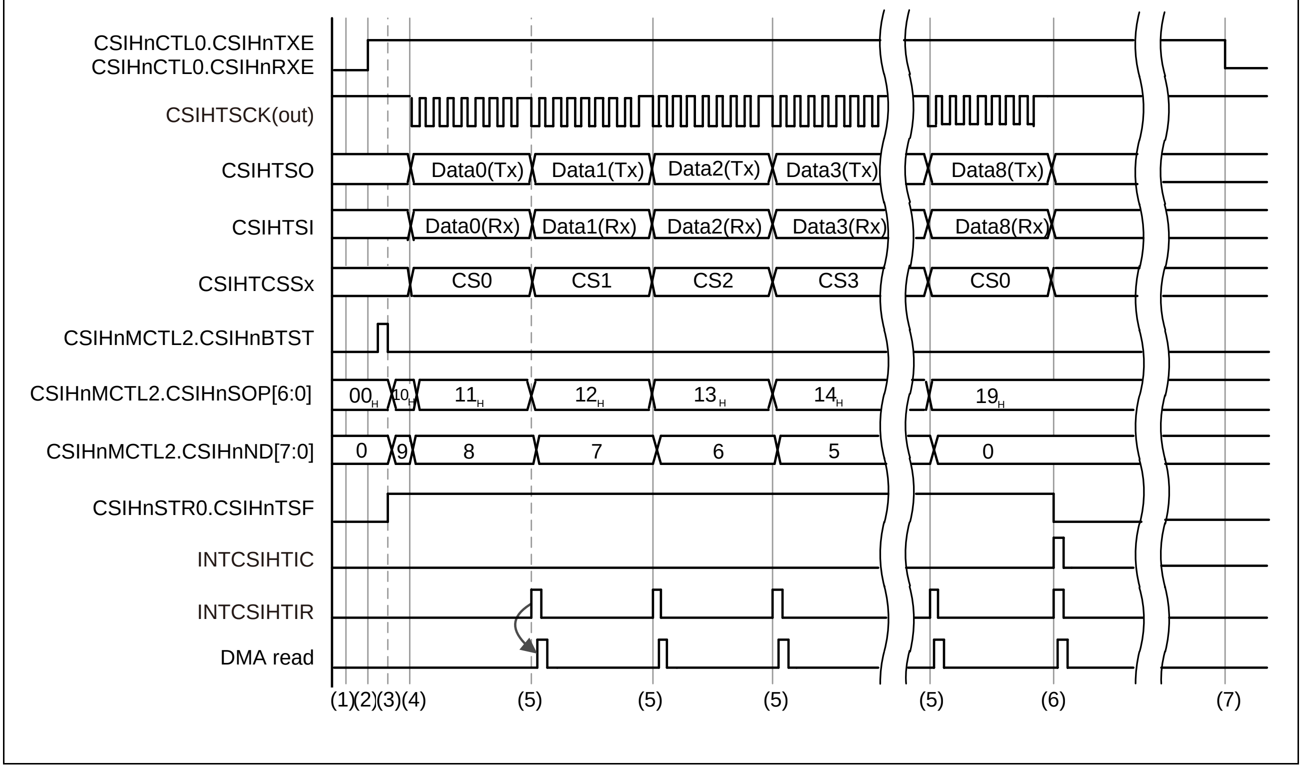


Figure 20.51 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIHnRX0W must be read.
6. When all transmissions are complete, the interrupt request, INTCSIHTIC is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.2.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

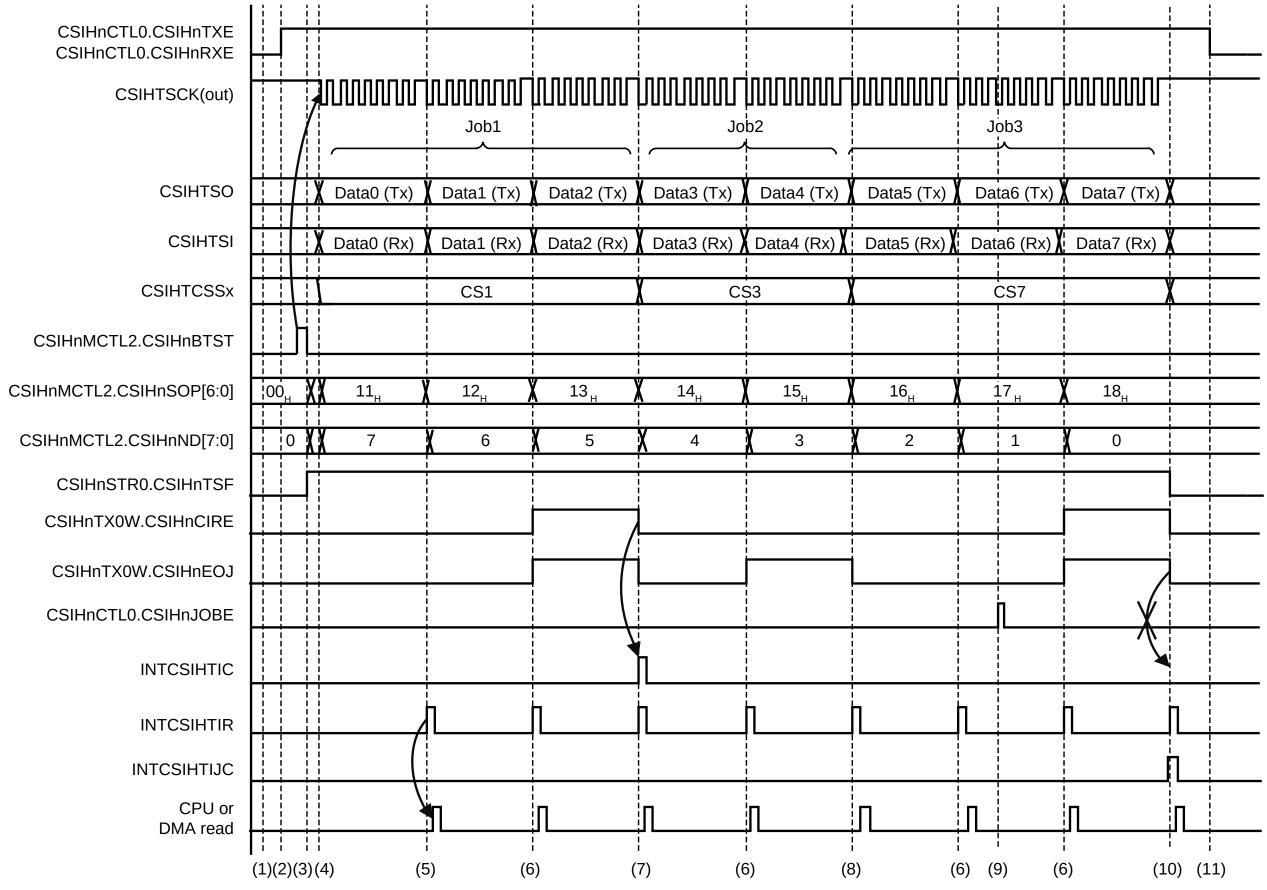


Figure 20.52 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1, CS3, and CS7.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIHnRX0W must be read.
6. The CSIHnTX0W.CSIHnEOJ = 1 setting indicates that the last data of the current job is sent.
7. The interrupt request INTCSIHTIC is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CHABnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
10. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of job3.
    The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.
    The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation.
    When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.3 Procedures in Dual Buffer Mode

Examples when job mode is enabled in master mode, disabled in master mode, and disabled in slave mode are provided below.

20.6.3.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CHABnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

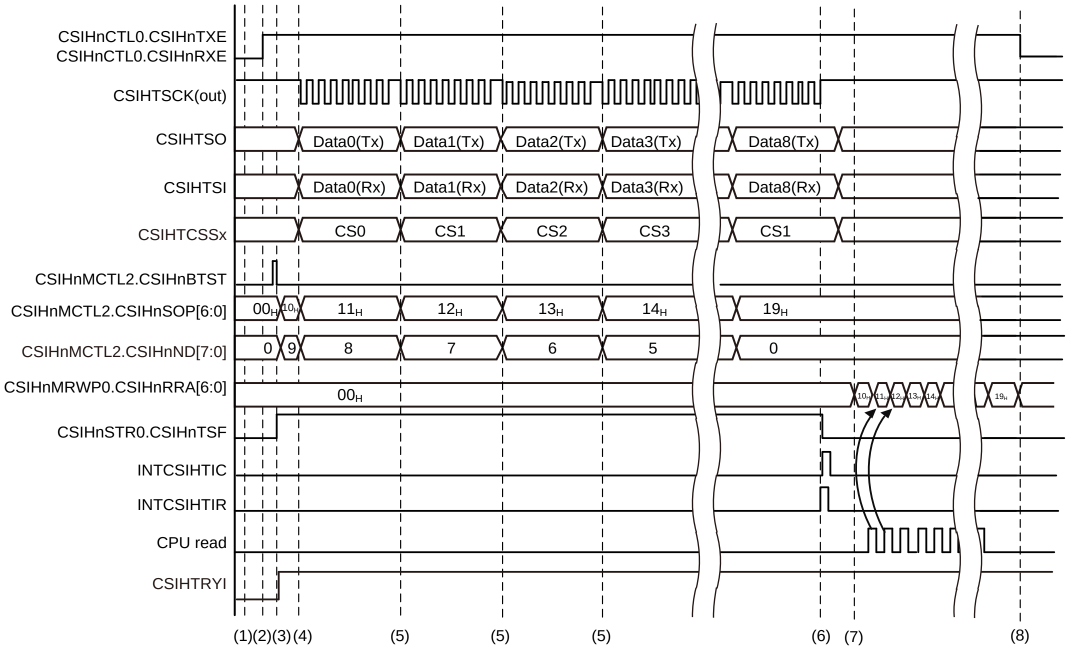


Figure 20.53 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0]. Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Permit buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated.
6. When the last data is transmitted/received, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data from the receive buffer.
7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation.
   When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.3.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

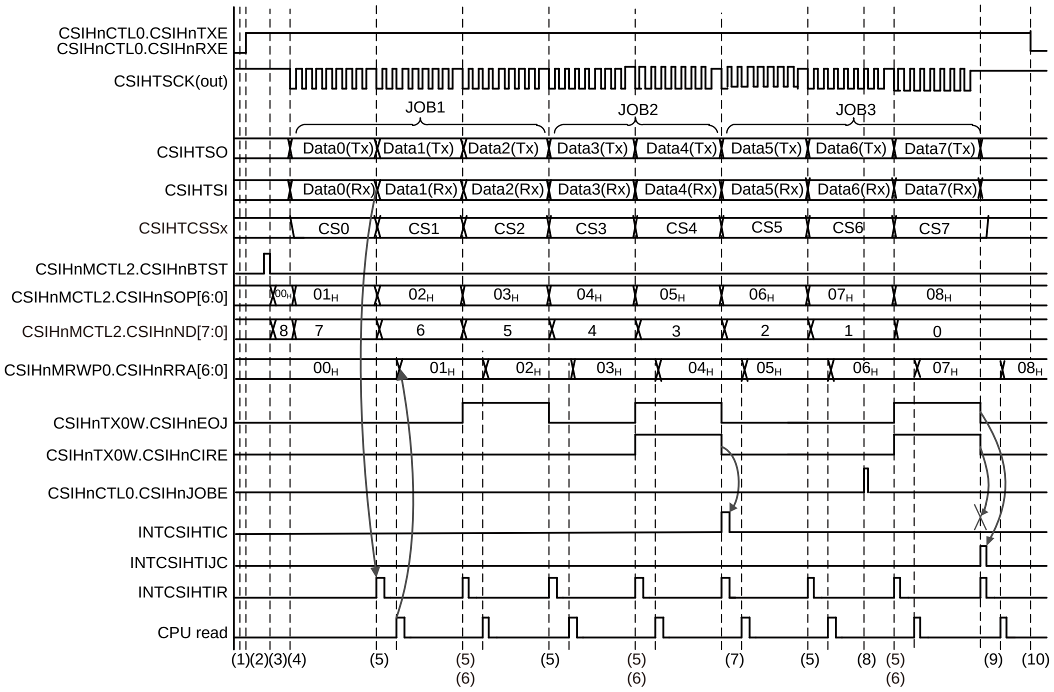


Figure 20.54 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS0 to CS7.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission. This is repeated until the last data is transmitted/received.
5. The INTCSIHTIR interrupt request is generated everytime data is received.
   The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
6. CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent.
7. The INTCSIHTIC interrupt request is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
9. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of JOB3.
   The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.
   The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in register CSIHnTX0W is not sent.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation.
    When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCTL1.CSIHnCKR = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Handshake function is enabled (CSIHnCTL1.CSIHnHSE = 1)

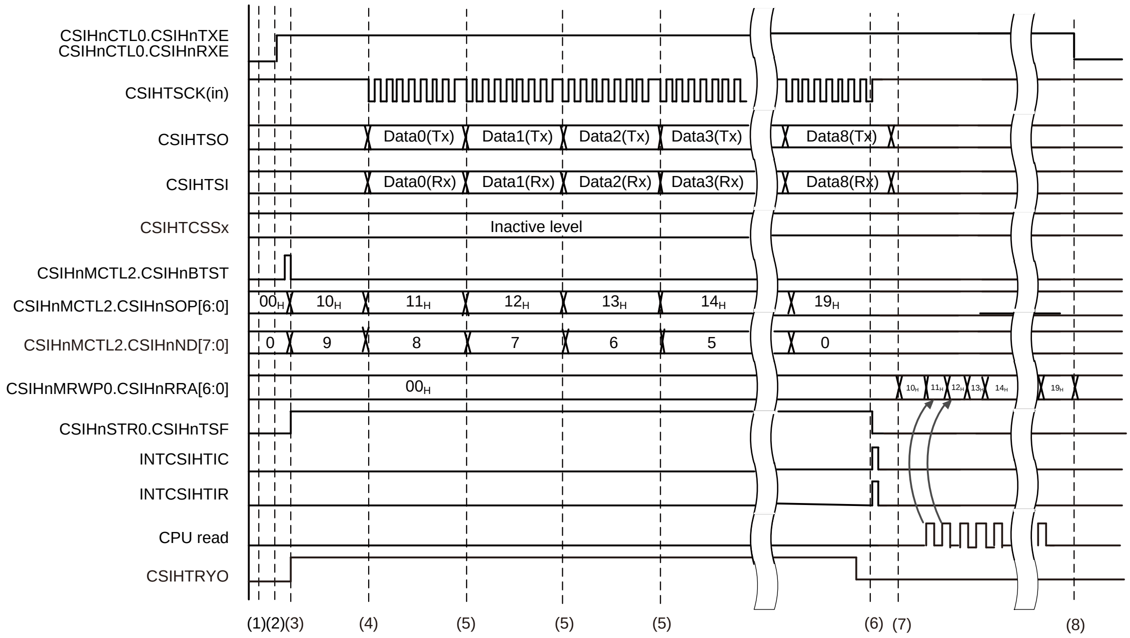


Figure 20.55 Slave in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in register CSIHnCFG0.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Specify the transfer start address by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits. Permit the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock is received from the master. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated because transmission data is sent from the buffer, and received data is stored in the buffer.
6. When the last data is transmitted/received, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data that is stored in the receive buffer.
7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation.
   When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

20.6.4.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

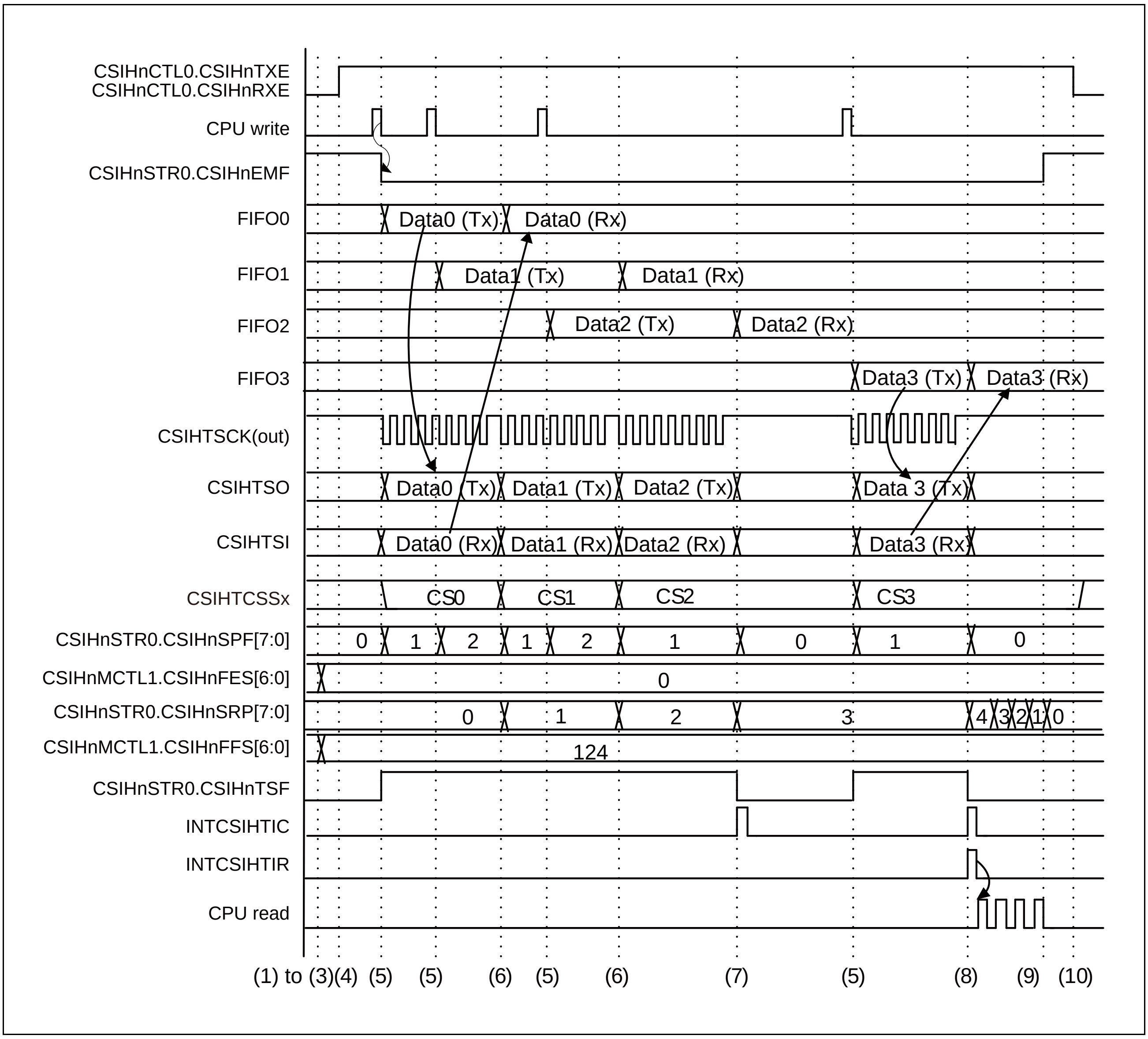


Figure 20.56 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Specify the job mode disable and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2. Specify the FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] = 00_B. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
2. Set CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
   (CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.)
3. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt output. Specify the conditions for generating the INTCSIHTIR interrupt request in the CSIHnMCTL1.CSIHnFFS[6:0] bits.
4. Set CSIHnCTL0.CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
5. Write the first transmit data to the transmit register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   Check that CSIHnSTR0.CSIHnEMF = 0.
6. The current transmission is completed. As the CSIHnFES[6:0] bits are not the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is not generated.
7. As the CSIHnFES[6:0] bits are the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is generated.
8. When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. Since CSIHnFES[6:0] = CSIHnSPF[7:0], the interrupt request INTCSIHTIC is generated.
   After the generation of an interrupt, the CPU starts reading received data that is stored in the receive buffer.
9. When the CPU completes reading the received data that is stored in the receive buffer, CSIHnSTR0.CSIHnEMF is set to 1 and the FIFO buffer becomes empty.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. In addition, if communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

20.6.4.2 Transmit/Receive Mode when Job Mode is Enabled in Master Mode

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- JOB1 consists of four data, JOB2 consists of three data, and JOB3 consists of five data.
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

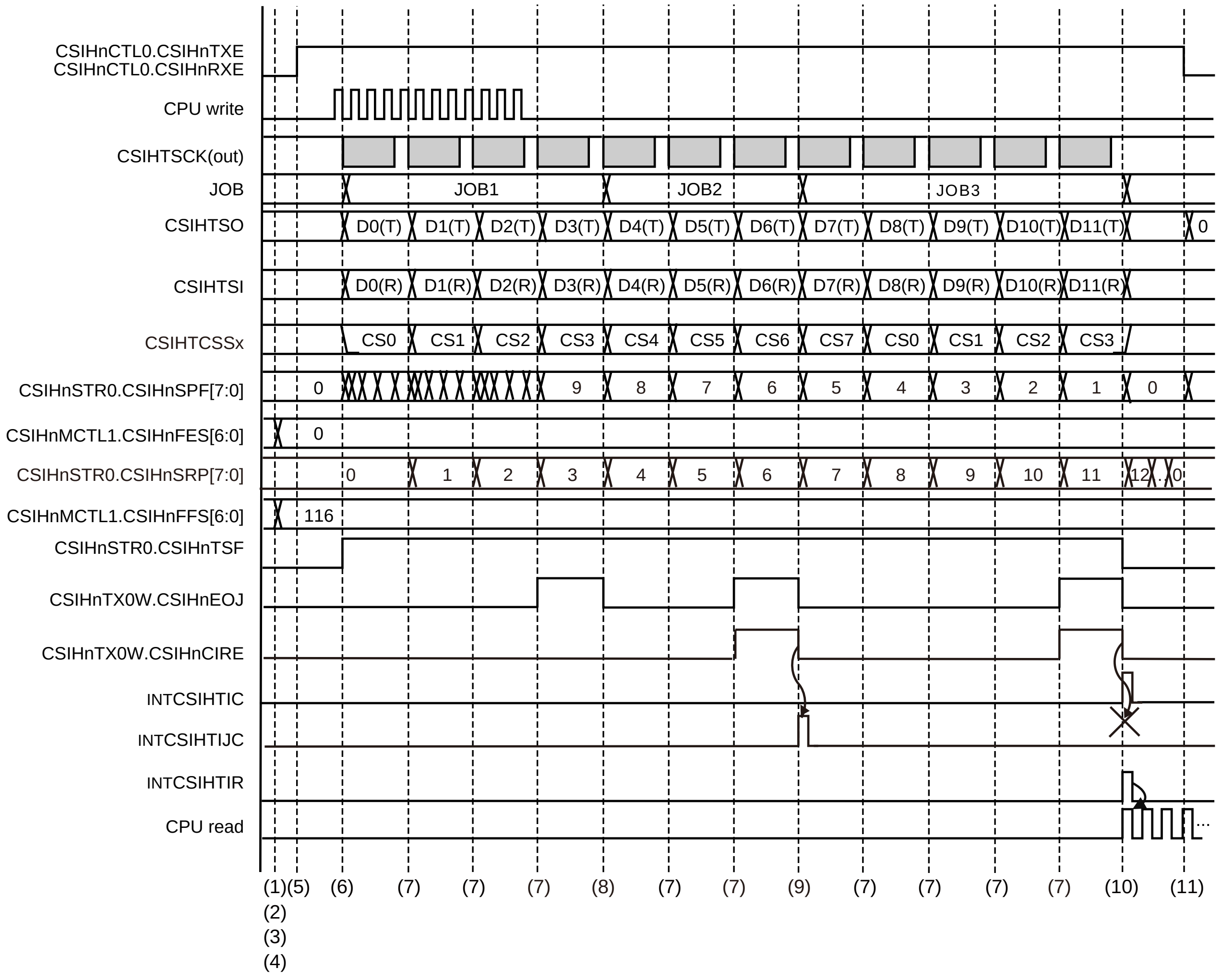


Figure 20.57 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Set job mode disable and master mode in the corresponding bits of the CSIHnCTL1 and CSIHnCTL2 registers. Set FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] to 00_B. This example uses chip select signals CS0 to CS7.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00_H.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt request. With CSIHnMCTL1.CSIHnFFS[6:0], specify the conditions for generating the INTCSIHTIR interrupt request.
5. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the first data to be sent to the CSIHnTX0W transmission register. Transmission starts automatically when the first data becomes available.
   Make sure CSIHnSTR0.CSIHnEMF is set to 0.
7. The current transmission is completed. Since CSIHnFES[6:0] is not the same as CSIHnSPF[7:0], the interrupt request INTCSIHTIC is not generated.
8. The INTCSIHTIJC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. The INTCSIHTIJC interrupt request is generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
10. The INTCSIHTIC interrupt request is generated because CSIHnFES[6:0] = CSIHnSPF[7:0]. INTCSIHTIC is generated so that INTCSIHTIJC is not generated.
    When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. After the generation of the INTCSIHTIR interrupt, CPU starts reading the received data stored in received buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

20.7 Detection and Correction of Errors in CSIHn RAM

For details of ECC, see **Section 40A.2.6, ECC for Peripheral RAM**, **Section 40B.2.6, ECC for Peripheral RAM**, **Section 40C.2.5, ECC for Peripheral RAM**.

Section 21 LIN Master Interface (RLIN2)

This section contains a generic description of the LIN master interface (RLIN2).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RLIN2.

21.1 Features of RH850/F1KH, RH850/F1KM RLIN2

21.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN2 units and channels.

Table 21.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	3	3	4
Name	RLIN24n (n = 0 to 2)	RLIN24n (n = 0 to 2)	RLIN24n (n = 0 to 3)

Table 21.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	2	3	3	3
Name	RLIN24n (n = 0)	RLIN24n (n = 0, 1)	RLIN24n (n = 0 to 2)	RLIN24n (n = 0 to 2)	RLIN24n (n = 0 to 2)

Table 21.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)

Table 21.4 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units (RH850/F1KH-D8)

Unit Name RLIN24n	Number of Channels per Unit	Unit Channel Number (i)	Channel Name RLIN2m	RH850/ F1KH-D8 176 Pins (10 ch)	RH850/ F1KH-D8 233 Pins (12 ch)	RH850/ F1KH-D8 324 Pins (16 ch)
RLIN240	4	0	RLIN20	✓	✓	✓
		1	RLIN21	✓	✓	✓
		2	RLIN22	✓	✓	✓
		3	RLIN23	✓	✓	✓
RLIN241	4	0	RLIN24	✓	✓	✓
		1	RLIN25	✓	✓	✓
		2	RLIN26	✓	✓	✓
		3	RLIN27	✓	✓	✓
RLIN242	4	0	RLIN28	✓	✓	✓
		1	RLIN29	✓	✓	✓
		2	RLIN210	—	✓	✓
		3	RLIN211	—	✓	✓
RLIN243	4	0	RLIN212	—	—	✓
		1	RLIN213	—	—	✓
		2	RLIN214	—	—	✓
		3	RLIN215	—	—	✓

Table 21.5 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units (RH850/F1KM-S4)

Unit Name RLIN24n	Number of Channels per Unit	Unit Channel Number (i)	Channel Name RLIN2m	RH850/ F1KM-S4 100 Pins (3 ch)	RH850/ F1KM-S4 144 Pins (6 ch)	RH850/ F1KM-S4 176 Pins (10 ch)	RH850/ F1KM-S4 233 Pins (12 ch)	RH850/ F1KM-S4 272 Pins (12 ch)
RLIN240	4	0	RLIN20	✓	✓	✓	✓	✓
		1	RLIN21	✓	✓	✓	✓	✓
		2	RLIN22	✓	✓	✓	✓	✓
		3	RLIN23	—	✓	✓	✓	✓
RLIN241	4	0	RLIN24	—	✓	✓	✓	✓
		1	RLIN25	—	✓	✓	✓	✓
		2	RLIN26	—	—	✓	✓	✓
		3	RLIN27	—	—	✓	✓	✓
RLIN242	4	0	RLIN28	—	—	✓	✓	✓
		1	RLIN29	—	—	✓	✓	✓
		2	RLIN210	—	—	—	✓	✓
		3	RLIN211	—	—	—	✓	✓

Table 21.6 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units (RH850/F1KM-S1)

Unit Name RLIN24n	Number of Channels per Unit	Unit Channel Number (i)	Channel Name RLIN2m	RH850/ F1KM-S1 48 Pins (2 ch)	RH850/ F1KM-S1 64 Pins (2 ch)	RH850/ F1KM-S1 80 Pins (2 ch)	RH850/ F1KM-S1 100 Pins (3 ch)
RLIN240	3	0	RLIN20	✓	✓	✓	✓
		1	RLIN21	✓	✓	✓	✓
		2	RLIN22	—	—	—	✓

Table 21.7 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual RLIN2 units are identified by the index “n” (n = 0 to 3).
m	Throughout this section, the individual channels are identified by the index “m” (m = 0 to 15).
i	Throughout this section, the individual channels of units that compose RLIN2 are identified by the index “i” (i = 0 to 3).
b	Throughout this section, the individual data buffers implemented in RLIN2 are identified by the index “b” (b = 1 to 8).

Table 21.8 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual RLIN2 units are identified by the index “n” (n = 0 to 2).
m	Throughout this section, the individual channels are identified by the index “m” (m = 0 to 11).
i	Throughout this section, the individual channels of units that compose RLIN2 are identified by the index “i” (i = 0 to 3).
b	Throughout this section, the individual data buffers implemented in RLIN2 are identified by the index “b” (b = 1 to 8).

Table 21.9 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RLIN2 units are identified by the index “n” (n = 0).
m	Throughout this section, the individual channels are identified by the index “m” (m = 0 to 2).
i	Throughout this section, the individual channels of units that compose RLIN2 are identified by the index “i” (i = 0 to 2).
b	Throughout this section, the individual data buffers implemented in RLIN2 are identified by the index “b” (b = 1 to 8).

For example, RLN24nGLWBR are the LIN wake-up baud rate select registers, which are the global registers of RLIN2. RLN24nmLiMD are the LIN mode registers, which are the channel registers.

The following lists the indices corresponding to each product.

Table 21.10 Indices Correspondence of Each Product (RH850/F1KH-D8)

Indices Correspondence of Each Product		
176 Pins	233 Pins	324 Pins
i = 0 to 3 (RLIN240, 1) i = 0, 1 (RLIN242)	i = 0 to 3 (RLIN240, 1, 2)	i = 0 to 3 (RLIN240, 1, 2, 3)
b = 1 to 8	b = 1 to 8	b = 1 to 8

Table 21.11 Indices Correspondence of Each Product (RH850/F1KM-S4)

Indices Correspondence of Each Product				
100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
i = 0 to 2 (RLIN240)	i = 0 to 3 (RLIN240) i = 0, 1 (RLIN241)	i = 0 to 3 (RLIN240, 1) i = 0, 1 (RLIN242)	i = 0 to 3 (RLIN240, 1, 2)	i = 0 to 3 (RLIN240, 1, 2)
b = 1 to 8	b = 1 to 8	b = 1 to 8	b = 1 to 8	b = 1 to 8

Table 21.12 Indices Correspondence of Each Product (RH850/F1KM-S1)

Indices Correspondence of Each Product			
48 Pins	64 Pins	80 Pins	100 Pins
i = 0, 1 (RLIN240)	i = 0, 1 (RLIN240)	i = 0, 1 (RLIN240)	i = 0 to 2 (RLIN240)
b = 1 to 8	b = 1 to 8	b = 1 to 8	b = 1 to 8

21.1.2 Register Base Addresses

RLIN2 base addresses are listed in the following table.

RLIN2 register addresses are given as offsets from the base addresses.

Table 21.13 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000_H
<RLIN241_base>	FFCE 0080_H
<RLIN242_base>	FFCE 0100_H
<RLIN243_base>	FFCE 0180_H

Table 21.14 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000_H
<RLIN241_base>	FFCE 0080_H
<RLIN242_base>	FFCE 0100_H

Table 21.15 Register Base Address (RH850/F1KM-S1)

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000_H

21.1.3 Clock Supply

The RLIN2 clock supply is shown in the following table.

Table 21.16 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RLIN24n	LIN communication clock sources	CKSCLK_ILIN	Communication clock
	Register access clock	CPUCLK_L, CKSCLK_ILIN	Bus clock

21.1.4 Interrupt Requests

RLIN2 interrupt requests are listed in the following table.

Table 21.17 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number
RLIN240		
INTRLIN20	RLIN20 interrupt	58
INTRLIN21	RLIN21 interrupt	59
INTRLIN22	RLIN22 interrupt	162
INTRLIN23	RLIN23 interrupt	163
RLIN241		
INTRLIN24	RLIN24 interrupt	226
INTRLIN25	RLIN25 interrupt	227
INTRLIN26	RLIN26 interrupt	275
INTRLIN27	RLIN27 interrupt	276
RLIN242		
INTRLIN28	RLIN28 interrupt	285
INTRLIN29	RLIN29 interrupt	286
INTRLIN210	RLIN210 interrupt	324
INTRLIN211	RLIN211 interrupt	325
RLIN243		
INTRLIN212	RLIN212 interrupt	184
INTRLIN213	RLIN213 interrupt	185
INTRLIN214	RLIN214 interrupt	186
INTRLIN215	RLIN215 interrupt	187

Table 21.18 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number
RLIN240		
INTRLIN20	RLIN20 interrupt	58
INTRLIN21	RLIN21 interrupt	59
INTRLIN22	RLIN22 interrupt	162
INTRLIN23	RLIN23 interrupt	163
RLIN241		
INTRLIN24	RLIN24 interrupt	226
INTRLIN25	RLIN25 interrupt	227
INTRLIN26	RLIN26 interrupt	275
INTRLIN27	RLIN27 interrupt	276
RLIN242		
INTRLIN28	RLIN28 interrupt	285
INTRLIN29	RLIN29 interrupt	286
INTRLIN210	RLIN210 interrupt	324
INTRLIN211	RLIN211 interrupt	325

Table 21.19 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number
RLIN240		
INTRLIN20	RLIN20 interrupt	58
INTRLIN21	RLIN21 interrupt	59
INTRLIN22	RLIN22 interrupt	162

21.1.5 Reset Sources

RLIN2 reset sources are listed in the following table. RLIN2 is initialized by these reset sources.

Table 21.20 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RLIN24n	All reset sources (ISORES)

21.1.6 External Input/Output Signals

External input/output signals of RLIN2 are listed in the following table.

Table 21.21 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN240		
RLIN2mRX (m = 0 to 3)	RLIN240 receive data input	RLIN2mRX (m = 0 to 3)
RLIN2mTX (m = 0 to 3)	RLIN240 transmit data output	RLIN2mTX (m = 0 to 3)
RLIN241		
RLIN2mRX (m = 4 to 7)	RLIN241 receive data input	RLIN2mRX (m = 4 to 7)
RLIN2mTX (m = 4 to 7)	RLIN241 transmit data output	RLIN2mTX (m = 4 to 7)
RLIN242		
RLIN2mRX (m = 8 to 11)	RLIN242 receive data input	RLIN2mRX (m = 8 to 11)
RLIN2mTX (m = 8 to 11)	RLIN242 transmit data output	RLIN2mTX (m = 8 to 11)
RLIN243		
RLIN2mRX (m = 12 to 15)	RLIN243 receive data input	RLIN2mRX (m = 12 to 15)
RLIN2mTX (m = 12 to 15)	RLIN243 transmit data output	RLIN2mTX (m = 12 to 15)

Table 21.22 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN240		
RLIN2mRX (m = 0 to 3)	RLIN240 receive data input	RLIN2mRX (m = 0 to 3)
RLIN2mTX (m = 0 to 3)	RLIN240 transmit data output	RLIN2mTX (m = 0 to 3)
RLIN241		
RLIN2mRX (m = 4 to 7)	RLIN241 receive data input	RLIN2mRX (m = 4 to 7)
RLIN2mTX (m = 4 to 7)	RLIN241 transmit data output	RLIN2mTX (m = 4 to 7)
RLIN242		
RLIN2mRX (m = 8 to 11)	RLIN242 receive data input	RLIN2mRX (m = 8 to 11)
RLIN2mTX (m = 8 to 11)	RLIN242 transmit data output	RLIN2mTX (m = 8 to 11)

Table 21.23 External Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN240		
RLIN2mRX (m = 0 to 2)	RLIN240 receive data input	RLIN2mRX (m = 0 to 2)
RLIN2mTX (m = 0 to 2)	RLIN240 transmit data output	RLIN2mTX (m = 0 to 2)

21.2 Overview

21.2.1 Functional Overview

The LIN Master Interface is a hardware LIN communication controller that complies with LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005), and automatically performs frame communication and error determination.

Table 21.24, LIN Master Interface Specifications shows the LIN Master Interface specifications.

Table 21.24 LIN Master Interface Specifications

Item		Specifications
Channel count		16 channels (In this product, 4-channel version of RLIN2 is included.)
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005)
	Variable frame structure	• Transmission break width: 13 to 28 Tbits • Transmission break delimiter width: 1 to 4 Tbits • Inter-byte space (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Response space: 0 to 7 Tbits*1 • Inter-byte space: 0 to 3 Tbits (space between data bytes in response area) • Transmit wake-up: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)
	Response field data byte count	Variable from 0 to 8 bytes
	Frame communication modes	• Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
	Wake-up transmission and reception	Available in LIN wake-up mode • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured
	Status	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
	Error status	• Bit error • Checksum error • Frame timeout error • Physical bus error • Framing error
	Baud rate selection	Baud rate conforming to the LIN specifications generated using baud rate generator
	Test mode	Self-test mode for user evaluation
	Interrupt function	• Successful frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection The logical OR of these three events is the interrupt source (INTRLIN2m) for each channel.

Note 1. Since the same register is used for configuration, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

21.2.2 Block Diagram

Figure 21.1, LIN Master Interface Block Diagram (324 Pins, RLIN2 16 Channels Embedded) shows a block diagram of the LIN master interface.

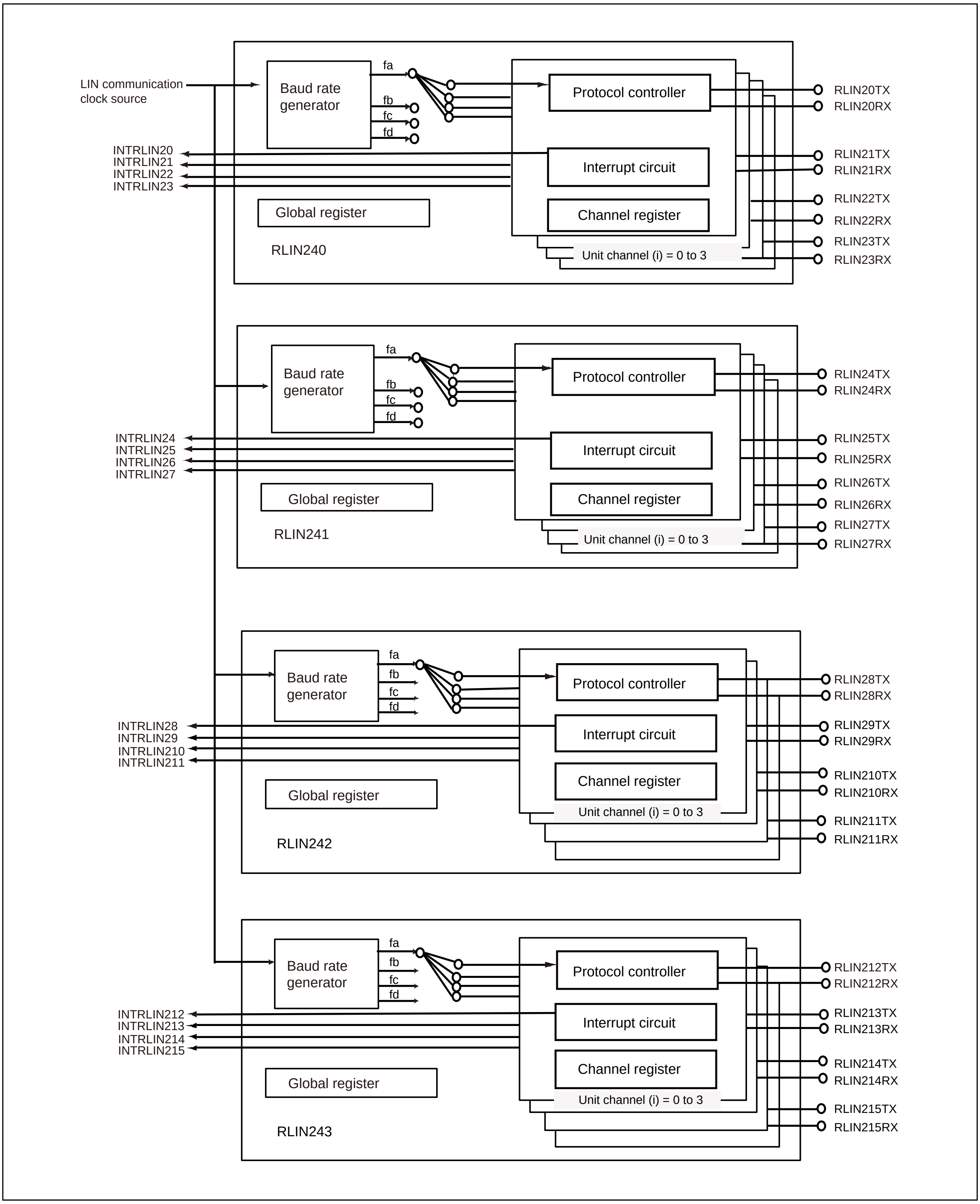


Figure 21.1 LIN Master Interface Block Diagram (324 Pins, RLIN2 16 Channels Embedded)

21.3 Registers

The registers of the LIN master interface include global registers and channel registers. As the global registers are allocated for each unit, they can be individually set for each unit. As the channel registers are allocated for each channel, they can individually control each channel.

21.3.1 List of Registers

RLIN2 registers are listed in the following table.

For details about <RLIN24n_base>, see **Section 21.1.2, Register Base Addresses**.

Table 21.25 List of Registers

Module Name	Register Name	Symbol	Address
Global registers			
RLN24n	LIN wake-up baud rate select register	RLN24nGLWBR	<RLIN24n_base> + 01_H
	LIN baud rate prescaler 0 register	RLN24nGLBRP0	<RLIN24n_base> + 02_H
	LIN baud rate prescaler 1 register	RLN24nGLBRP1	<RLIN24n_base> + 03_H
	LIN self-test control register	RLN24nGLSTC	<RLIN24n_base> + 04_H
Channel registers			
RLN24nm	LIN mode register	RLN24nmLiMD	<RLIN24n_base> + $08_H + i \times 20_H$
	LIN break field configuration register	RLN24nmLiBFC	<RLIN24n_base> + $09_H + i \times 20_H$
	LIN space configuration register	RLN24nmLiSC	<RLIN24n_base> + $0A_H + i \times 20_H$
	LIN wake-up configuration register	RLN24nmLiWUP	<RLIN24n_base> + $0B_H + i \times 20_H$
	LIN interrupt enable register	RLN24nmLiIE	<RLIN24n_base> + $0C_H + i \times 20_H$
	LIN error detection enable register	RLN24nmLiEDE	<RLIN24n_base> + $0D_H + i \times 20_H$
	LIN control register	RLN24nmLiCUC	<RLIN24n_base> + $0E_H + i \times 20_H$
	LIN transmission control register	RLN24nmLiTRC	<RLIN24n_base> + $10_H + i \times 20_H$
	LIN mode status register	RLN24nmLiMST	<RLIN24n_base> + $11_H + i \times 20_H$
	LIN status register	RLN24nmLiST	<RLIN24n_base> + $12_H + i \times 20_H$
	LIN error status register	RLN24nmLiEST	<RLIN24n_base> + $13_H + i \times 20_H$
	LIN data field configuration register	RLN24nmLiDFC	<RLIN24n_base> + $14_H + i \times 20_H$
	LIN ID buffer register	RLN24nmLiIDB	<RLIN24n_base> + $15_H + i \times 20_H$
	LIN checksum buffer register	RLN24nmLiCBR	<RLIN24n_base> + $16_H + i \times 20_H$
	LIN data buffer 1 register	RLN24nmLiDBR1	<RLIN24n_base> + $18_H + i \times 20_H$
	LIN data buffer 2 register	RLN24nmLiDBR2	<RLIN24n_base> + $19_H + i \times 20_H$
	LIN data buffer 3 register	RLN24nmLiDBR3	<RLIN24n_base> + $1A_H + i \times 20_H$
	LIN data buffer 4 register	RLN24nmLiDBR4	<RLIN24n_base> + $1B_H + i \times 20_H$
	LIN data buffer 5 register	RLN24nmLiDBR5	<RLIN24n_base> + $1C_H + i \times 20_H$
	LIN data buffer 6 register	RLN24nmLiDBR6	<RLIN24n_base> + $1D_H + i \times 20_H$
	LIN data buffer 7 register	RLN24nmLiDBR7	<RLIN24n_base> + $1E_H + i \times 20_H$
	LIN data buffer 8 register	RLN24nmLiDBR8	<RLIN24n_base> + $1F_H + i \times 20_H$

Note: When writing to a register not used, write the value after reset.

21.3.2 Global Registers

21.3.2.1 RLN24nGLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLWBR: <RLIN24n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 21.26 RLN24nGLWBR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified by the LCKS bit setting in the RLN24nmLiMD registers is used (when LIN 1.3 is used). 1: In LIN wake-up mode, the clock fa is used regardless of the setting of the LCKS bit in the RLN24nmLiMD registers (when LIN 2.X is used).

Set the RLN24nGLWBR register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

LWBR0 Bit (Wake-Up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR register to 0. This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.

When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. With this setting, fa is selected as the LIN system clock (f_{LIN}) during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register (the LCKS bit is not changed) and the 2.5-Tbit or longer low-level width of the input signal can be measured. Setting the baud rate to 19200 bps while fa is selected allows the 130 µs or longer low-level width of the input signal to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register.

21.3.2.2 RLN24nGLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLBRP0: <RLIN24n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.27 RLN24nGLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the LIN communication clock source by N+1. Setting range: 00_H to FF_H

Set the RLN24nGLBRP0 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the LIN communication clock source by N+1.

21.3.2.3 RLN24nGLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLBRP1: <RLIN24n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.28 RLN24nGLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the LIN communication clock source by M+1. Setting range: 00_H to FF_H

Set the RLN24nGLBRP1 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the LIN communication clock source by M+1.

21.3.2.4 RLN24nGLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLSTC: <RLIN24n_base> + 04$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.29 RLN24nGLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing A7$_H$, 58$_H$, and 01$_H$ successively to the RLN24nGLSTC register places the module in LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN24nGLSTC register cancels protection of LIN self-test mode.

Set the RLN24nGLSTC register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

Writing A7$_H$, 58$_H$, and 01$_H$ successively to the RLN24nGLSTC register places the module into LIN self-test mode.

When successive writing is completed and the module is placed in LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For details about transitioning to LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000$_B$”, and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For details about exiting LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN24nGLSTC register if it is not a part of successive writing of A7$_H$, 58$_H$, and 01$_H$.

21.3.3 Channel Registers

21.3.3.1 RLN24nmLiMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiMD: <RLIN24n_base> + 08_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	LCKS[1:0]		—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R

Table 21.30 RLN24nmLiMD Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiMD register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When the LWBR0 bit in the RLN24nGLWBR is 1 (when LIN 2.x is used) and the RLN24nmLiMST register is 01_H (LIN wake-up mode), regardless of the setting of the LCKS bit, fa is input to the protocol controller (LCKS bit is not changed).

21.3.3.2 RLN24nmLiBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiBFC: <RLIN24n_base> + 09_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.31 RLN24nmLiBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN24nmLiBFC register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.
1 Tbit to 4 Tbits can be specified.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.
13 Tbits to 28 Tbits can be specified.

21.3.3.3 RLN24nmLiSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiSC: <RLIN24n_base> + $0A_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 21.32 RLN24nmLiSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN24nmLiSC register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be specified.

These bits are enabled only during response transmission; they are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.

0 Tbit to 7 Tbits can be specified.

The response space setting is enabled only during response transmission; the setting is disabled during response reception.

The inter-byte space (header) value is equal to the response space value.

21.3.3.4 RLN24nmLiWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiWUP: <RLIN24n_base> + $0B_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 21.33 RLN24nmLiWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiWUP register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

WUTL[3:0] Bits (Wake-Up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be specified.

When the LWBR0 bit in the RLN24nGLWBR is 1 (when LIN 2.x is used), regardless of the setting of the LCKS bit in the RLN24nmLiMD register, fa is selected as the LIN system clock (f_{LIN}) (LCKS bit is not changed).

21.3.3.5 RLN24nmLiIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiIE: <RLIN24n_base> + $0C_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 21.34 RLN24nmLiIE Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN24nmLiIE register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables interrupt request upon detection of an error.

With 0 set, the interrupt request is not generated when the ERR flag in the RLN24nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the ERR flag in the RLN24nmLiST register is set to 1.

Errors that constitute interrupt sources are bit errors, physical bus errors, frame timeout errors, framing errors, and checksum errors.

Detection of a bit error, physical bus error, frame timeout error, and framing error can be enabled or disabled using the RLN24nmLiEDE register.

FRCIE Bit (Successful Frame/Wake-Up Reception Interrupt Request Enable)

The FRCIE bit enables or disables interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).

With 0 set, the interrupt request is not generated when the FRC flag in the RLN24nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the FRC flag in the RLN24nmLiST register is set to 1.

FTCIE Bit (Successful Frame/Wake-Up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request is not generated when the FTC flag in the RLN24nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the FTC flag in the RLN24nmLiST register is set to 1.

21.3.3.6 RLN24nmLiEDE — LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiEDE: <RLIN24n_base> + $0D_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.35 RLN24nmLiEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Frame Timeout Error Detection Enable 0: Disables frame timeout error detection. 1: Enables frame timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN24nmLiEDE register while the OMM0 bit in the RLN24nmLiMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN24nmLiEST register.

For details on the framing error, see **Section 21.14, Error Status**.

FTERE Bit (Frame Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error.

With 0 set, the frame timeout error is not detected.

With 1 set, the frame timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN24nmLiEST register.

For details on the frame timeout error, see **Section 21.14, Error Status**.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.

With 0 set, the physical bus error is not detected.

With 1 set, the physical bus error is detected.

When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN24nmLiEST register.

For details on the physical bus error, see **Section 21.14, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLN24nmLiEST register.

For details on the bit error, see **Section 21.14, Error Status**.

21.3.3.7 RLN24nmLiCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiCUC: <RLIN24n_base> + $0E_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.36 RLN24nmLiCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: Transition to LIN reset mode. 1: Exit LIN reset mode.

Set the RLN24nmLiCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN24nmLiCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN24nmLiMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the operating mode (LIN wake-up mode or LIN operation mode) that is entered after exiting LIN reset mode.

Setting this bit to 0 selects LIN wake-up mode.

Setting this bit to 1 selects LIN operation mode.

This bit is enabled only when the OMM0 bit in the RLN24nmLiMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN24nmLiTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.

Setting this bit to 0 causes RLIN2 to enter LIN reset mode.

Setting this bit to 1 causes RLIN2 to exit LIN reset mode.

21.3.3.8 RLN24nmLiTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiTRC: <RLIN24n_base> + 10_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.37 RLN24nmLiTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission Start 0: Response transmission is stopped in frame separate mode. 1: Response transmission is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission/Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

RTS Bit (Response Transmission Start)

Set the RTS bit (response transmit start bit) to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame transmission and transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H to the RLN24nmLiTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode). When the OMM1 bit is 0 (LIN wake-up mode), do not write 1.

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is stopped).

FTS Bit (Frame Transmission/Wake-Up Transmission/Reception Start)

Set the FTS bit to 1 to start frame/wake-up transmission.

Also set this bit to 1 to allow wake-up reception (counting of the low level width of the input signal).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

21.3.3.9 RLN24nmLiMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: RLN24nmLiMST: <RLIN24n_base> + 11_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.38 RLN24nmLiMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

When the OMM0 bit is 0_B (LIN reset mode), the value of this bit is disabled.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

21.3.3.10 RLN24nmLiST — LIN Status Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiST: <RLIN24n_base> + 12_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 21.39 RLN24nmLiST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN24nmLiST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

Writing to this register is prohibited while the FTS bit in the RLN24nmLiTRC register is 1 (frame transmission or wake-up transmission/reception is started)

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value from before 1 was written.

"1" is set upon completion of header transmission, but an interrupt request is not generated.

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value from before 1 is written.

“1” is set upon completion of Data 1 reception, but an interrupt request is not generated.

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when the value of any of the flags of the RLN24nmLiEST registers is 1). Here, an interrupt request is generated if the ERRIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST register while in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Successful Frame/Wake-Up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value from before 1 is written.

The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request is generated if the FRCIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled).

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

FTC Flag (Successful Frame/Wake-Up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value from before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request is generated if the FTCIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled).

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

21.3.3.11 RLN24nmLiEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiEST: <RLIN24n_base> + 13_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 21.40 RLN24nmLiEST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Frame Timeout Error Flag 0: Frame timeout error has not been detected. 1: Frame timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN24nmLiEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN24nmLiTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value from before 1 is written.

The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value from before 1 is written.

The FER flag is set to 1 upon framing error detection when the FERE bit of the RLN24nmLiEST register is 1 (framing error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

FTER Flag (Frame Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value from before 1 is written.

The FTER flag is set to 1 upon frame timeout error detection when the FTERE bit of the RLN24nmLiEDE register is 1 (frame timeout error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value from before 1 is written.

The PBER flag is set to 1 upon physical bus error detection when the PBERE bit of the RLN24nmLiEDE register is 1 (physical bus error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value from before 1 is written.

The BER flag is set to 1 upon bit error detection when the BERE bit of the RLN24nmLiEDE register is 1 (bit error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

21.3.3.12 RLN24nmLiDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiDFC: <RLIN24n_base> + 14_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.41 RLN24nmLiDFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 bytes + checksum 0 0 0 1: 1 byte + checksum 0 0 1 0: 2 bytes + checksum : 0 1 1 1: 7 bytes + checksum 1 0 0 0: 8 bytes + checksum Settings other than the above are prohibited.

Set the RLN24nmLiDFC register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response transmission mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN24nmLiTRC register is 1), response is transmitted/received without the RTS bit in the RLN24nmLiTRC register being set.

With 1 set, frame separate mode is selected. When the RTS bit of the RLN24nmLiTRC register is set to 1 during header transmission, response transmission is executed after header transmission has ended.

For response reception (the RFT bit is 0), set the FSM bit to 0.

When transitioning to LIN self-test mode, set this bit to 0 before transition.

For details on frame separate mode, see **Section 21.11.1, Transmission of LIN Frames**.

CSM Bit (Checksum Select)

The CSM bit sets checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When frame timeout error detection is enabled (the FTERE bit in the RLN24nmLiEDE register is 1), the timeout time depends on the setting of this bit. For details, see **Section 21.14, Error Status**.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low level width of the input signal is counted).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

21.3.3.13 RLN24nmLiIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiIDB: <RLIN24n_base> + 15_H + i × 20_H

Value after reset: Undefined

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.42 RLN24nmLiIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bit (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bit (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN24nmLiIDB register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted prior to communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 sets P0 and IDP1 sets P1.

Since parity is not automatically calculated, set the calculation value. Note that even if the specified calculation result is incorrect, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

21.3.3.14 RLN24nmLiCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

Address: RLN24nmLiCBR: <RLIN24n_base> + 16_H + i × 20_H

Value after reset: Undefined

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.43 RLN24nmLiCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

while in LIN operation mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC register is 1 (transmission):
  After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.
  Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

Set the RLN24nmLiCBR register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

21.3.3.15 RLN24nmLiDBRb — LIN Data Buffer b Register

Access: This register can be read or written in 8-bit units.

Address:
RLN24nmLiDBR1: <RLIN24n_base> + 18_H + i × 20_H
RLN24nmLiDBR2: <RLIN24n_base> + 19_H + i × 20_H
RLN24nmLiDBR3: <RLIN24n_base> + $1A_H$ + i × 20_H
RLN24nmLiDBR4: <RLIN24n_base> + $1B_H$ + i × 20_H
RLN24nmLiDBR5: <RLIN24n_base> + $1C_H$ + i × 20_H
RLN24nmLiDBR6: <RLIN24n_base> + $1D_H$ + i × 20_H
RLN24nmLiDBR7: <RLIN24n_base> + $1E_H$ + i × 20_H
RLN24nmLiDBR8: <RLIN24n_base> + $1F_H$ + i × 20_H

Value after reset: Undefined

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.44 RLN24nmLiDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting range: 00_H to FF_H

- For response transmission:

  The LiDBRb registers specify the data to be transmitted in the response field.
  Configure these registers when:

  - The RFT bit in RLN24nmLiDFC register is 1 (transmission).
  - The FSM bit in RLN24nmLiDFC register is 0 (not frame separate mode).
  - The FTS bit in RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

  or

  - The RFT bit in RLN24nmLiDFC register is 1 (transmission).
  - The FSM bit in RLN24nmLiDFC register is 1 (frame separate mode).
  - The RTS bit in RLN24nmLiTRC register is 0 (response transmission is stopped).

- For response reception:

  The LiDBRb registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started).

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted prior to communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

21.4 Interrupt Sources

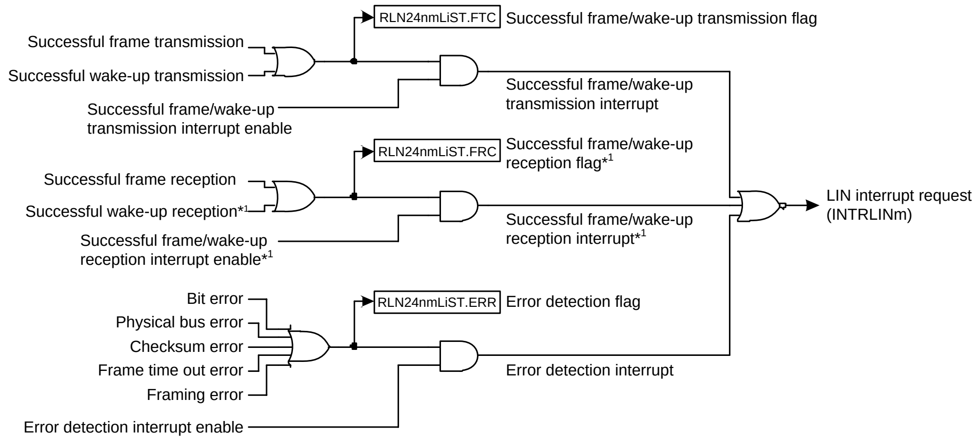
The LIN interrupts are interrupt requests generated by the LIN master interface.

There are three interrupt sources for each channel; successful frame/wake-up transmission, successful frame/wakeup reception, and error detection.

Interrupt requests from these three sources are ORed to generate one interrupt request “LIN interrupt”.

The respective interrupt request is output when the corresponding flag in the RLN24nmLiST register is set to 1 while the corresponding bit in the RLN24nmLiIE register is 1 (interrupt enabled). However, if an interrupt is requested when the corresponding flag in the RLN24nmLiST register has been set to 1, it is ignored. Therefore, clear the corresponding flag to 0 to enable the interrupt again.

Figure 21.2, LIN Interrupt Block Diagram shows a block diagram of the LIN interrupt.



Note 1. The wake-up receive signal indicates the input signal low level width count.

Figure 21.2 LIN Interrupt Block Diagram

21.5 Modes

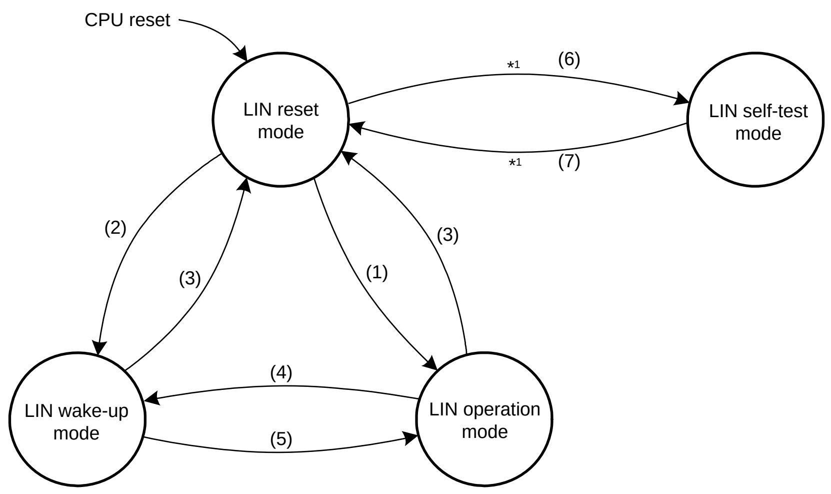
The LIN master interface provides the following four modes:

- LIN reset mode
- LIN operation mode
- LIN wake-up mode
- LIN self-test mode

The mode transitions except LIN self-test mode is controlled independently for respective channels.

Figure 21.3, Mode Transitions shows mode transitions. **Table 21.45, Transition Condition of Each Mode** describes mode transition conditions.

Table 21.46, Operations Available in Each Mode lists operations available in each mode.



Note 1. See **Section 21.15, LIN Self-Test Mode** for the details.

Figure 21.3 Mode Transitions

Table 21.45 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN operation mode	RLN24nmLiCUC.OM1,OM0 = 11_B
(2)	LIN reset mode	→	LIN wake-up mode	RLN24nmLiCUC.OM1,OM0 = 01_B
(3)	LIN wake-up mode LIN operation mode	→	LIN reset mode	RLN24nmLiCUC.OM0 = 0_B
(4)	LIN operation mode	→	LIN wake-up mode	RLN24nmLiCUC.OM1,OM0 = 01_B
(5)	LIN wake-up mode	→	LIN operation mode	RLN24nmLiCUC.OM1,OM0 = 11_B
(6)	LIN reset mode	→	LIN self-test mode	See **Section 21.15, LIN Self-Test Mode**.
(7)	LIN self-test mode	→	LIN reset mode	See **Section 21.15, LIN Self-Test Mode**.

Table 21.46 Operations Available in Each Mode

LIN Operation Mode	LIN Wake-Up Mode	LIN Self-Test Mode
Header transmission Response transmission Response reception Error detection	Wake-up transmission Wake-up reception Error detection	self-test

Whether a transition has been made to LIN reset mode, LIN operation mode, or LIN wake-up mode can be verified by reading the OMM1 and OMM0 bits in the RLN24nmLiMST register.

For a description of LIN self-test mode, see **Section 21.15, LIN Self-Test Mode**.

21.6 LIN Reset Mode

Setting the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The transition to LIN reset mode can be verified by determining that the OMM0 bit in the RLN24nmLiMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication stops.

From LIN reset mode, transitions to LIN operation mode, LIN wake-up mode, and LIN self-test mode can be made.

When the mode transitions to LIN reset mode, the following registers are initialized to their reset values and retain their initial values while in LIN reset mode:

- RLN24nmLiTRC register
- RLN24nmLiST register
- RLN24nmLiEST register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN24nGLWBR register
- RLN24nGLBRP0 register
- RLN24nGLBRP1 register
- RLN24nmLiMD register
- RLN24nmLiBFC register
- RLN24nmLiSC register
- RLN24nmLiWUP register
- RLN24nmLiIE register
- RLN24nmLiEDE register
- RLN24nmLiDFC register
- RLN24nmLiIDB register
- RLN24nmLiCBR register
- RLN24nmLiDBRb register

21.7 LIN Operation Mode

While in LIN operation mode, frame processing (header transmission, response transmission, response reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST register to 11_B. Communication settings should be performed after the RLN24nmLiMST register has become 11_B.

21.8 LIN Wake-Up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST register to 01_B. Communication settings should be performed after the RLN24nmLiMST register has become 01_B.

21.9 Header Transmission/Response Transmission/Response Reception

21.9.1 Header Transmission

Figure 21.4, Operation in Header Transmission shows the operation of the LIN master interface in header transmission. **Table 21.47, Processing in Header Transmission** shows processing in header transmission.

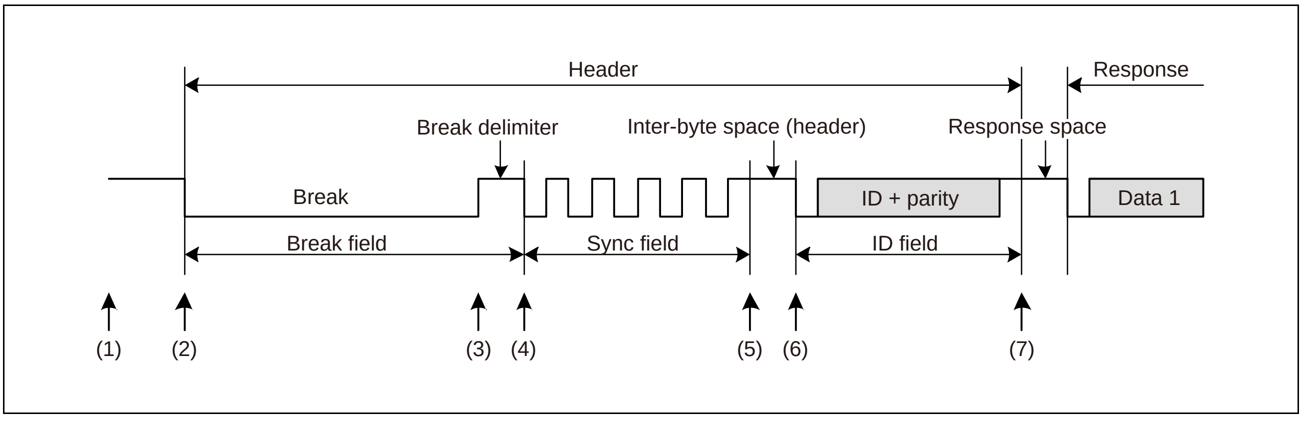


Figure 21.4 Operation in Header Transmission

Table 21.47 Processing in Header Transmission

	Software Processing	LIN Master Interface Processing
(1)	● Sets a baud rate ● Enables interrupts ● Enables error detection ● Sets frame configuration parameters ● Transitions to LIN operation mode ● Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data)	Waits for the FTS bit in the RLN24nmLiTRC register to be set by software (idle).
(2)	Sets the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55_H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets the successful header transmission flag.

NOTE

For information about error detection, see **Section 21.14, Error Status**.

21.9.2 Response Transmission

Figure 21.5, Operation in Response Transmission shows the operation of the LIN master interface in response transmission. **Table 21.48, Processing in Response Transmission** shows the processing in response transmission.

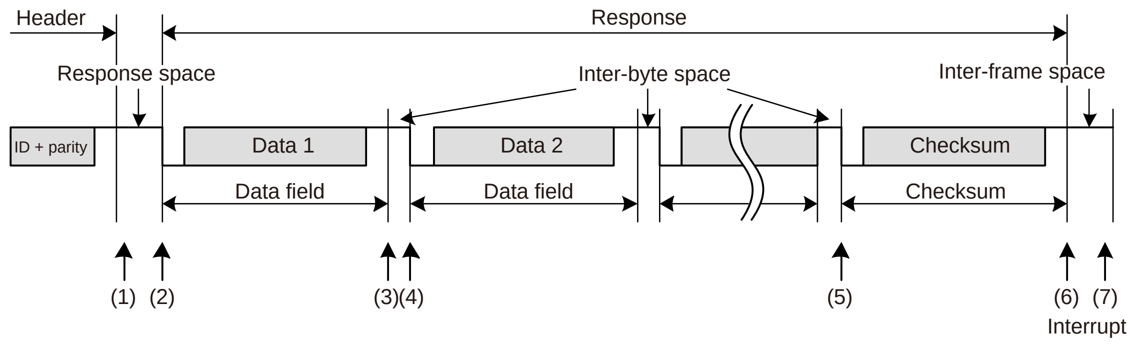


Figure 21.5 Operation in Response Transmission

Table 21.48 Processing in Response Transmission

	Software Processing	LIN Master Interface Processing
(1)	(When in frame separate mode) ● Sets the RTS bit in the RLN24nmLiTRC register to 1 (response transmission started) (When not in frame separate mode) ● Waits for an interrupt request	(When in frame separate mode) ● Waits for the RTS bit in the RLN24nmLiTRC register to be set to 1 by software. (During this time, “1” is output.) ● When the bit is set to 1, sends a response space. (When not in frame separate mode) ● Sends a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		● Transmits data 2. ● Transmits an inter-byte space ● Transmits data 3. ● Transmits an inter-byte space (Repeats the transmission of data and inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC register.) : :
(5)		Transmits the checksum.
(6)		● Sets the successful frame/wake-up transmission flag. ● Sets the FTS bit in the RLN24nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped) (When in frame separate mode) ● Sets the RTS bit in the RLN24nmLiTRC register to 0 (response transmission stopped).
(7)	● Processing after communication Checks the RLN24nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 21.14, Error Status**.

21.9.3 Response Reception

Figure 21.6, Operation in Response Reception shows the operation of the LIN master interface in response reception. **Table 21.49, Processing in Response Reception** shows the processing in response reception.

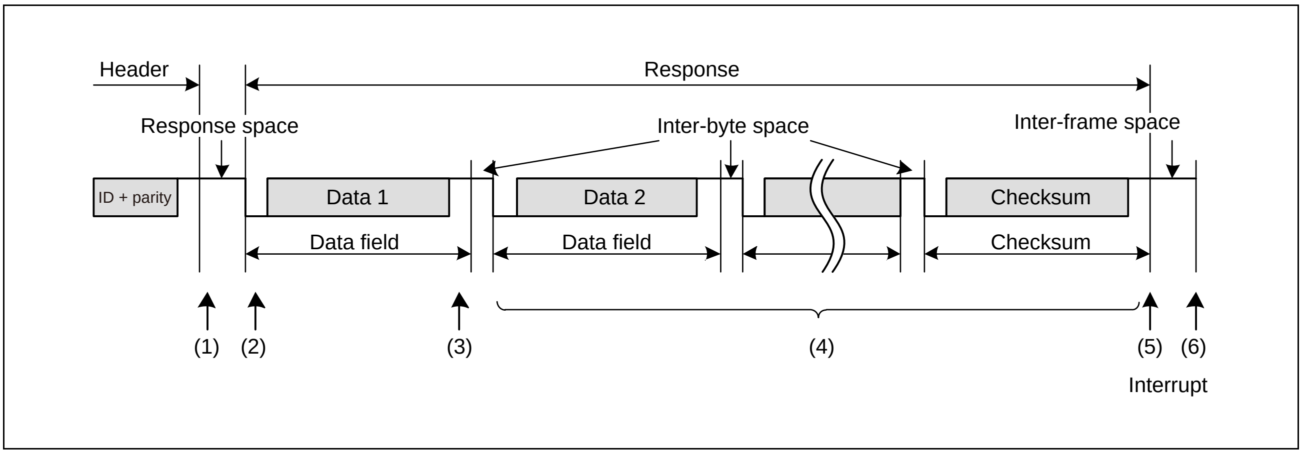


Figure 21.6 Operation in Response Reception

Table 21.49 Processing in Response Reception

	Software Processing	LIN Master Interface Processing
(1)	Waits for an interrupt request (no processing).	Waits for detection of a start bit.
(2)	Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		● Receives data 2 when the start bit is detected. ● Receives data 3 when the start bit is detected. (Repeats the reception of data as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC register.) : : ● Receives the checksum when the start bit is detected.
(5)		● Evaluates the checksum. ● Sets the successful frame/wake-up reception flag. ● Sets the FTS bit in the RLN24nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	● Processing after communication Reads the received data. Checks the RLN24nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 21.14, Error Status**.

21.10 Data Transmission/Reception

21.10.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag of the RLN24nmLiEST register (see **Section 21.14, Error Status**).

In the LIN master interface, because data is generated every 1 Tbit = $16f_{LIN}$, the sampling point for received data is the 13th clock cycle (81.25% position).

Figure 21.7, Example of Data Transmission Timing shows an example of data transmission timing.

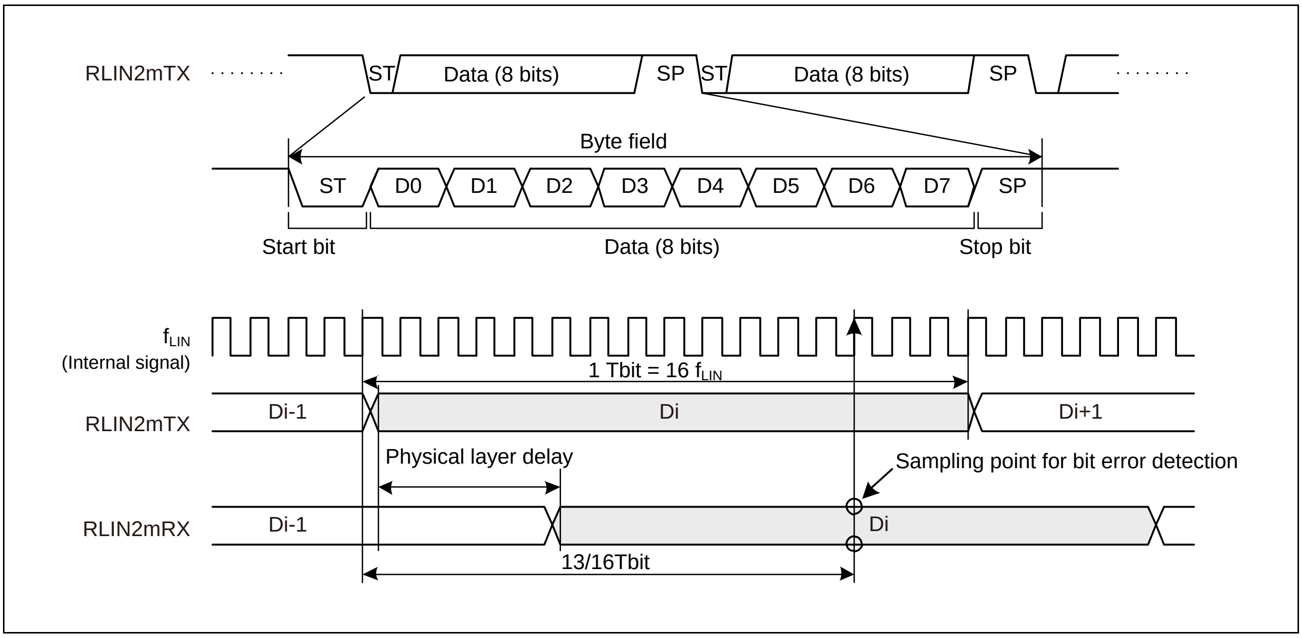


Figure 21.7 Example of Data Transmission Timing

21.10.2 Data Reception

Data reception is performed by using the synchronized RLIN2mRX signal (an internal signal) that is the input from the RLIN2mRX pin synchronized with the LIN system clock (f_{LIN}).

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN2mRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN2mRX signal is low level. The falling edge is not recognized as a start bit if the RLIN2mRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

The bit sampling period after detection of the start bit is one Tbit.

Figure 21.8, Example of Data Reception Timing shows an example of data reception timing.

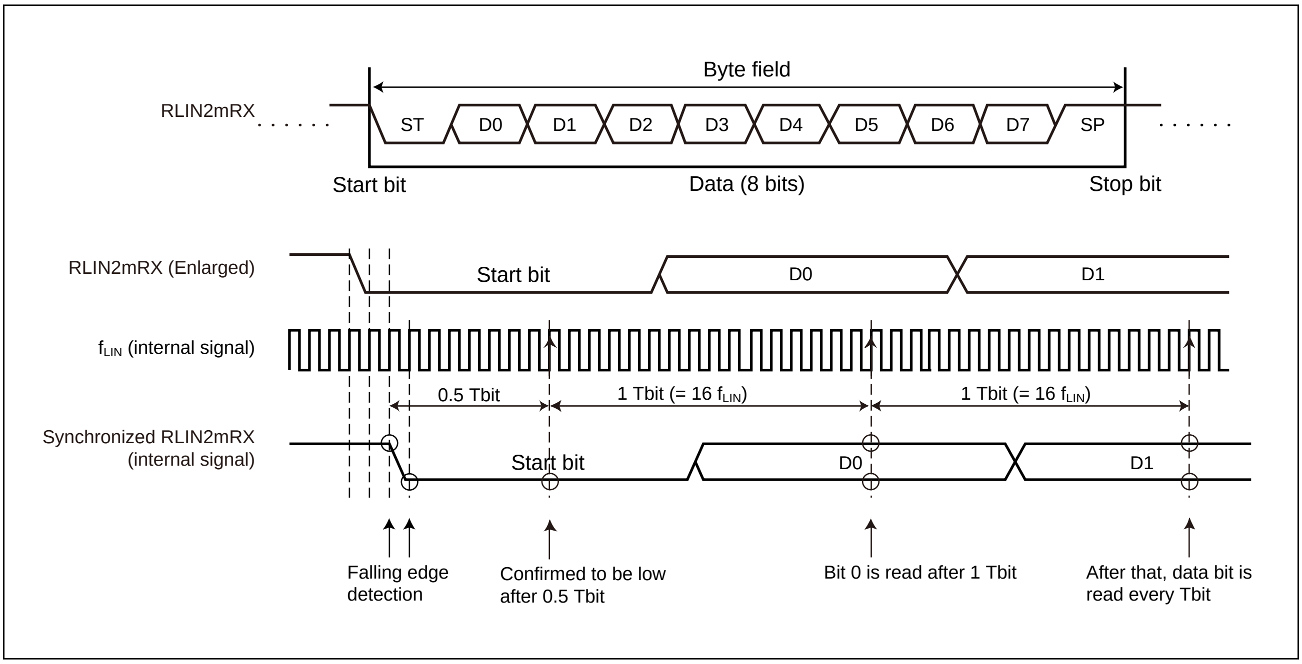


Figure 21.8 Example of Data Reception Timing

21.11 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN master interface sends or receives data continuously.

21.11.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-byte transmission, the contents stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN24nmLiDBR5 to RLN24nmLiDBR8 are not transmitted. The transmitted checksum data is stored in the RLN24nmLiCBR register.

Figure 21.9, LIN Transmission Processing and Corresponding Buffers shows the LIN transmission processing and the corresponding buffers.

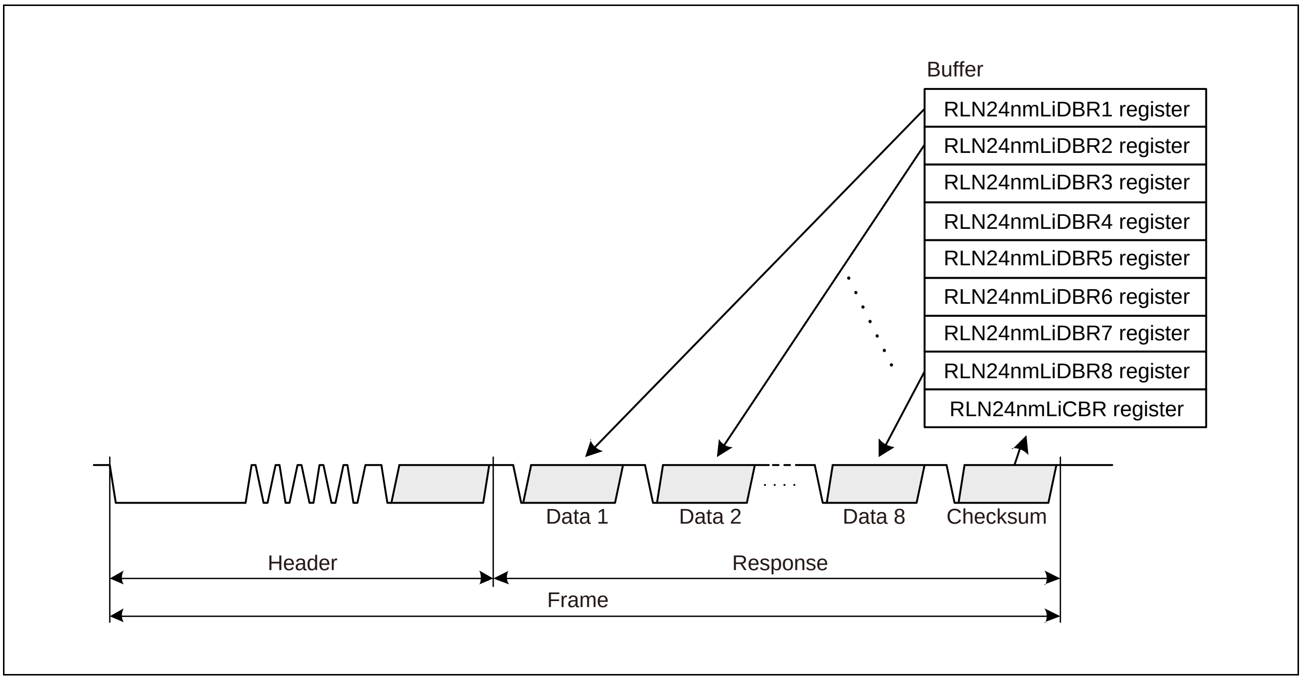


Figure 21.9 LIN Transmission Processing and Corresponding Buffers

(1) Frame Separate Mode

Setting the FSM bit in the RLN24nmLiDFC register to 1 sets the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN24nmLiST register is set to 1 (successful header transmission).

21.11.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR4, respectively; no data is stored in registers RLN24nmLiDBR5 to RLN24nmLiDBR8. The received checksum data is stored in the RLN24nmLiCBR register.

Figure 21.10, LIN Reception Processing and Corresponding Buffers shows the LIN reception processing and the corresponding buffers.

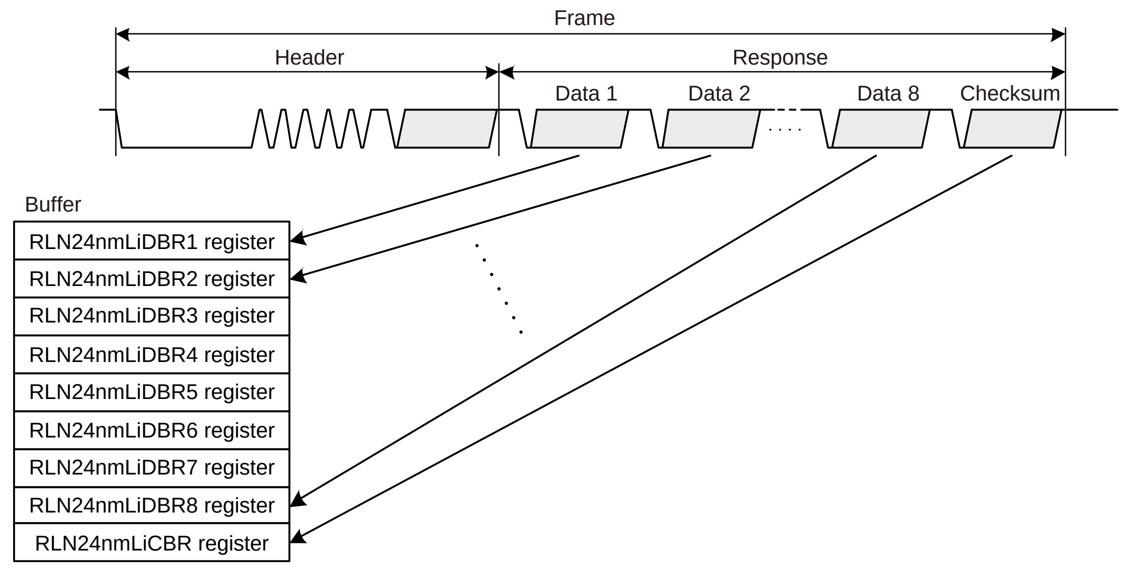


Figure 21.10 LIN Reception Processing and Corresponding Buffers

(1) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the RLN24nmLiST register is set to 1 (successful data 1 reception).

21.12 Wake-Up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

21.12.1 Wake-Up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN24nmLiDFC register to 1 (transmission) and the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal is set using the WUTL[3:0] bits in the RLN24nmLiWUP register. However, if the value of the LWBR0 bit of the RLN24nGLWBR register is 1 (when LIN 2.x is used), the LIN system clock (f_{LIN}) has the low-level width of fa (JW) regardless of the setting of the LCKS bit of the RLN24nmLiMD register. By setting the baud rate to 19200 bps when fa is selected and setting the WUTL[3:0] bits of the RLN24nmLiWUP register to 0100_B (5 Tbits), 260 µs low level width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN24nmLiMD register.

If a wake-up low level is output without any error, the FTC flag in the RLN24nmLiST register is set to 1 (successful frame or wake-up transmission); when the FTCIE bit in the RLN24nmLiIE register is 1 (successful frame/wake-up transmission interrupt enabled), an interrupt request is generated.

If an error is detected, wake-up transmission is aborted and the error flag for the error detected (the PBER flag or BER flag in the RLN24nmLiEST register) is set to 1 (physical bus error detection / bit error detection).

Figure 21.11, Wake-Up Transmission Timing shows the wake-up transmission timing.

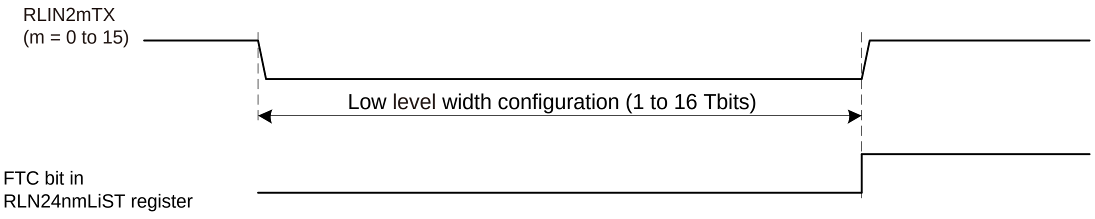


Figure 21.11 Wake-Up Transmission Timing

21.12.2 Wake-Up Reception

The detection of a wake-up signal involves the use of an input signal low level width count function.

The input signal low level width count function measures the low level width of the input signal to the RLIN2mRX pin, using the same sampling point as data reception. This allows an input signal of f_{LIN} with a low-level width of 2.5-Tbit or longer to be measured.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR register to 0. When LIN Specification Package Revision 2.x is used, set LWBR0 bit to 1.

When LWBR0 bit is set to 1, regardless of the setting of the LCKS bit in the RLN24nmLiMD register, fa is selected as the LIN system clock (f_{LIN}) (the LCKS bit is not changed).

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 µs or longer to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN24nmLiDFC register to 0 (reception), and the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN24nmLiST register is set to 1 (successful frame or wake-up reception) and if the FRCIE bit in the RLN24nmLiIE register is 1 (successful frame or wake-up reception interrupt enabled), an interrupt request is generated.

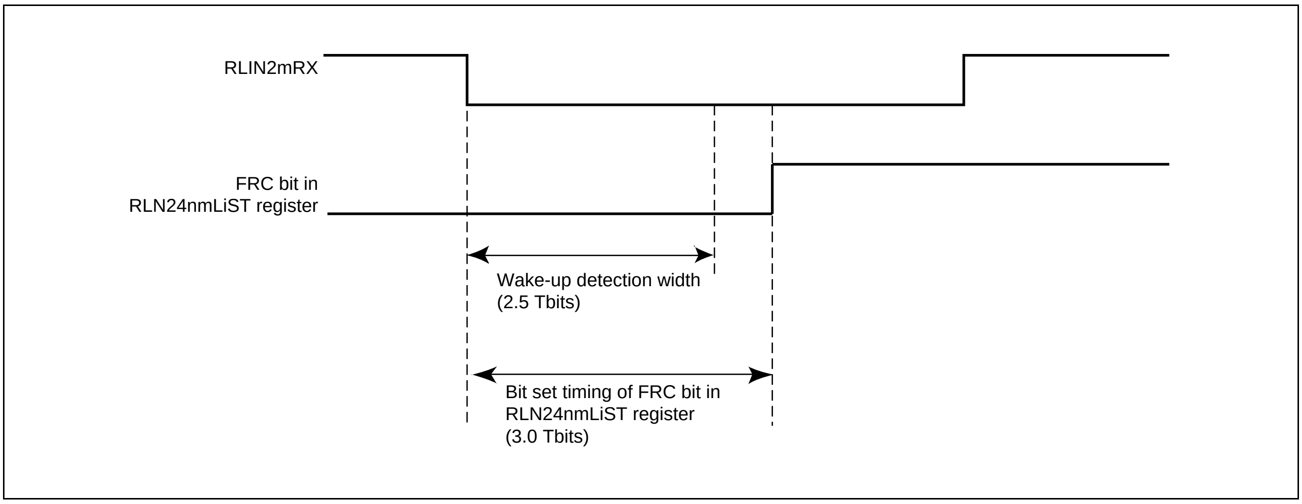


Figure 21.12 Input Signal Low Level Count Function

21.12.3 Wake-Up Collision

If the master node and the slave node transmit wake-up signals simultaneously, a collision will occur on the LIN bus; however the a collision of wake-up signals is not detected in the LIN master interface.

21.13 Status

During LIN mode operation, the LIN master interface can detect seven types of statuses.

Three of these statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, can generate interrupt requests.

Table 21.50, Types of Statuses shows the types of statuses.

Table 21.50 Types of Statuses

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN24nmLiCUC register is set to not-LIN-reset-mode, if the LIN master interface actually exits LIN reset mode.	After the OM0 bit in the RLN24nmLiCUC register is set to LIN reset mode, if the LIN master interface enters LIN reset mode.	All modes	OMM0 bit in the RLN24nmLiMST register	—
Operation mode	After the OM1 bit in the RLN24nmLiCUC register is set to LIN operation mode, if the LIN master interface actually enters LIN operation mode.	After the OM1 bit in the RLN24nmLiCUC register is set to LIN wake-up mode, if the LIN master interface enters LIN wake-up mode.	● LIN operation mode ● LIN wake-up mode	OMM1 bit in the RLN24nmLiMST register	—
Successful frame/wake-up transmission	When a frame (header transmission + response transmission) or a wake-up signal is transmitted successfully.	● When the next communication is started ● When cleared by software ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	FTC flag in the RLN24nmLiST register	✓
Successful frame/wake-up reception	When a frame (header transmission + response reception) or a wake-up signal is received successfully.	● When the next communication is started ● When cleared by software ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	FRC flag in the RLN24nmLiST register	✓
Error detection	If any of the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST register is set to 1 (error detected).	● When the next communication is started ● When cleared by software*[1] ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	ERR flag in the RLN24nmLiST register	✓
Successful data 1 reception	When the RFT bit in the RLN24nmLiDFC register is 0 (reception) and the first byte of the response field is received successfully.*[2]	● When the next communication is started ● When cleared by software ● After transition to LIN reset mode	LIN operation mode	D1RC flag in the RLN24nmLiST register	—
Successful header transmission	When a header field is transmitted successfully.	● When the next communication is started ● When cleared by software ● After transition to LIN reset mode	LIN operation mode	HTRC flag in the RLN24nmLiST register	—

Note 1. While in LIN operation mode, the ERR flag in the RLN24nmLiST register is cleared to 0 by writing 0 to the CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN24nmLiEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN24nmLiDFC register are 0000_B (0 bytes + checksum).

21.14 Error Status

21.14.1 Types of Error Statuses

The LIN master interface can detect five types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN24nmLiEST register.

All error statuses represent interrupt sources.

Table 21.51, Types of Error Statuses shows the types of error statuses.

Table 21.51 Types of Error Statuses

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Selection of Detection Enable/Disable	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1	● LIN operation mode ● LIN wake-up mode	Aborted	✓	BER flag in the RLN24nmLiEST register
Physical bus error	● LIN bus is detected to be high level when sending a break ● LIN bus is detected to be low level when sending a break delimiter ● LIN bus is detected to be high level when sending a wake-up	● LIN operation mode ● LIN wake-up mode	Aborted	✓	PBER flag in the RLN24nmLiEST register
Frame timeout error	A frame transmission/reception does not complete within a given time*2	LIN operation mode	Aborted	✓	FTER flag in the RLN24nmLiEST register
Framing error	In response field reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	✓	FER flag in the RLN24nmLiEST register
Checksum error	In response field reception, checksum test results in an error	LIN operation mode	—	×	CSER flag in the RLN24nmLiEST register

Note 1. If a bit error is detected, the process is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. The timeout time depends on the response field data length (the RFDL[3:0] bits in the RLN24nmLiDFC register) and the checksum selection (the CSM bit in the RLN24nmLiDFC register), and this can be calculated according to the following formula:
When classic checksum is selected (when the CSM bit in the RLN24nmLiDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in the RLN24nmLiDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

The error status is cleared when the next communication is started, when it is cleared by software, or at a transition to LIN reset mode.

21.14.2 Target Time Domain for Error Detection

Figure 21.13, Target Time Domain for Error Detection shows the time domain in which the LIN master interface monitors for error detection.

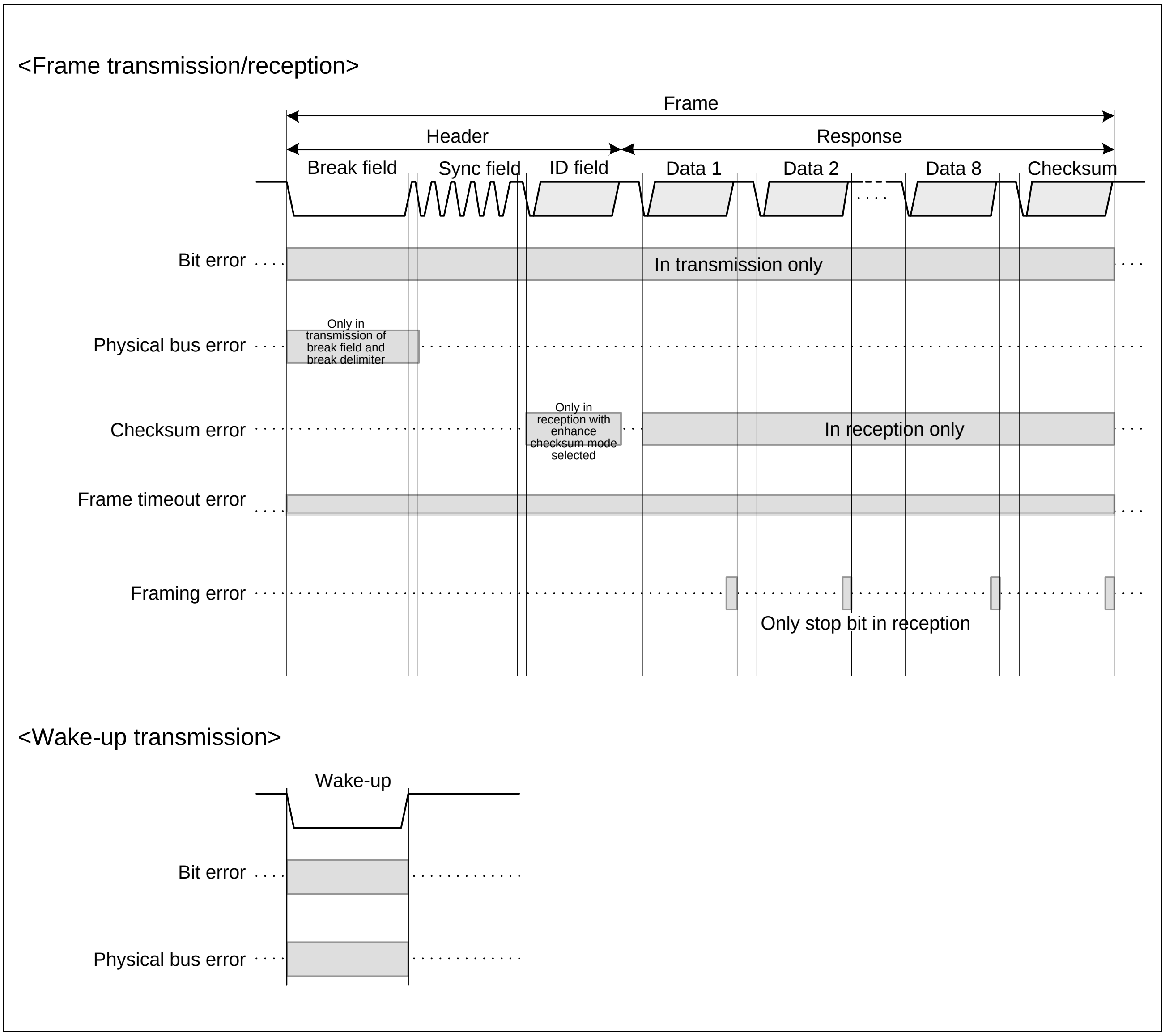


Figure 21.13 Target Time Domain for Error Detection

21.15 LIN Self-Test Mode

The LIN master interface provides LIN self-test mode. When the LIN master interface enters LIN self-test mode, RLIN2mTX and RLIN2mRX are disconnected from the external pins, and are internally connected in the LIN master interface. Thus, the frame transmitted from RLIN2mTX is looped back to RLIN2mRX.

Two types of self-test can be performed:

- LIN self-test mode (transmission): header transmission and response transmission
- LIN self-test mode (reception): header transmission and response reception

In LIN self-test mode, the operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator. Regardless of the setting of the baud rate related registers, the baud rate operates at the LIN communication clock source/16 [bps].

In LIN self-test mode, the following functions are not supported:

- LIN wake-up mode
- Frame separate mode

Do not use these functions.

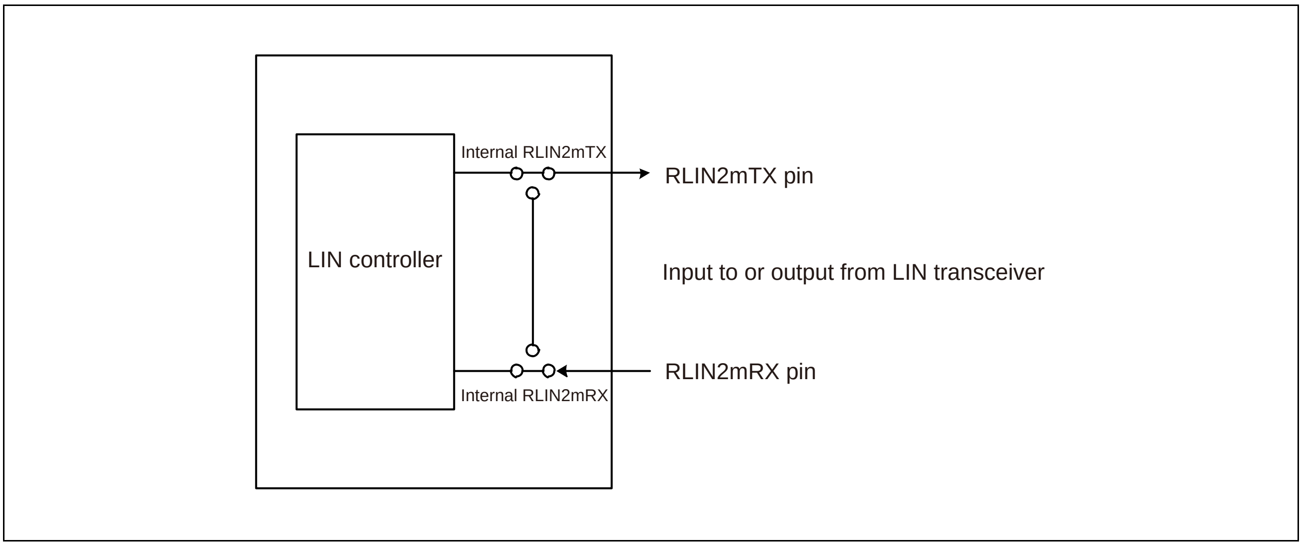


Figure 21.14 Connection in LIN Reset Mode, LIN Wake-Up Mode, and LIN Operation Mode

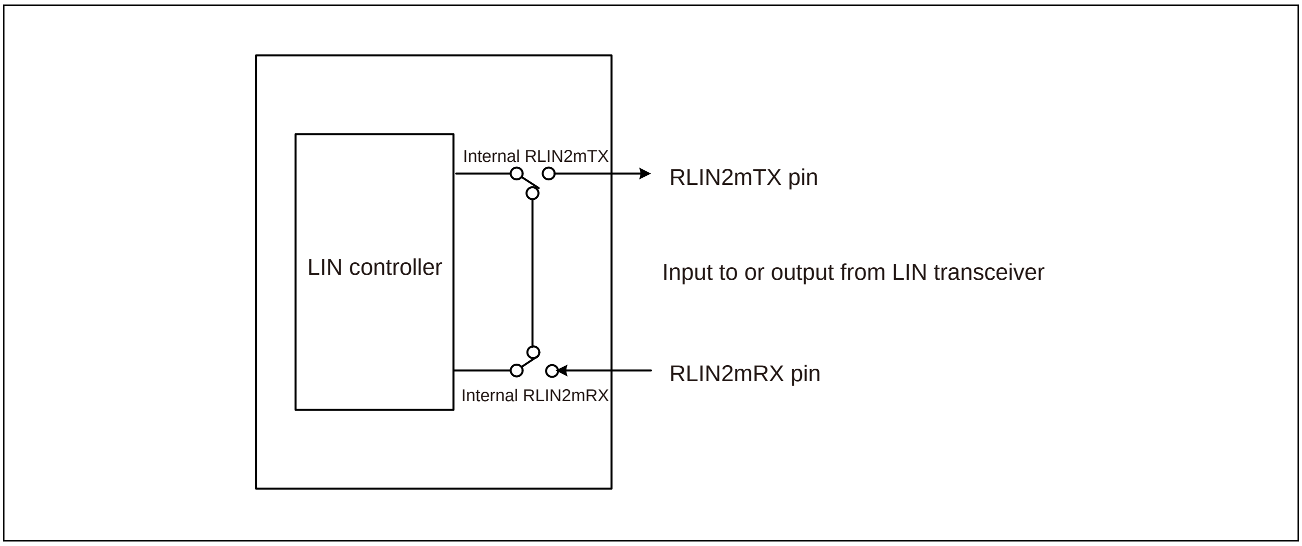


Figure 21.15 Connection in LIN Self-Test Mode

21.15.1 Transition to LIN Self-Test Mode

Writing to the RLN24nGLSTC register enables LIN self-test mode.

The LSTM bit in the RLN24nGLSTC register being set to 1 indicates that the mode has transitioned to LIN self-test mode.

A specific sequence is required to transition to LIN self-test mode. In this sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Switch all channels of the unit to LIN reset mode.
  Set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN24nmLiMST register and confirm that it is 0 (LIN reset mode).
- 1st write: RLN24nGLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN24nGLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN24nGLSTC register = $0000\ 0001_B$ (01_H)
- Confirm that all channels have transitioned to LIN self-test mode
  Read the LSTM bit in the RLN24nGLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the 1st write step. In addition, if a write to another LIN-related register in the same unit is performed during transition to LIN self-test mode (three consecutive write operations to the RLN24nGLSTC register), the transition is also canceled.

21.15.2 Transmission in LIN Self-Test Mode

To execute a self-test on transmission, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 register = xxxx $xxxx_B$*[1]
  RLN24nGLBRP1 register = xxxx $xxxx_B$*[1]
  RLN24nmLiMD register = 0000 $xx00_B$*[1]
- Set interrupt enable register and error enable related registers.
  RLN24nmLiIE register = 0000 $0xxx_B$*[2]
  RLN24nmLiEDE register = 0000 $xxxx_B$
- Set the break field and space related registers.
  RLN24nmLiBFC register = 00xx $xxxx_B$
  RLN24nmLiSC register = 00xx $0xxx_B$
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are 11_B.
- Set the transmit frame related registers.
  RLN24nmLiDFC register = 00x1 $xxxx_B$
  RLN24nmLiIDB register = xxxx $xxxx_B$
  RLN24nmLiDBR1 to RLN24nmLiDBR8 registers = xxxx $xxxx_B$
- Start header transmission → response transmission
  Set the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN self-test mode (transmission) is executed, interrupts are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN master interface.
  To suspend the LIN self-test mode (transmission) while it is running, set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB, RLN24nmLiDBRb, and RLN24nmLiCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC register is cleared.

NOTE

x: Don't care

Note 1. The settings of the following registers are not reflected in the operation of the LIN self-test mode: the RLN24nGLBRP0 register, the RLN24nGLBRP1 register, and the LCKS bit in the RLN24nmLiMD register. Therefore, setting these registers is not necessary.

Note 2. If necessary, set the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

21.15.3 Reception in LIN Self-Test Mode

To execute a self-test on reception, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 register = xxxx xxxx_B*[1]
  RLN24nGLBRP1 register = xxxx xxxx_B*[1]
  RLN24nmLiMD register = 0000 xx00_B*[1]
- Set the interrupt enable and error enable related registers.
  RLN24nmLiIE register = 0000 0xxx_B*[2]
  RLN24nmLiEDE register = 0000 x0xx_B
- Set the break field and space related registers.
  RLN24nmLiBFC register = 00xx xxxx_B
  RLN24nmLiSC register = 00xx 0xxx_B*[1]
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are 11_B.
- Set the receive frame related registers.
  RLN24nmLiDFC register = 00x0 xxxx_B
  RLN24nmLiIDB register = xxxx xxxx_B
  RLN24nmLiDBR1 to RLN24nmLiDBR8 registers = xxxx xxxx_B
  RLN24nmLiCBR register = xxxx xxxx_B
  Since the checksum value to be transmitted is not automatically calculated, perform the calculation and specify the calculated value in the RLN24nmLiCBR register. By specifying an incorrect checksum, the checksum error can be tested.
- Start header transmission → response reception
  Set the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started). The LIN self-test mode (reception) is executed, interrupts are generated, and status and error status are also updated. To suspend the LIN self-test mode (reception) while it is running, set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB, RLN24nmLiDBRb, and RLN24nmLiCBR registers (the data is reversed before being stored because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC register is cleared.

NOTE

x: Don’t care

Note 1. The settings of the following registers are not reflected to the operation of the LIN self-test mode: the RLN24nGLBRP0 register, the RLN24nGLBRP1 register, the LCKS bit in the RLN24nmLiMD register, and the IBS bit and IBHS bit (response space only) in the RLN24nmLiSC register. Therefore, setting these registers is not necessary.

Note 2. If necessary, set the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

21.15.4 Exiting LIN Self-Test Mode

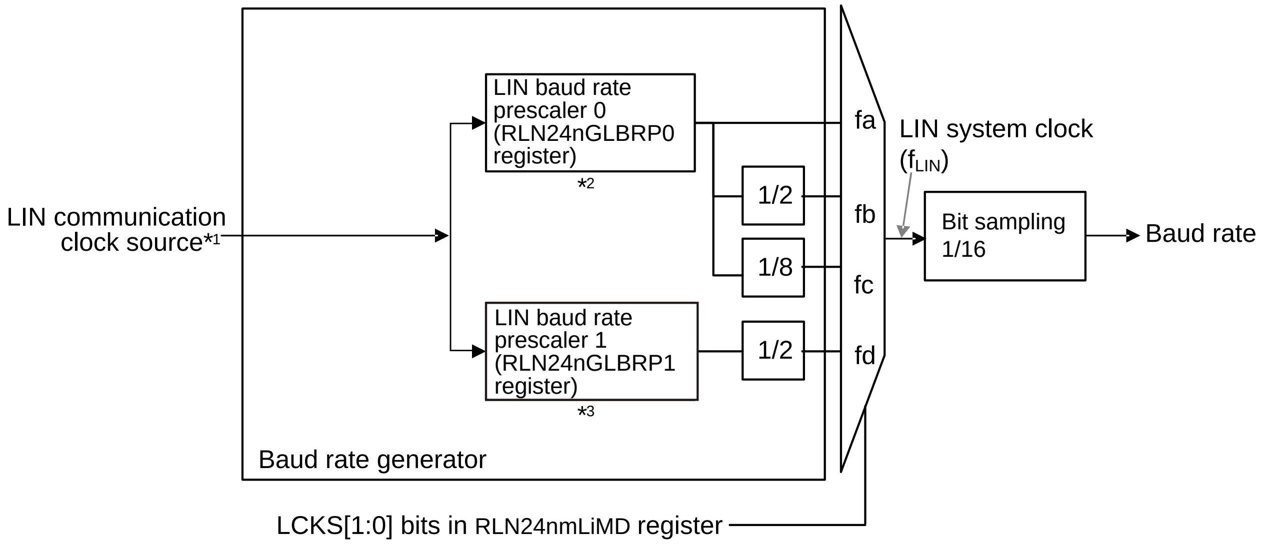
To exit LIN self-test mode, perform the procedure below:

- Switch all channels of the unit to LIN reset mode.
  Write 0 to the OM0 bit in the RLN24nmLiCUC register to make a transition to LIN reset mode. However, if the OMM1 and OMM0 bits in the RLN24nmLiMST register are not 11_B in any channels of the unit after the transition to LIN self-test mode, write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register in any one channel. Check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are set to 11_B, and then make a transition to LIN reset mode.
- Verify that the LIN master interface has exited LIN self-test mode.
  Read the LSTM bit in the RLN24nGLSTC register and confirm that it is not 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN24nmLiMST register and confirm that it is 0 (LIN reset mode).

21.16 Baud Rate Generator

The LIN system clock (f_{LIN}) is obtained by dividing the LIN communication clock source frequency by the baud rate generator, and the baud rate is obtained by dividing that clock by 16. The inverse of this baud rate is called the bit time (Tbit).

Figure 21.16, Block Diagram of Baud Rate Generation shows a block diagram of baud rate generation.



Note 1. For the LIN communication clock source, see **Section 12AB.5.3, Clock Domain Settings**, **12C.5.3, Clock Domain Settings**.

Note 2. According to the RLN24nGLBRP0 register, the clock frequency is divided by N + 1 (N = 0 to 255).

Note 3. According to the RLN24nGLBRP1 register, the clock frequency is divided by M + 1 (M = 0 to 255).

Figure 21.16 Block Diagram of Baud Rate Generation

Set the LIN communication clock source in a range from 4 MHz to 40 MHz.

By setting the RLN24nGLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting system clock frequencies are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These system clock frequencies are divided by 16 in the bit timing generator, enabling baud rates of 19200 bps, 9600 bps, and 2400 bps to be generated. Also, by setting the RLN24nGLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting system clock frequency is fd = 10417 × 16. This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for calculating baud rate is shown below.

Baud rate:

= {Frequency of LIN communication clock source} ÷ (RLN24nGLBRP0 + 1) ÷ 16 [bps] (When fa is selected)

= {Frequency of LIN communication clock source} ÷ (RLN24nGLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected)

= {Frequency of LIN communication clock source} ÷ (RLN24nGLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected)

= {Frequency of LIN communication clock source} ÷ (RLN24nGLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected)

Section 22 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RLIN3.

22.1 Features of RH850/F1KH, RH850/F1KM RLIN3

22.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has a single channel interface.

Table 22.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of units	8	8	8
Name	RLIN3n (n = 0 to 7)	RLIN3n (n = 0 to 7)	RLIN3n (n = 0 to 7)

Table 22.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of units	3	6	8	8	8
Name	RLIN3n (n = 0 to 2)	RLIN3n (n = 0 to 5)	RLIN3n (n = 0 to 7)	RLIN3n (n = 0 to 7)	RLIN3n (n = 0 to 7)

Table 22.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of units	1	2	3	4
Name	RLIN3n (n = 0)	RLIN3n (n = 0, 1)	RLIN3n (n = 0 to 2)	RLIN3n (n = 0 to 3)

Table 22.4 Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name (Channel Name) RLIN3n	Number of Channels per Unit	RH850/F1KH-D8 176 Pins (8 ch)	RH850/F1KH-D8 233 Pins (8 ch)	RH850/F1KH-D8 324 Pins (8 ch)
RLIN30	1	✓	✓	✓
RLIN31	1	✓	✓	✓
RLIN32	1	✓	✓	✓
RLIN33	1	✓	✓	✓
RLIN34	1	✓	✓	✓
RLIN35	1	✓	✓	✓
RLIN36	1	✓	✓	✓
RLIN37	1	✓	✓	✓

Table 22.5 Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name (Channel Name) RLIN3n	Number of Channels per Unit	RH850/F1KM-S4 100 Pins (3 ch)	RH850/F1KM-S4 144 Pins (6 ch)	RH850/F1KM-S4 176 Pins (8 ch)	RH850/F1KM-S4 233 Pins (8 ch)	RH850/F1KM-S4 272 Pins (8 ch)
RLIN30	1	✓	✓	✓	✓	✓
RLIN31	1	✓	✓	✓	✓	✓
RLIN32	1	✓	✓	✓	✓	✓
RLIN33	1	—	✓	✓	✓	✓
RLIN34	1	—	✓	✓	✓	✓
RLIN35	1	—	✓	✓	✓	✓
RLIN36	1	—	—	✓	✓	✓
RLIN37	1	—	—	✓	✓	✓

Table 22.6 Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name (Channel Name) RLIN3n	Number of Channels per Unit	RH850/F1KM-S1 48 Pins (1 ch)	RH850/F1KM-S1 64 Pins (2 ch)	RH850/F1KM-S1 80 Pins (3 ch)	RH850/F1KM-S1 100 Pins (4 ch)
RLIN30	1	✓	✓	✓	✓
RLIN31	1	—	✓	✓	✓
RLIN32	1	—	—	✓	✓
RLIN33	1	—	—	—	✓

Note: The channel names are same as those of the corresponding units.

Table 22.7 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index “n”: for example, RLN3nLCUC (n = 0 to 7) is the LIN control register.
b	Throughout this section, the individual transmit/received data buffers of RLIN3n are identified by the index “b”: for example, RLN3nLDBRb (b = 1 to 8) is the data buffer register.

Table 22.8 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index “n”: for example, RLN3nLCUC (n = 0 to 7) is the LIN control register.
b	Throughout this section, the individual transmit/received data buffers of RLIN3n are identified by the index “b”: for example, RLN3nLDBRb (b = 1 to 8) is the data buffer register.

Table 22.9 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index “n”: for example, RLN3nLCUC (n = 0 to 3) is the LIN control register.
b	Throughout this section, the individual transmit/received data buffers of RLIN3n are identified by the index “b”: for example, RLN3nLDBRb (b = 1 to 8) is the data buffer register.

The following lists the index value corresponding to each product.

Table 22.10 Index Correspondence of Each Product (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index Correspondence to Product
All Products
b = 1 to 8

22.1.2 Register Base Addresses

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses.

Table 22.11 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<RLIN30_base>	FFCE 2000$_H$
<RLIN31_base>	FFCE 2040$_H$
<RLIN32_base>	FFCE 2080$_H$
<RLIN33_base>	FFCE 20C0$_H$
<RLIN34_base>	FFCE 2100$_H$
<RLIN35_base>	FFCE 2140$_H$
<RLIN36_base>	FFCE 2180$_H$
<RLIN37_base>	FFCE 21C0$_H$

Table 22.12 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<RLIN30_base>	FFCE 2000$_H$
<RLIN31_base>	FFCE 2040$_H$
<RLIN32_base>	FFCE 2080$_H$
<RLIN33_base>	FFCE 20C0$_H$
<RLIN34_base>	FFCE 2100$_H$
<RLIN35_base>	FFCE 2140$_H$
<RLIN36_base>	FFCE 2180$_H$
<RLIN37_base>	FFCE 21C0$_H$

Table 22.13 Register Base Addresses (RH850/F1KM-S1)

Base Address Name	Base Address
<RLIN30_base>	FFCE 2000$_H$
<RLIN31_base>	FFCE 2040$_H$
<RLIN32_base>	FFCE 2080$_H$
<RLIN33_base>	FFCE 20C0$_H$

22.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 22.14 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RLIN3n	LIN communication clock sources	CKSCLK_ILIN*1,*2	Communication clock
	Register access clock	CPUCLK_L, CKSCLK_ILIN	Bus clock

Note 1. The clock domain CKSCLK_ILIN divided clock can be supplied only to RLIN30 channel.

Note 2. Set the LIN communication clock source in the range of 4 MHz to 40 MHz.

22.1.4 Interrupt Requests

RLIN3 interrupt requests are listed in the following table.

Table 22.15 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RLIN30			
INTRLIN3n (n = 0)	RLIN30 interrupt	33	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	34	10
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	35	11
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	36	—
RLIN31			
INTRLIN3n (n = 1)	RLIN31 interrupt	120	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	121	86
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	122	87
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	123	—
RLIN32			
INTRLIN3n (n = 2)	RLIN32 interrupt	164	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	165	44
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	166	45
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	167	—
RLIN33			
INTRLIN3n (n = 3)	RLIN33 interrupt	228	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	229	111
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	230	112
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	231	—
RLIN34			
INTRLIN3n (n = 4)	RLIN34 interrupt	232	—
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	233	50
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	234	51
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	235	—
RLIN35			
INTRLIN3n (n = 5)	RLIN35 interrupt	236	—
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	237	121
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	238	122
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	239	—
RLIN36			
INTRLIN3n (n = 6)	RLIN36 interrupt	360	—
INTRLIN3nUR0 (n = 6)	RLIN36 transmit interrupt	361	119
INTRLIN3nUR1 (n = 6)	RLIN36 receive completion interrupt	362	120
INTRLIN3nUR2 (n = 6)	RLIN36 status interrupt	363	—
RLIN37			
INTRLIN3n (n = 7)	RLIN37 interrupt	364	—
INTRLIN3nUR0 (n = 7)	RLIN37 transmit interrupt	365	16
INTRLIN3nUR1 (n = 7)	RLIN37 receive completion interrupt	366	77
INTRLIN3nUR2 (n = 7)	RLIN37 status interrupt	367	—

Table 22.16 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RLIN30			
INTRLIN3n (n = 0)	RLIN30 interrupt	33	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	34	10
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	35	11
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	36	—
RLIN31			
INTRLIN3n (n = 1)	RLIN31 interrupt	120	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	121	86
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	122	87
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	123	—
RLIN32			
INTRLIN3n (n = 2)	RLIN32 interrupt	164	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	165	44
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	166	45
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	167	—
RLIN33			
INTRLIN3n (n = 3)	RLIN33 interrupt	228	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	229	111
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	230	112
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	231	—
RLIN34			
INTRLIN3n (n = 4)	RLIN34 interrupt	232	—
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	233	50
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	234	51
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	235	—
RLIN35			
INTRLIN3n (n = 5)	RLIN35 interrupt	236	—
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	237	121
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	238	122
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	239	—
RLIN36			
INTRLIN3n (n = 6)	RLIN36 interrupt	360	—
INTRLIN3nUR0 (n = 6)	RLIN36 transmit interrupt	361	119
INTRLIN3nUR1 (n = 6)	RLIN36 receive completion interrupt	362	120
INTRLIN3nUR2 (n = 6)	RLIN36 status interrupt	363	—
RLIN37			
INTRLIN3n (n = 7)	RLIN37 interrupt	364	—
INTRLIN3nUR0 (n = 7)	RLIN37 transmit interrupt	365	16
INTRLIN3nUR1 (n = 7)	RLIN37 receive completion interrupt	366	77
INTRLIN3nUR2 (n = 7)	RLIN37 status interrupt	367	—

Table 22.17 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RLIN30			
INTRLIN3n (n = 0)	RLIN30 interrupt	33	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	34	10
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	35	11
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	36	—
RLIN31			
INTRLIN3n (n = 1)	RLIN31 interrupt	120	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	121	86
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	122	87
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	123	—
RLIN32			
INTRLIN3n (n = 2)	RLIN32 interrupt	164	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	165	44
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	166	45
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	167	—
RLIN33			
INTRLIN3n (n = 3)	RLIN33 interrupt	228	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	229	111
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	230	112
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	231	—

22.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 22.18 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RLIN3n	All reset sources (ISORES)

22.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 22.19 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN30		
RLIN3nRX (n = 0)	RLIN30 receive data input	RLIN30RX
RLIN3nTX (n = 0)	RLIN30 transmit data output	RLIN30TX
RLIN31		
RLIN3nRX (n = 1)	RLIN31 receive data input	RLIN31RX
RLIN3nTX (n = 1)	RLIN31 transmit data output	RLIN31TX
RLIN32		
RLIN3nRX (n = 2)	RLIN32 receive data input	RLIN32RX
RLIN3nTX (n = 2)	RLIN32 transmit data output	RLIN32TX
RLIN33		
RLIN3nRX (n = 3)	RLIN33 receive data input	RLIN33RX
RLIN3nTX (n = 3)	RLIN33 transmit data output	RLIN33TX
RLIN34		
RLIN3nRX (n = 4)	RLIN34 receive data input	RLIN34RX
RLIN3nTX (n = 4)	RLIN34 transmit data output	RLIN34TX
RLIN35		
RLIN3nRX (n = 5)	RLIN35 receive data input	RLIN35RX
RLIN3nTX (n = 5)	RLIN35 transmit data output	RLIN35TX
RLIN36		
RLIN3nRX (n = 6)	RLIN36 receive data input	RLIN36RX
RLIN3nTX (n = 6)	RLIN36 transmit data output	RLIN36TX
RLIN37		
RLIN3nRX (n = 7)	RLIN37 receive data input	RLIN37RX
RLIN3nTX (n = 7)	RLIN37 transmit data output	RLIN37TX

Table 22.20 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN30		
RLIN3nRX (n = 0)	RLIN30 receive data input	RLIN30RX
RLIN3nTX (n = 0)	RLIN30 transmit data output	RLIN30TX
RLIN31		
RLIN3nRX (n = 1)	RLIN31 receive data input	RLIN31RX
RLIN3nTX (n = 1)	RLIN31 transmit data output	RLIN31TX
RLIN32		
RLIN3nRX (n = 2)	RLIN32 receive data input	RLIN32RX
RLIN3nTX (n = 2)	RLIN32 transmit data output	RLIN32TX
RLIN33		
RLIN3nRX (n = 3)	RLIN33 receive data input	RLIN33RX
RLIN3nTX (n = 3)	RLIN33 transmit data output	RLIN33TX
RLIN34		
RLIN3nRX (n = 4)	RLIN34 receive data input	RLIN34RX
RLIN3nTX (n = 4)	RLIN34 transmit data output	RLIN34TX
RLIN35		
RLIN3nRX (n = 5)	RLIN35 receive data input	RLIN35RX
RLIN3nTX (n = 5)	RLIN35 transmit data output	RLIN35TX
RLIN36		
RLIN3nRX (n = 6)	RLIN36 receive data input	RLIN36RX
RLIN3nTX (n = 6)	RLIN36 transmit data output	RLIN36TX
RLIN37		
RLIN3nRX (n = 7)	RLIN37 receive data input	RLIN37RX
RLIN3nTX (n = 7)	RLIN37 transmit data output	RLIN37TX

Table 22.21 External Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN30		
RLIN3nRX (n = 0)	RLIN30 receive data input	RLIN30RX
RLIN3nTX (n = 0)	RLIN30 transmit data output	RLIN30TX
RLIN31		
RLIN3nRX (n = 1)	RLIN31 receive data input	RLIN31RX
RLIN3nTX (n = 1)	RLIN31 transmit data output	RLIN31TX
RLIN32		
RLIN3nRX (n = 2)	RLIN32 receive data input	RLIN32RX
RLIN3nTX (n = 2)	RLIN32 transmit data output	RLIN32TX
RLIN33		
RLIN3nRX (n = 3)	RLIN33 receive data input	RLIN33RX
RLIN3nTX (n = 3)	RLIN33 transmit data output	RLIN33TX

22.2 Overview

22.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, LIN slave, or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
    - LIN wake-up mode
    - LIN operation mode
- LIN self-test mode

LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
    - LIN wake-up mode
    - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

Table 22.22, LIN/UART Interface Specifications shows the LIN/UART interface specifications.

Table 22.22 LIN/UART Interface Specifications

Item		Specifications	
	Channel count	Up to 8 channels	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602	
	Variable frame structure	Master	• Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Transmission response space width: 0 to 7 Tbits*1 • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
		Slave	• Break reception width: 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate] • Transmission response space width: 0 to 7 Tbits • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible	
	Frame communication modes	Master	• Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
		Slave	• Mode in which header is automatically received with fixed baud rate • Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected
	Wake-up transmission and reception	LIN wake-up mode provided • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
		Slave	• Successful response/wake-up transmission • Successful response/wake-up reception*2 • Successful header reception • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)

Table 22.22 LIN/UART Interface Specifications

Item		Specifications	
LIN communication function	Error status	Master	● Bit error ● Checksum error ● Frame timeout error/response timeout error ● Physical bus error ● Framing error ● Response preparation error
		Slave	● Bit error ● Checksum error ● Frame timeout error/response timeout error ● Sync field error ● ID parity error ● Framing error ● Response preparation error
	Baud rate selection	Baud rates conforming to the LIN specifications generated using baud rate generator	
	Test mode	Self-test mode for user evaluation	
	Interrupt function	Master	● Successful header/frame/wake-up transmission ● Successful frame/wake-up reception*2 ● Error detection
		Slave	● Successful response/wake-up transmission ● Successful Header/response/wake-up reception*2 ● Error detection
UART communication function	Data buffer	● Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) ● UART buffer (exclusively for transmission; variable data length from 1 to 9. Character length of 7 and 8 bits supported) ● Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)	
	Data format	Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit.	
		Transmission stop bit: 1 or 2 bits	
		Parity function: odd, even, 0, or none	
		LSB- or MSB-first transfer selectable	
		Reverse input/output of transmission/reception data possible	
	Status	● Transmission status ● Reception status ● Successful UART buffer transmission ● Error detection ● Expansion bit detection ● ID match ● Reset mode status	
	Error status	● Bit error ● Framing error ● Parity error ● Overrun error	
	Baud rate selection	With the built-in baud rate generator, any baud rate can be set.	
	When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register.		

Table 22.22 LIN/UART Interface Specifications

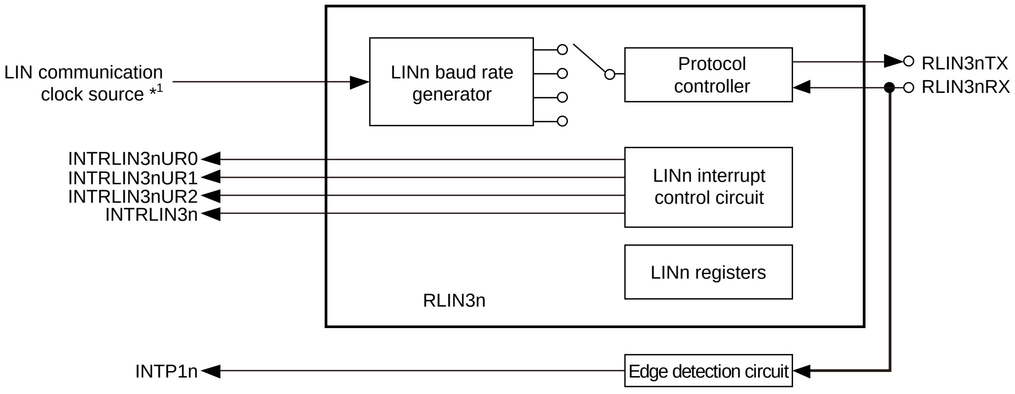
Item	Specifications	
UART communication function	Reception of the stop bit is guaranteed. (Start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
	Interrupt function	• Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

22.2.2 Block Diagram

Figure 22.1, LIN/UART Interface Block Diagram shows a block diagram of the LIN/UART interface.



Note 1. For the LIN communication clock source, see **Section 12AB.5.3, Clock Domain Settings**, **12C.5.3, Clock Domain Settings**.

Figure 22.1 LIN/UART Interface Block Diagram

22.2.3 Terms Used in Block Diagram

• RLIN3nTX, RLIN3nRX:	LIN/UART interface I/O pins
• LINn baud rate generator:	Generates the LIN/UART interface communication clock
• LINn registers:	LIN/UART interface registers
• LINn interrupt control circuit:	Controls interrupt requests generated by the LIN/UART interface

22.3 Registers

22.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n_base>, see **Section 22.1.2, Register Base Addresses**.

Table 22.23 List of Registers

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	LIN wake-up baud rate select register	RLN3nLWBR	<RLIN3n_base> + 01_H	✓	✓	✓
	LIN / UART baud rate prescaler 01 register	RLN3nLBRP01	<RLIN3n_base> + 02_H	—	✓	✓
	LIN / UART baud rate prescaler 0 register	RLN3nLBRP0	<RLIN3n_base> + 02_H	✓	✓	✓
	LIN / UART baud rate prescaler 1 register	RLN3nLBRP1	<RLIN3n_base> + 03_H	✓	✓	✓
	LIN self-test control register	RLN3nLSTC	<RLIN3n_base> + 04_H	✓	✓	—
	LIN / UART mode register	RLN3nLMD	<RLIN3n_base> + 08_H	✓	✓	✓
	LIN break field configuration register/ UART configuration register	RLN3nLBFC	<RLIN3n_base> + 09_H	✓	✓	✓
	LIN / UART space configuration register	RLN3nLSC	<RLIN3n_base> + $0A_H$	✓	✓	✓
	LIN wake-up configuration register	RLN3nLWUP	<RLIN3n_base> + $0B_H$	✓	✓	—
	LIN interrupt enable register	RLN3nLIE	<RLIN3n_base> + $0C_H$	✓	✓	—
	LIN / UART error detection enable register	RLN3nLEDE	<RLIN3n_base> + $0D_H$	✓	✓	✓
	LIN / UART control register	RLN3nLCUC	<RLIN3n_base> + $0E_H$	✓	✓	✓
	LIN / UART transmission control register	RLN3nLTRC	<RLIN3n_base> + 10_H	✓	✓	✓
	LIN / UART mode status register	RLN3nLMST	<RLIN3n_base> + 11_H	✓	✓	✓
	LIN / UART status register	RLN3nLST	<RLIN3n_base> + 12_H	✓	✓	✓
	LIN / UART error status register	RLN3nLEST	<RLIN3n_base> + 13_H	✓	✓	✓
	LIN / UART data field configuration register	RLN3nLDFC	<RLIN3n_base> + 14_H	✓	✓	✓
	LIN / UART ID buffer register	RLN3nLIDB	<RLIN3n_base> + 15_H	✓	✓	✓
	LIN checksum buffer register	RLN3nLCBR	<RLIN3n_base> + 16_H	✓	✓	—
	UART data buffer 0 register	RLN3nLUDB0	<RLIN3n_base> + 17_H	—	—	✓
	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLIN3n_base> + 18_H	✓	✓	✓
	LIN / UART data buffer 2 register	RLN3nLDBR2	<RLIN3n_base> + 19_H	✓	✓	✓
	LIN / UART data buffer 3 register	RLN3nLDBR3	<RLIN3n_base> + $1A_H$	✓	✓	✓
	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLIN3n_base> + $1B_H$	✓	✓	✓
	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLIN3n_base> + $1C_H$	✓	✓	✓
	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLIN3n_base> + $1D_H$	✓	✓	✓
	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLIN3n_base> + $1E_H$	✓	✓	✓
	LIN / UART data buffer 8 register	RLN3nLDBR8	<RLIN3n_base> + $1F_H$	✓	✓	✓
	UART operation enable register	RLN3nLUOER	<RLIN3n_base> + 20_H	—	—	✓
	UART option register 1	RLN3nLUOR1	<RLIN3n_base> + 21_H	—	—	✓
	UART transmission data register	RLN3nLUTDR	<RLIN3n_base> + 24_H	—	—	✓
	UART transmission data register L	RLN3nLUTDRL	<RLIN3n_base> + 24_H	—	—	✓
	UART transmission data register H	RLN3nLUTDRH	<RLIN3n_base> + 25_H	—	—	✓

Table 22.23 List of Registers

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	UART reception data register	RLN3nLURDR	<RLIN3n_base> + 26_H	—	—	✓
	UART reception data register L	RLN3nLURDRL	<RLIN3n_base> + 26_H	—	—	✓
	UART reception data register H	RLN3nLURDRH	<RLIN3n_base> + 27_H	—	—	✓
	UART wait transmission data register	RLN3nLUWTDR	<RLIN3n_base> + 28_H	—	—	✓
	UART wait transmission data register L	RLN3nLUWTDRL	<RLIN3n_base> + 28_H	—	—	✓
	UART wait transmission data register H	RLN3nLUWTDRH	<RLIN3n_base> + 29_H	—	—	✓

Remark: ✓: Used, —: Not used

Note: When writing to an unused register, write the value after reset.

22.3.2 LIN Master Related Registers

22.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.24 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x)

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B or 1111_B (16 sampling).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.
When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (f_{LIN}) during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 us or longer to be detected in the LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

22.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.25 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00_H to FF_H

Configure the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

22.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.26 RLN3nLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00_H to FF_H

Configure the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M + 1.

22.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access:	This register can be read or written in 8-bit units.
Address:	<RLIN3n_base> + 04_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.27 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000_B”, and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

22.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base>+ 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.28 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 0: LIN master mode

Configure the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 22.4, Interrupt Sources**.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is supplied with fa regardless of the setting of this bit (the LCKS bit is not changed).

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an LIN master, set these bits to 00_B.

22.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.29 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the specified values result in a frame length exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.
1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.
13 Tbits to 28 Tbits can be set.

22.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 22.30 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Specify the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

22.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 22.31 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.
1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), fa is selected as the LIN system clock (f_{LIN}) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

22.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 22.32 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Configure the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Request Enable)

The SHIE bit enables or disables interrupt request upon successful transmission of a header.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors. Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).
With 0 set, the interrupt request for RLIN3n successful reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

22.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 22.33 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable*[1] 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable*[1] 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details on the timeout error, see **Section 22.7.7, Error Statuses**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 22.7.7, Error Statuses**.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is reflected in the FTER flag of the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 22.7.7, Error Statuses**.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.
With 0 set, the physical bus error is not detected.
With 1 set, the physical bus error is detected.
When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.
For details on the physical bus error, see **Section 22.7.7, Error Statuses**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 22.7.7, Error Statuses**.

22.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.34 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode 1: LIN operation mode
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode.
With 1 set, LIN operation mode.
This bit is enabled only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode.
With 1 set, LIN reset mode is canceled.

22.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.35 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission/Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

RTS Bit (Response Transmission/Reception Start)

In frame separate mode, set the RTS bit to 1 after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.
Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

22.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.36 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

22.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 22.37 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value it had before 1 is written.
The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value it had before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register while in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Successful Frame/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value it had before 1 is written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n successful reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Frame/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value it had before 1 is written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

22.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 22.38 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value it had before 1 is written.
The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value it had before 1 is written.
The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

22.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.39 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.

With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit selects the response communication mode.
With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register.
With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.
When transitioning to LIN self-test mode, set this bit to 0 before transition.
For details on frame separate mode, see **Section 22.7.4.1, Transmission of LIN Frames**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit selects the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error detection is enabled (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details, see **Section 22.7.7, Error Statuses**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).
To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).
Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

22.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.40 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

22.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.41 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Stores the checksum value transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

22.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.42 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the received data. Setting range: 00_H to FF_H

- For response transmission:
  These registers set the data to be transmitted in the response field.
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 0 (not frame separate mode)
  - FTS in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped)

  or
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is stopped)
- For response reception:
  These registers store the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started).
- For transmission of response data of 9 bytes or more:
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is stopped)
- For reception of response data of 9 bytes or more:
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

22.3.3 LIN Slave Related Registers

22.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 22.43 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 0 1 1: 4 samplings 0 1 1 1: 8 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
When the frame communication is performed in LIN slave mode (fixed baud rate) (LMD[1:0] bits in the RLN3nLMD register = 11_B), set these bits to “0000_B” or “1111_B” (16 samplings).
When the frame communication is performed in LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits to “0011_B” (4 samplings) or “0111_B” (8 samplings).

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

In LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits so that the prescaler clock becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4 MHz*[1]
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4 MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB bits set to “0011_B” (4 samplings).

22.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 can be read or written in 16-bit units.
RLN3nLBRP0 can be read or written in 8-bit units.
RLN3nLBRP1 can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.44 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Configure the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units using registers RLN3nLBRP0 and RLN3nLBRP1.

NOTE

In LIN slave mode [auto baud rate], the system automatically sets the result of baud rate correction to the RLN3nLBRP01 register on successful reception of the sync field.

22.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 04$_{H}$

Value after reset: 00$_{H}$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.45 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing A7$_{H}$, 58$_{H}$, and 01$_{H}$ successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0$_{B}$ (LIN reset mode).

Writing A7$_{H}$, 58$_{H}$, and 01$_{H}$ successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the module is placed in LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return "000000$_{B}$", and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7$_{H}$, 58$_{H}$, and 01$_{H}$.

22.3.3.4 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 22.46 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3 interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are used.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 1 0: LIN Slave mode (auto baud rate) 1 1: LIN Slave mode (fixed baud rate)

Configure the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 22.4, Interrupt Sources**.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use this module as an LIN slave, set these bits to “10_B” (auto baud rate) or “11_B” (fixed baud rate).

22.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LBLT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 22.47 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LBLT	Reception Break (Low-Level) Detection Width Setting 0: A break (low-level) is detected in 9.5 or 10 Tbits 1: A break (low-level) is detected in 10.5 or 11 Tbits

Configure the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is “10_B” (LIN slave mode (auto baud rate))

  0: Low-level width of 10 Tbits or longer is detected.

  1: Low-level width of 11 Tbits or longer is detected.

- When RLN3nLMD.LMD is “11_B” (LIN slave mode (fixed baud rate))

  0: Low-level width of 9.5 Tbits or longer is detected.

  1: Low-level width of 10.5 Tbits or longer is detected.

22.3.3.6 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 22.48 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Response Space Setting b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This register is enabled only during response transmission, and disabled during response reception.

Some combinations of the specified values result in a frame or response length exceeding the timeout time. Specify the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the inter-byte space width of the response transmission.
0 Tbit to 3 Tbits can be set.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the transmission width of the response space.
0 Tbit to 7 Tbits can be set.

22.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 22.49 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low-level width of the wake-up frame transmission.
1 Tbit to 16 Tbits can be set.

22.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 22.50 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Request Enable 0: Disables successful header reception interrupt request. 1: Enables successful header reception interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Request Enable 0: Disables successful Response/wake-up reception interrupt request. 1: Enables successful Response/wake-up reception interrupt request.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Request Enable 0: Disables successful Response/wake-up transmission interrupt request. 1: Enables successful Response/wake-up transmission interrupt request.

Configure the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Reception Interrupt Request Enable)

The SHIE bit enables or disables an interrupt request upon successful reception of a header.
With 0 set, the interrupt request for RLIN3n successful reception is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n successful reception is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Error types that are interrupt sources are the bit error, frame/response timeout error, framing error, sync filed error, ID parity error, checksum error, and response preparation error.
Detection of the bit error, frame/response timeout error, sync filed error, ID parity error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Response/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (input signal low-level width count).
With 0 set, the interrupt request for RLIN3n successful reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n successful reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Response/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a response or a wake-up frame.
With 0 set, the interrupt request for RLIN3n successful transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n successful transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

22.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 0D$_{\text{H}}$

Value after reset: 00$_{\text{H}}$

Bit	7	6	5	4	3	2	1	0
	LTES	IPERE	—	SFERE	FERE	TERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Table 22.51 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6	IPERE	ID Parity Error Detection Enable 0: Disables ID Parity error detection. 1: Enables ID Parity error detection.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SFERE	Sync Field Error Detection Enable 0: Disables Sync Field error detection. 1: Enables Sync Field error detection.
3	FERE	Framing Error Detection Enable*1 0: Disables framing error detection. 1: Enables framing error detection.
2	TERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable*1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0$_{\text{B}}$ (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details on the timeout error, see **Section 22.7.7, Error Statuses**.

IPERE Bit (ID Parity Error Detection Enable)

This bit enables or disables detection of the ID parity error.
With 0 set, the ID parity error is not detected.
With 1 set, the ID parity error is detected.
When this bit is set to 1, the detection result is reflected in the IPER flag of the RLN3nLEST register.
For details on the ID parity error, see **Section 22.7.7, Error Statuses**.

SFERE Bit (Sync Field Error Detection Enable)

This bit enables or disables detection of the sync field error.
With 0 set, the sync field error is not detected.
With 1 set, the sync field error is detected.
Regardless of the setting of this bit, when a sync field error is detected, this module waits for the next header.
When this bit is set to 1, the detection result is reflected in the SFER flag of the RLN3nLEST register.
For details on the sync filed error, see **Section 22.7.7, Error Statuses**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 22.7.7, Error Statuses**.

TERE Bit (Timeout Error Detection Enable)

The TERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is reflected in the TER flag of the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
The timeout error should not be used in LIN slave mode [auto baud rate] (when the LMD[1:0] bits in the RLN3nLMD register are “10_B”).
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 22.7.7, Error Statuses**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 22.7.7, Error Statuses**.

22.3.3.10 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.52 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode 1: LIN operation mode
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific LIN operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN wake-up mode.
With 1 set, LIN/UART interface enters LIN operation mode.
This bit is enabled only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN reset mode.
With 1 set, LIN reset mode of LIN/UART interface is canceled.

22.3.3.11 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	LNRR	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 22.53 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LNRR	No LIN Response Request 0: Response for the reception ID 1: No response for the reception ID
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped. 1: Response transmission/reception is started.
0	FTS	LIN Communication Start 0: Header reception/wake-up transmission/reception is stopped. 1: Header reception/wake-up transmission/reception is started.

LNRR Bit (No LIN Response Request)

Set this bit to 1 if no response is to be transmitted/received after receiving the header and checking the received ID. Once set, this bit is automatically cleared to 0 upon detection of new sync field or transition to LIN reset mode. Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 04_H using the store instruction.
Do not set this bit and the RTS bit to 1 simultaneously.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, use this bit only after the completion of the header. (Do not use this bit for the second or later data group.)

RTS Bit (Response Transmission/Reception Start)

Set this bit to 1 to start response transmission or reception after receiving the header and checking the received ID.
Once set, this bit is automatically cleared to 0 upon completion of response transmission or reception (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Do not set this bit and the LNRR bit to 1 simultaneously
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

FTS Bit (LIN Communication Start)

Set this bit to 1 to start header reception or wake-up transmission/reception.
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

22.3.3.12 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> +11$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.54 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

22.3.3.13 RLN3nLST — LIN Status Register

Access:	This register can be read or written in 8-bit units.
Address:	<RLIN3n_base> + 12_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 22.55 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header reception has not been completed. 1: Header reception has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Response/Wake-up Reception Flag 0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.
0	FTC	Successful Response/Wake-up Transmission Flag 0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, writing a value to this register is disabled. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Reception Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value before 1 is written.
The HTRC flag is set to 1 upon completion of header reception. Here, an interrupt request for RLIN3n successful reception is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if header reception is completed while this bit is 1, an interrupt is not generated. To clear this bit to 0, write 0 to the bit.
To detect a new header in the response field upon completion of header reception, clear this bit after it is set to 1.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value it had before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if an error is detected while this bit is 1, an interrupt is not generated. To clear the bit to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLN3nLEST register. This clears the ERR flag to 0.

FRC Flag (Successful Response/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value it had before 1 is written.
The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt request for RLIN3n successful reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response reception or wake-up reception is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Response/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value it had before 1 is written.
The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response transmission or wake-up transmission is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

22.3.3.14 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	IPER	CSER	SFER	FER	TER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 22.56 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	IPER	ID Parity Error Flag 0: ID parity error has not been detected. 1: ID parity error has been detected.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	SFER	Sync Field Error Flag 0: Sync field error has not been detected. 1: Sync field error has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	TER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value it had before 1 is written.
The RPER flag is set to 1 upon response preparation error detection. Write 0 to clear this bit.

IPER Flag (ID Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
When the IPERE bit of the RLN3nLEDE register is 1 (ID parity error detection enabled), this bit is set to 1 upon ID parity error detection. Write 0 to clear this bit.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value it had before 1 is written.
The CSER flag is set to 1 upon checksum error detection. Write 0 to clear this bit.

SFER Flag (Sync Field Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
When the SFERE bit of the RLN3nLEDE register is 1 (sync field error detection enabled), this bit is set to 1 upon sync field error detection. Write 0 to clear this bit.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. Write 0 to clear this bit.

TER Flag (Timeout Error Flag)

Only 0 can be written to the TER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the TERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), this flag is set to 1 upon frame timeout error or response timeout error detection. Write 0 to clear this bit.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. Write 0 to clear this bit.

22.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	—	LCS	RCDS	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.57 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception continues without waiting for reception of the next header.)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LCS	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RCDS	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

When multi-byte response transmission/reception function is not used, set it to “0”.

Set this bit when the RTS bit is 0 (response transmission/reception stopped).

LCS Bit (Checksum Select)

The LCS bit selects the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error detection is enabled (the TERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 22.7.7, Error Statuses**.
When the length of the response field data is 0 bytes (the RFDL bit is 0), do not set this bit to “1” (enhanced).
When response data of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.
During transmission or reception of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.
Set this bit when the RTS bit is 0 (response transmission/reception stopped).

RCDS Bit (Response Field Communication Direction Select)

This bit selects the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped) in LIN operation mode or when the FTS bit is 0 (header reception or wake-up transmission/reception stopped) in LIN wake-up mode).
When response data of 9 bytes or more is to be transmitted or received, do not change this bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
Set these bits when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped).
When response data of 9 bytes or more is to be transmitted or received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit in the RLN3nLDFC register is 1) include the checksum.

22.3.3.16 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.58 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value received in the ID field.

The value in the RLN3nLIDB register is enabled after the completion of header reception. In LIN mode (LIN operation mode, LIN wake-up mode), writing to this register is disabled.

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) received in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

When the IPERE bit in the RLN3nLEDE register is 1 (ID parity detection enabled), the received value and the value calculated internally are compared. If they do not match, IPER (ID parity error flag) is set.

ID[5:0] Bits (ID Setting)

The ID bits store the 6-bit ID value received in the ID field of the LIN frame.

22.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units.
However, in LIN self-test mode, this register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.59 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Stores the checksum value transmitted or received.

In LIN operation mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

22.3.3.18 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.60 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00_H to FF_H

- For response transmission:
  The RLN3nLDBRb registers set the data to be transmitted in the response field.
  These registers should be set when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stop).
- For response reception:
  The RLN3nLDBRb registers store the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

22.3.4 UART Related Registers

22.3.4.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 22.61 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 1 0 1: 6 samplings 0 1 1 0: 7 samplings 0 1 1 1: 8 samplings 1 0 0 0: 9 samplings 1 0 0 1: 10 samplings 1 0 1 0: 11 samplings 1 0 1 1: 12 samplings 1 1 0 0: 13 samplings 1 1 0 1: 14 samplings 1 1 1 0: 15 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In UART mode, it is possible to set the NSPB bits from 6 samplings to 16 samplings.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

22.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read or written in 16-bit units.

RLN3nLBRP0 register can be read or written in 8-bit units.

RLN3nLBRP1 register can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H

RLN3nLBRP0: <RLIN3n_base> + 02_H

RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.62 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Configure the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units using the registers RLN3nLBRP0 and RLN3nLBRP1.

22.3.4.3 RLN3nLMD — UART Mode Register

Access:	This register can be read or written in 8-bit units.
Address:	<RLIN3n_base> + 08_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	—	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R/W	R/W

Table 22.63 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 1: UART mode

Configure the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an UART, set these bits to 01_B.

22.3.4.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS	UPS[1:0]		USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.64 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmission data normal output 1: Transmission data with inverted output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data with inverted output
4, 3	UPS[1:0]	UART Parity Select 00: Parity disabled 01: Even parity 10: 0 Parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit:1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB First 1: MSB First
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Configure the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.
With 0 set, transmit data is output without inversion.
With 1 set, inverted transmit data is output.
The setting of this bit is valid in all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

This bit specifies the input polarity for UART communication.
With 0 set, received data is input without inversion.
With 1 set, received data is input with inversion.
The setting of this bit is valid in all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of UTPS bit.

When setting this bit to “1” and expansion bit reception ((with expansion bit comparison) or (with data comparison)) is performed, set the inverse of the expected value to the UEBDL bit in the RLN3nLUOR1 register and RLN3nLIDB register to enable comparison of the inverted values of the received values.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to “00_B”, data is communicated without the parity.

  [Transmission]
  A parity bit is not added to transmit data.

  [Reception]
  Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01_B”, data is communicated with the even parity.

  [Transmission]
  If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

  [Reception]
  If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10_B”, data is communicated with 0 parity.

  [Transmission]
  Regardless of the number of 1s in transmit data, “0” is added to the parity bit.

  [Reception]
  The value of the parity bit is not evaluated. Therefore, no parity error occurs.

- When these bits are set to “11_B”, data is communicated with the odd parity.

  [Transmission]
  If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

  [Reception]
  If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.
With 0 set, stop bit length of 1 bit is selected.
With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.
With 0 set, LSB first is selected.
With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.
With 0 set, the character length is 8 bits.
With 1 set, the character length is 7 bits.
When the character length of one frame is 9 bits (the UEBE bit in the RLN3nLUOR1 register is 1), the setting of this bit is ignored.

22.3.4.5 RLN3nLSC — UART Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 22.65 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frames when transmitting data from the UART buffer. 0 to 3 Tbits can be set.

Set IBS[1:0] bits to “00_B” when UART buffer is not used.

When data is transferred from the UART transmission data register (RLN3nLUTDR) and the UART wait transmission data register (RLN3nLUWTDR), the setting of these bits is ignored.Set these bits to “00_B”.

22.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Table 22.66 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is reflected in the FER flag of the RLN3nLEST register.
For details on the framing error, see **Section 22.8.5, Error Statuses**.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.
With 0 set, the overrun error is not detected.
With 1 set, the overrun error is detected.
When this bit is set to 1, the detection result is reflected in the OER flag of the RLN3nLEST register.
For details on the overrun error, see **Section 22.8.5, Error Statuses**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is reflected in the BER flag of the RLN3nLEST register.
In full-duplex communication, do not set this bit to “1”.
Do not set this register when the NSPB bits in the RLN3nLWBR register are 0101_B (6 samplings) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filtering is enabled).

For details on the bit error, see **Section 22.8.5, Error Statuses**.

22.3.4.7 RLN3nLCUC — UART Control Register

Access: This register can be read or written in 8-bit units.
Address: <RLIN3n_base> + $0E_H$
Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 22.67 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is reflected in the RLN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.
With 0 set, LIN reset mode.
With 1 set, LIN reset mode is canceled.

22.3.4.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R

Table 22.68 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.

Only 1 can be written to this bit; 0 cannot be written.

Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enable) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).

Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. This bit is also automatically cleared to 0 upon transition to LIN reset mode.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

When writing 1 to this bit while the UTSW bit in the RLN3nLDFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the reception of stop bit is completed), write only during the reception of stop bit.

22.3.4.9 RLN3nLMST — UART Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.69 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

22.3.4.10 RLN3nLST — UART Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 22.70 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	URS	Reception Status Flag 0: Reception is stopped. 1: Reception is started.
4	UTS	Transmission Status Flag 0: Transmission is stopped. 1: Transmission is started.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to “00_H” upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains “00_H”. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

URS Flag (Reception Status Flag)

At the start of the reception, this flag is set to 1.
The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.
The reception is ended under the following conditions.

- Sampling point of the first bit of the stop bits

UTS Flag (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.
The transmission is started under the following conditions.

- When transmission data is specified in the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the completion of transmission.
The transmission is ended under the following conditions.

- When transmission of data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not specified
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or when ID's match (when at least one of the flags of the RLN3nLEST register is 1). Here, an interrupt request for RLIN3n status is generated. However, if an error or expansion bit is detected or ID's match while this bit is 1, an interrupt is not generated. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

FTC Flag (Successful UART Buffer Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written.
Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLN3nLDFC register from the UART buffer. Here, an interrupt request for RLIN3n transmission is generated.
Write 0 to clear this flag.

22.3.4.11 RLN3nLEST — UART Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 22.71 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Match Flag 0: The received data does not match the ID value. 1: The received data matches the ID value.
4	EXBT	Expansion Bit Detection Flag 0: Expansion bit has not been detected. 1: Expansion bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is retained. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

IDMT Flag (ID Match Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.

The IDMT flag is set to 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)
- The UEBDCE bit in the RLN3nLUOR1 register is 1 (expansion bit/data comparison enabled)
  - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register match.
  - The 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

EXBT Flag (Expansion Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches with the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.
Write 0 to clear this flag.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). Write 0 to clear this flag.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). Write 0 to clear this flag.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.
The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).
Write 0 to clear this flag.

22.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	UTSW	—	MDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 22.72 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MDL[3:0]	UART Buffer Data Length Select b3 b0 0 0 0 0: 9 data 0 0 0 1: 1 data 0 0 1 0: 2 data 0 0 1 1: 3 data 0 1 0 0: 4 data 0 1 0 1: 5 data 0 1 1 0: 6 data 0 1 1 1: 7 data 1 0 0 0: 8 data 1 0 0 1: 9 data Settings other than the above are prohibited.

UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.
With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.
With 1 set, transmission is started after the completion of the stop bit reception.
Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.
This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).
Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.
Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

22.3.4.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.73 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	Specifies the ID value that is referred in expansion bit data comparison

ID Bit (ID Bit)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (reception is stopped).

22.3.4.14 RLN3nLUDB0 — UART Data Buffer 0 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 17_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	UDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.74 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H

If the data length selection corresponds to 9 data bytes (RLN3nLDFC.MDL bit is “0_H” or “9_H”) for multi-byte UART transmission, then the first data value for UART communication is present in this buffer.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 22.75, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 22.8.1.2, UART Buffer Transmission**, **22.8.1.2(1), UART Buffer Transmission**.

Table 22.75 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—*1	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

Note 1. In the case of 7-bit data length, write the value after reset.

22.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H

RLN3nLDBR2: <RLIN3n_base> + 19_H

RLN3nLDBR3: <RLIN3n_base> + $1A_H$

RLN3nLDBR4: <RLIN3n_base> + $1B_H$

RLN3nLDBR5: <RLIN3n_base> + $1C_H$

RLN3nLDBR6: <RLIN3n_base> + $1D_H$

RLN3nLDBR7: <RLIN3n_base> + $1E_H$

RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.76 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Specifies the data to be transmitted. Setting range: 00_H to FF_H

These registers specify the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 22.77, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 22.8.1.2, UART Buffer Transmission**, **22.8.1.2(1), UART Buffer Transmission**.

Table 22.77 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

	RLN3nLDBRb							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—*1	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

Note 1. In the case of 7-bit data length, write the value after reset.

22.3.4.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UROE	UTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.78 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	UROE	Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.
In LIN reset mode, this register cannot be written to.
In LIN reset mode, the register retains 00_H.

UROE Bit (Reception Enable)

The UROE bit enables or disables reception.
With 0 set, reception is disabled.
With 1 set, reception is enabled.
Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. Note that transmission is also aborted.

Do not set this bit to 1 when data transmission from the UART buffer is in progress.

UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.
With 0 set, transmission is disabled.
With 1 set, transmission is enabled.
Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. Note that reception is also aborted.

22.3.4.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 21_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22.79 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison.
3	UTIGTS	Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).

With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.

With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

Do not set this bit to 1 when the UEBDCE bit is 1 (expansion bit/data comparison enable).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit selects the generation timing of the transmission interrupt.
With 0 set, the transmission interrupt is generated at the start of transmission.
With 1 set, the transmission interrupt is generated at the completion of transmission.
When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register).
When transmission from the UART buffer is performed with 1 set, the transmission interrupt is generated only at the completion of the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register).

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.
With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.
With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).
Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).
Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit selects the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (expansion bit comparison enabled).
With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.
With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.
With 0 set, expansion bit operation is disabled.
With 1 set, expansion bit operation is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

22.3.4.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read or written in 16-bit units.
RLN3nLUTDRL register can be read or written in 8-bit units.
RLN3nLUTDRH register can be read or written in 8-bit units.

Address: RLN3nLUTDR: <RLIN3n_base> + 24_H
RLN3nLUTDRL: <RLIN3n_base> + 24_H
RLN3nLUTDRH: <RLIN3n_base> + 25_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.80 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UTD[8:0]	Specifies the data to be transmitted. Setting range: 000_H to $1FF_H$

The RLN3nLUTDR register specifies the data to be transmitted from the transmission data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 22.81 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	—	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
9-bit; LSB first	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8

22.3.4.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register is a read-only register that can be read in 16-bit units.

RLN3nLURDRL register is a read-only register that can be read in 8-bit units.

RLN3nLURDRH register is a read-only register that can be read in 8-bit units.

Address: RLN3nLURDR: <RLIN3n_base> + 26_H

RLN3nLURDRL: <RLIN3n_base> + 26_H

RLN3nLURDRH: <RLIN3n_base> + 27_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	URD [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.82 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8 to 0	URD [8:0]	Store the received data

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the received data is stored in this register and can be read out.

This register is updated upon reception of stop bit in the received data.

This register is also updated even when an error is caused by the parity or stop bit.

However, the value of this register is not updated if an overrun error occurs when the OERE bit of the RLN3nLEDE register is 1 (overrun detection enabled). The value of this register is updated even if an overrun error occurs when the OERE bit is 0 (overrun detection disabled).

Read this register upon occurrence of a reception error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8-bit units.

However, when expansion bit is used (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 22.83 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	—	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
9-bit; LSB first	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8

22.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access:	RLN3nLUWTDR register can be read or written in 16-bit units. RLN3nLUWTDRL register can be read or written in 8-bit units. RLN3nLUWTDRH register can be read or written in 8-bit units.
Address:	RLN3nLUWTDR: <RLIN3n_base> + 28_H RLN3nLUWTDRL: <RLIN3n_base> + 28_H RLN3nLUWTDRH: <RLIN3n_base> + 29_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UWTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.84 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000_H to $1FF_H$

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

Also, write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the value of the RLN3nLUTDR register is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 22.85 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	—	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
9-bit; LSB first	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8

22.4 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt
- RLIN3n interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the RLIN3n interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt request.

Table 22.86, Interrupt Sources lists the sources for each interrupt.

Table 22.86 Interrupt Sources

		LIOS Bit in RLN3nLMD Register is 0	LIOS Bit in RLN3nLMD Register is 1*1		
		RLIN3n Interrupt	RLIN3n Transmission Interrupt	RLIN3n Successful Reception Interrupt	RLIN3n Status Interrupt
LIN mode	LIN master mode	• Successful frame transmission • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful frame reception • Successful wake-up reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
	LIN slave mode	• Successful response transmission • Successful response reception • Successful wake-up transmission • Successful wake-up reception • Successful header reception • Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error	• Successful response transmission • Successful wake-up transmission	• Successful response reception • Successful wake-up reception • Successful header reception	• Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error
UART mode		—	• Transmission start/successful transmission	• Successful reception • Expansion bit mismatch	• Bit error • Overrun error • Framing error • Expansion bit match • ID match • Parity error

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is set to 1.

22.5 Modes

The LIN/UART interface provides the following four modes, depending on the specific function to be performed:

- LIN reset mode
- LIN mode
  - LIN master mode
  - LIN slave mode [auto baud rate]
  - LIN slave mode [fixed baud rate]
- UART mode
- LIN self-test mode

Figure 22.2, Mode Transitions shows mode transitions. **Table 22.87, Transition Condition of Each Mode** describes mode transition conditions. **Table 22.88, Operations Available in Each Mode** lists operations available in each mode.

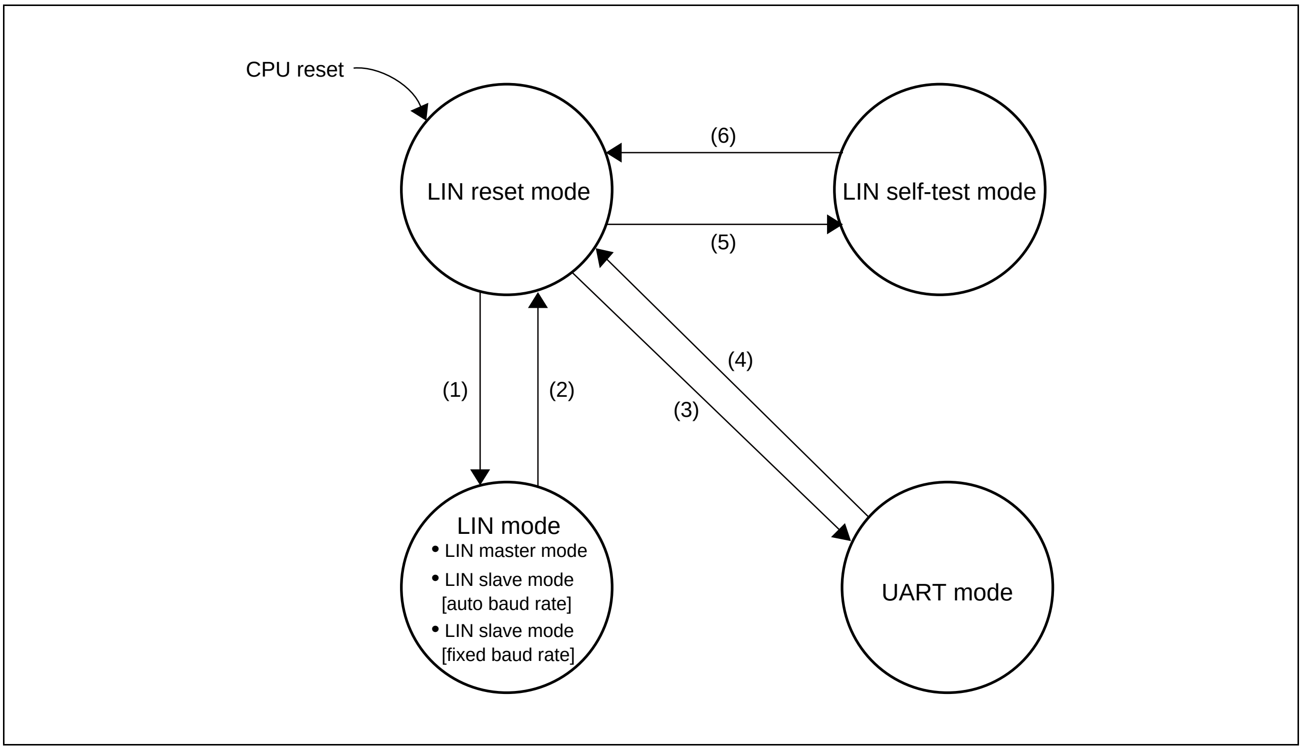


Figure 22.2 Mode Transitions

Table 22.87 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode	
			- LIN master mode	- LMD bits in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			- LIN slave mode [auto baud rate]	- LMD bits in RLN3nLMD register = 11_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			- LIN slave mode [fixed baud rate]	- LMD bits in RLN3nLMD register = 10_B and OM1 and OM0 bits of RLN3nLCUC register = 01_B or 11_B
(2)	LIN mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(3)	LIN reset mode	→	UART mode	LMD bits in RLN3nLMD register = 01_B and OM0 bit in RLN3nLCUC register = 1_B
(4)	UART mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(5)	LIN reset mode	→	LIN self-test mode	See **Section 22.9, LIN Self-Test Mode**.
(6)	LIN self-test mode	→	LIN reset mode	See **Section 22.9, LIN Self-Test Mode**.

Table 22.88 Operations Available in Each Mode

LIN Mode			
LIN Master Mode	LIN Slave Mode [auto baud rate] LIN Slave Mode [fixed baud rate]	UART Mode	LIN Self-Test Mode
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Header reception Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception Error detection	Self-test

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 22.9, LIN Self-Test Mode**.

22.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The transition to LIN reset mode can be verified by checking that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode transitions to LIN reset mode, the following registers are initialized to their reset values and they retain their initial values while in LIN reset mode:

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

22.7 LIN Mode

LIN mode can operate in the following submodes: LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B causes the transition to LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.

The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. The baud rate can be set to 1 kbps to 20 kbps.

Set the LPRS[2:0] bits in the RLN3nLWBR register so that the prescaler clock (the clock obtained by dividing the frequency of the LIN communication clock source by the prescaler) is configured as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4 MHz*[1]
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4 MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to “0011_B” (4 samplings).

LIN slave mode [fixed baud rate] allows automatic detection of the break field, the sync field, and the ID field at a baud rate that is preset by the baud rate generator.

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 10_B (LIN slave mode [auto baud rate]) and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B causes the transition to LIN slave mode [auto baud rate]; and setting the LMD bits in the RLN3nLMD register to 11_B (LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B causes the transition to LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B or 11_B.

When transitioning form one submode to another submode within LIN mode, transition to LIN reset mode first and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operating modes:

- LIN operation mode
- LIN wake-up mode

Figure 22.3, Transition of Operating Modes shows the transition of operating modes. **Table 22.89, Transition Conditions for Operating Modes** describes the transition conditions of operating modes.

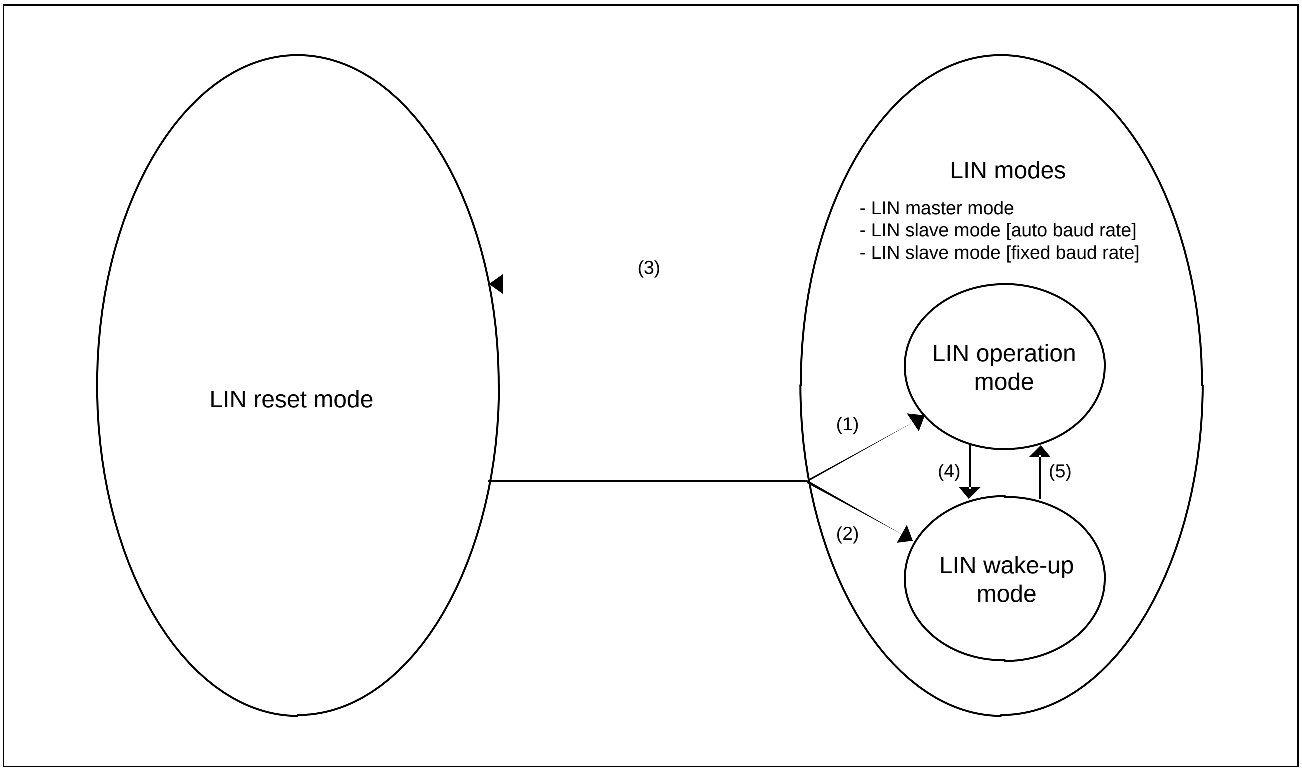


Figure 22.3 Transition of Operating Modes

Table 22.89 Transition Conditions for Operating Modes

	Operation Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode ● LIN operation mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 11_B
(2)	LIN reset mode	→	LIN mode ● LIN wake-up mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B
(3)	LIN mode ● LIN operation mode ● LIN wake-up mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(4)*[1]	LIN mode ● LIN operation mode	→	LIN mode ● LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = 01_B
(5)*[1]	LIN mode ● LIN wake-up mode	→	LIN mode ● LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = 11_B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

While in LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B switches the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be configured after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B switches the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be configured after the OMM1 and OMM0 bits have become 01_B.

22.7.1 LIN Master Mode

22.7.1.1 Header Transmission

Figure 22.4, Operation in Header Transmission shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 22.90, Processing in Header Transmission** provides processing in header transmission.

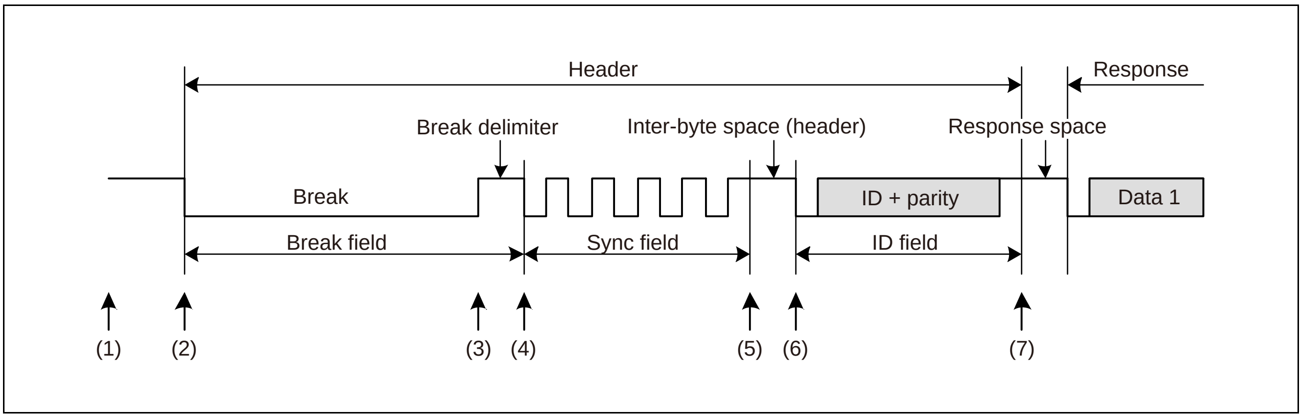


Figure 22.4 Operation in Header Transmission

Table 22.90 Processing in Header Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Transitions to LIN master mode: LIN operation mode • Configures frame information to be transmitted (ID, parity, data length, response direction, Checksum method, and transmission data)	Waits for the FTS bit in the RLN3nLTRC register to be set by software (idle)
(2)	Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55_H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets the successful header transmission flag.

NOTE

For information about error detection conditions, see **Section 22.7.7, Error Statuses**.

22.7.1.2 Response Transmission

Figure 22.5, Operation in Response Transmission shows the operation of the LIN/UART interface (LIN master mode) in response transmission.
Table 22.91, Processing in Response Transmission provides processing in response transmission.

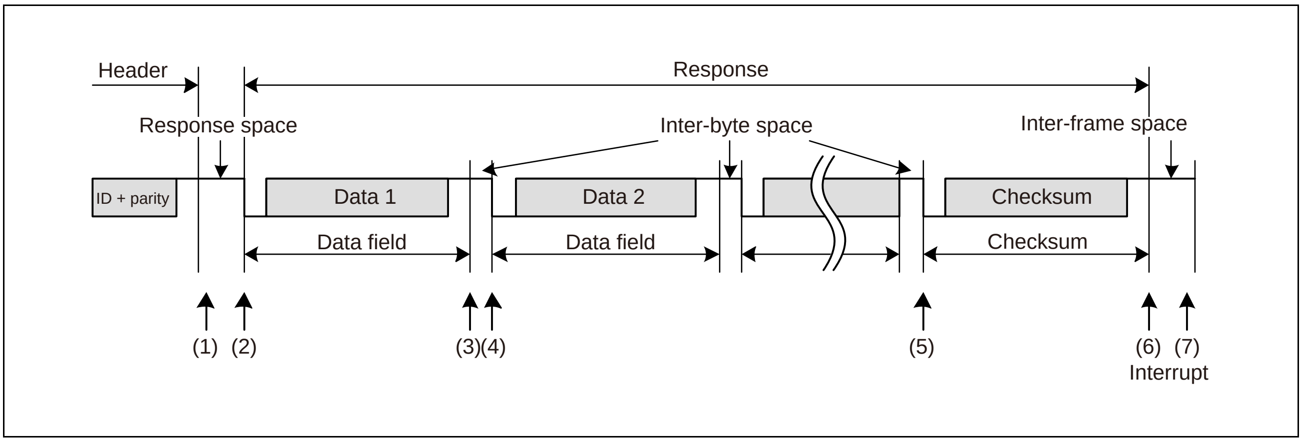


Figure 22.5 Operation in Response Transmission

Table 22.91 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	[When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started) [When not in frame separate mode] • Waits for an interrupt request	[When in frame separate mode] • Waits for the RTS bit in the RLN3nLTRC register to be set to 1 by software. • When the bit is set to 1, transmits a response space. [When not in frame separate mode] • Transmits a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets the successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception is stopped).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection conditions, see **Section 22.7.7, Error Statuses**.

22.7.1.3 Response Reception

Figure 22.6, Operation in Response Reception shows the operation of the LIN/UART interface (LIN master mode) in response reception. **Table 22.92, Processing in Response Reception** provides processing in response reception.

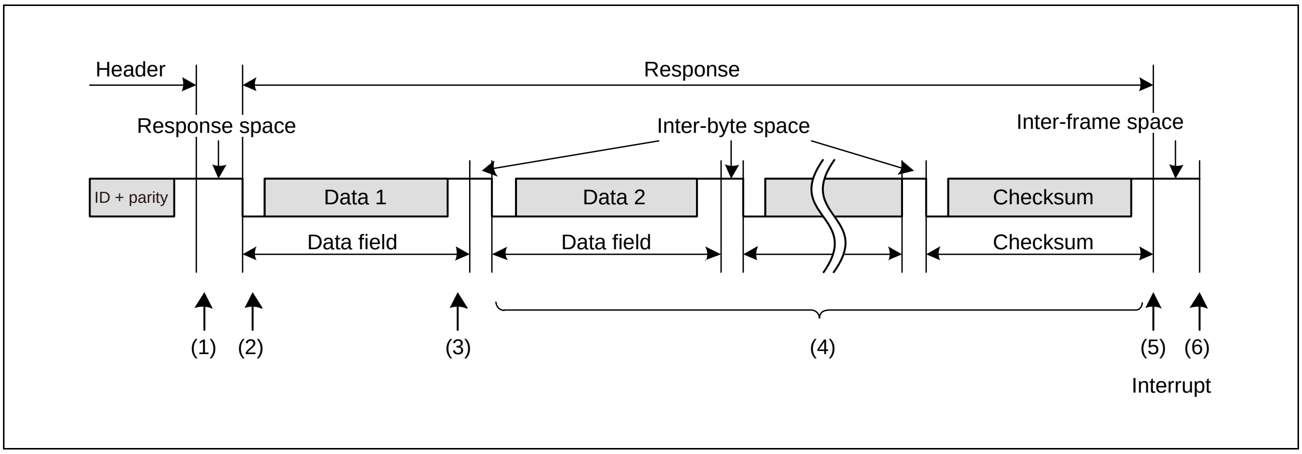


Figure 22.6 Operation in Response Reception

Table 22.92 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	Waits for an interrupt request (no processing)	Waits for detection of a start bit.
(2)		Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		● Receives data 2 when the start bit is detected. ● Receives data 3 when the start bit is detected. (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : ● Receives the checksum when the start bit is detected.
(5)		● Determines the checksum. ● Sets the successful frame/wake-up reception flag. ● Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	● Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 22.7.7, Error Statuses**.

22.7.2 LIN Slave Mode

22.7.2.1 Header Reception

Figure 22.7, Operation in Header Reception shows the operation of the LIN/UART interface (LIN slave mode) in header reception. **Table 22.93, Processing in Header Reception** provides processing in header reception.

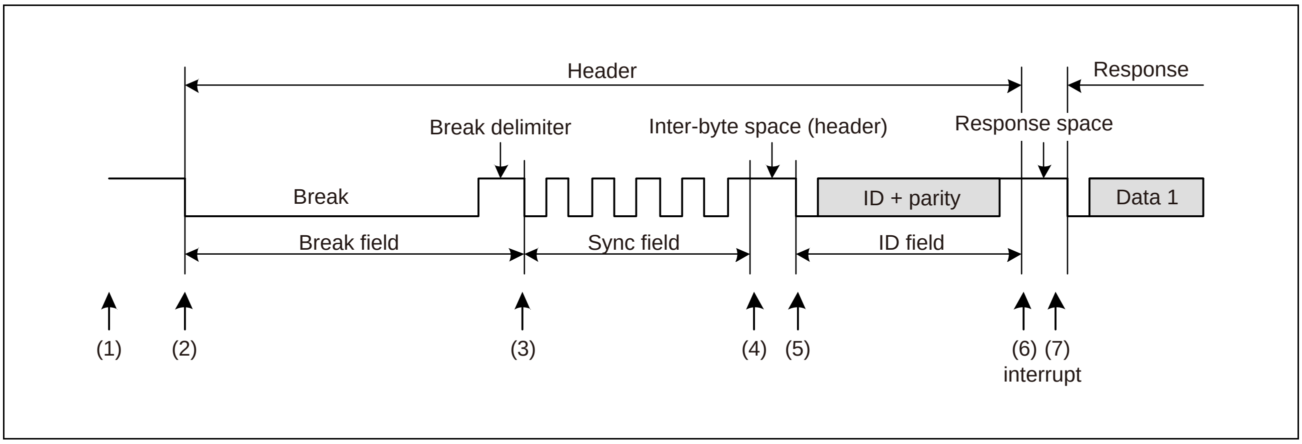


Figure 22.7 Operation in Header Reception

Table 22.93 Processing in Header Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Transitions to the LIN slave mode: LIN operation mode • Sets the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started)	Waits for the FTS bit in the RLN3nLTRC register to be set by software.
(2)	Waits for an interrupt request.	Waits for detection of break field
(3)		Detects a break field (LIN slave mode [fixed baud rate]. For details about the break field detection timing in the case of LIN slave mode [auto baud rate], see [Auto Baud Rate Correction Function].)
(4)		• Detects a sync field (55_H) • Sets the baud rate generator (in the case of LIN slave mode [auto baud rate]) • Clears the no-response request bit (LNRR bit).
(5)		• Receives an ID field. • Checks the ID parity bit
(6)		Sets the header reception complete flag.
(7)	• Checks the RLN3nLST register, and clears flags. • Checks the RLN3nLIDB register, and prepares a response.	• Completes a header reception process. • Waits for a response request.

NOTE

The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred. For information about error detection conditions, see **Section 22.7.7, Error Statuses**.

[Auto Baud Rate Correction Function]

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first "Low level" width is 10 times (if the BLT bit of the RLN3nLBFC register is "0") or 11 times (if the BLT bit of the RLN3nLBFC register is "1") or greater than the bit width calculated from the average of the starting 2 bits (the period of the consecutive falling edges from the beginning of the sync field) of the sync field, the system concludes that the detection of break field was successful and verifies that the data in the sync field is 55_H. If the data in the sync field is indeed 55_H and the system determines that sync field reception was successful, the system automatically sets the baud rate correction result to the RLN3nLBRP01 register.

If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.

On the other hand, if the data in the sync field is not 55_H and the system determines that sync field reception failed, the system sets the sync field error flag and an error interrupt is generated. In that case, baud rate correction is not performed and the LIN/UART interface waits for the detection of the next break field (low level).

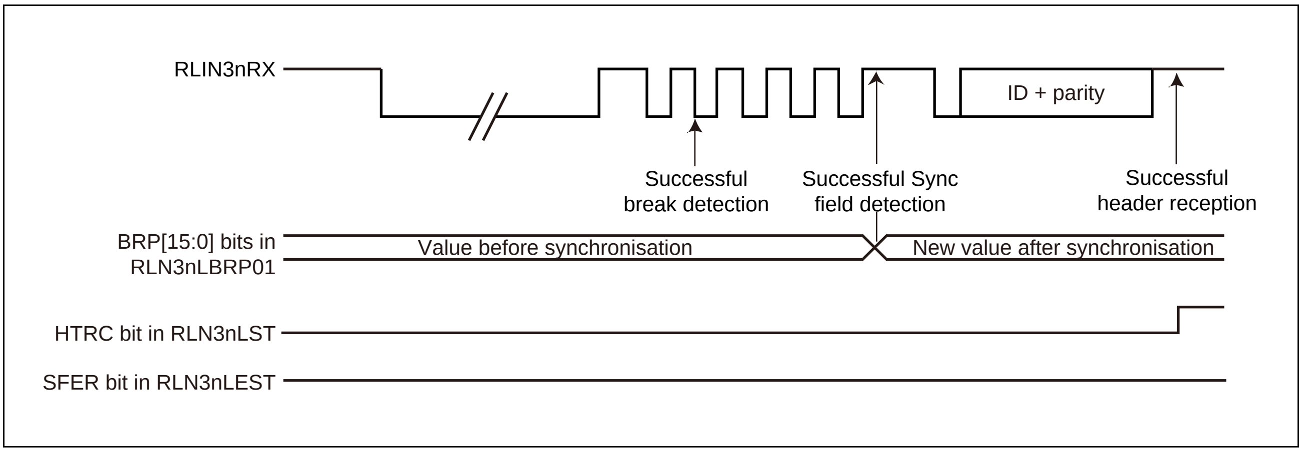


Figure 22.8 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

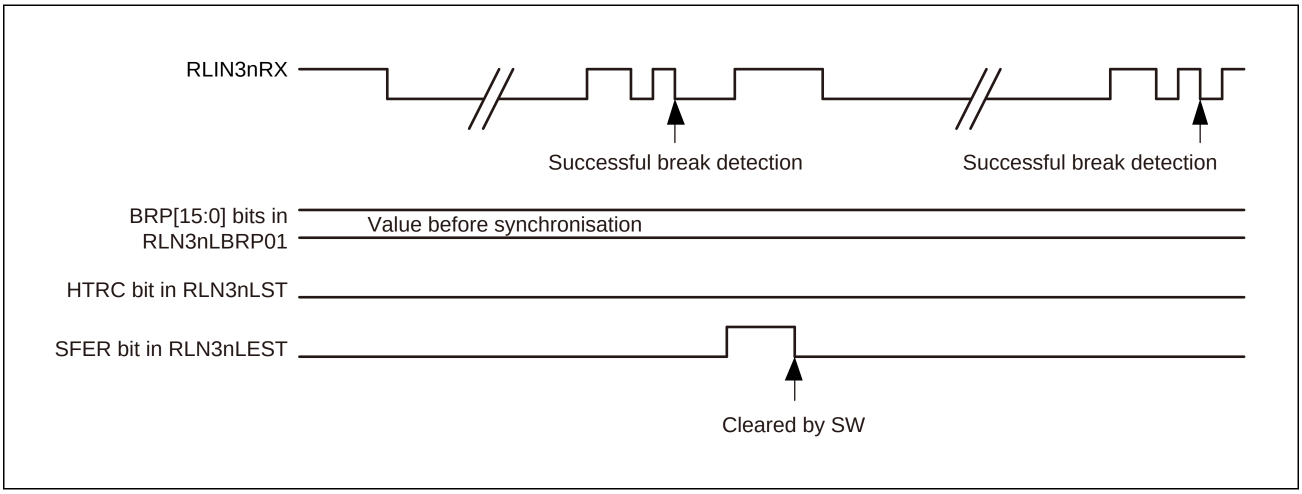


Figure 22.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

22.7.2.2 Response Transmission

Figure 22.10, Operation in Response Transmission shows the operation of the LIN/UART interface (LIN slave mode) in response transmission. **Table 22.94, Processing in Response Transmission** provides processing in response transmission.

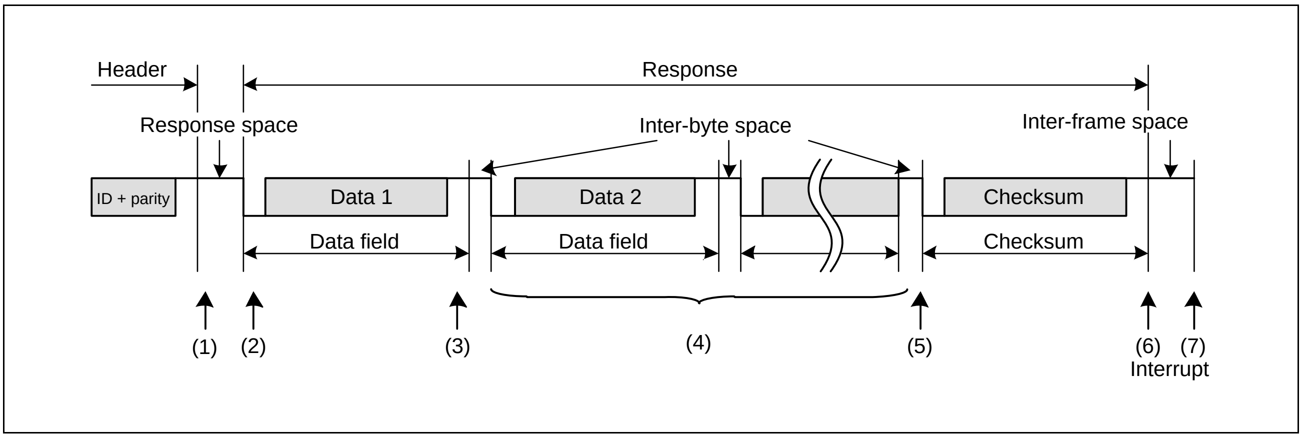


Figure 22.10 Operation in Response Transmission

Table 22.94 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Configures the RLN3nLDFC register. • Configures the RLN3nLDBRb registers.(b = 1 to 8) • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started)	• Waits for the RTS or LNRR bit in the RLN3nLTRC register to be set by software • Transmits the response space after the RTS bit of the RLN3nLTRC register is set to 1
(2)	Waits for an interrupt request.	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets the successful response/wake-up transmission flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	• Completes the response transmission process. • Waits for a new break.

NOTE

- For information about error detection, see **Section 22.7.7, Error Statuses**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred.

22.7.2.3 Response Reception

Figure 22.11, Operation in Response Reception shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 22.95, Processing in Response Reception** provides processing in response reception.

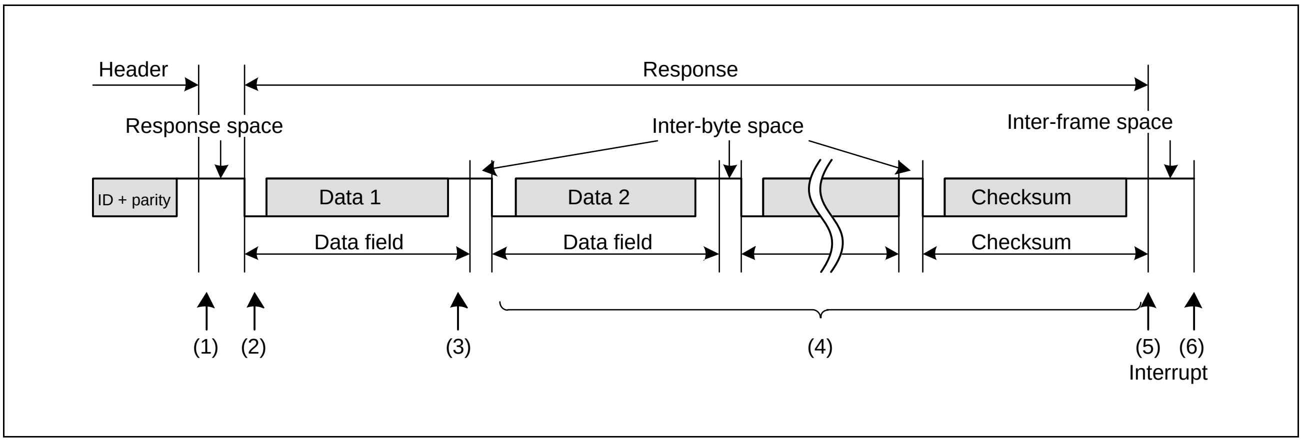


Figure 22.11 Operation in Response Reception

Table 22.95 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the RLN3nLDFC register. • Sets the response transmission/reception start bit (RTS bit) to 1.	• Waits for the RTS (response transmission/reception start) or LNRR (no-response request) bit to be set by software. • Waits for detection of the start bit.
(2)	Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful response/wake-up reception flag or error flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	• Completes the response process. • Waits for a new break.

NOTE

- For information about error detection conditions, see **Section 22.7.7, Error Statuses**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred.

22.7.2.4 No-response Request

Figure 22.12, Operation when No Response is Requested shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 22.96, Processing when No Response is Requested** shows the processing that occurs when no response is requested.

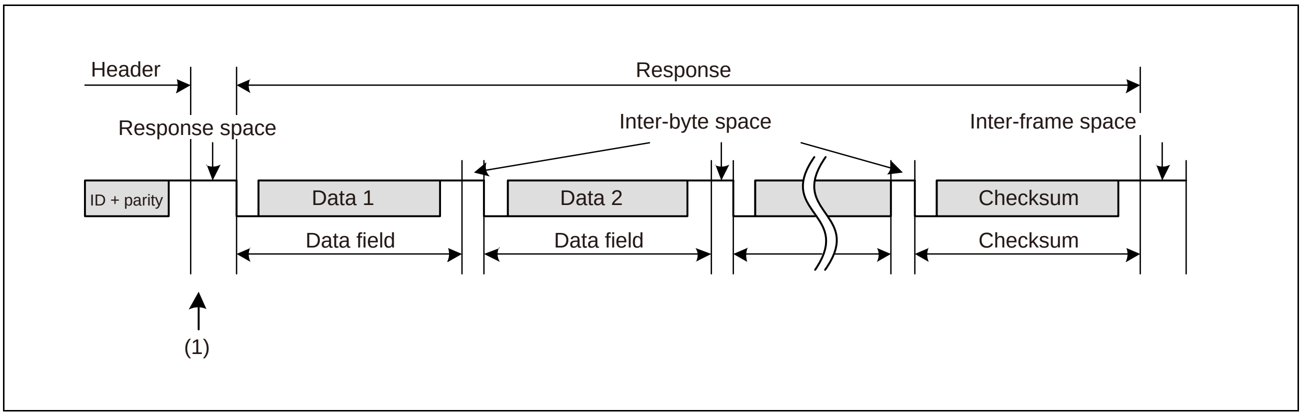


Figure 22.12 Operation when No Response is Requested

Table 22.96 Processing when No Response is Requested

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the no-response request bit (LNRR bit) to 1.	• Waits for the LNRR (no-response request) bit to be set by software • Completes the frame reception process • Waits for a new break

22.7.3 Data Transmission/Reception

22.7.3.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag of the RLN3nLEST register (see **Section 22.7.7, Error Statuses**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be $16f_{LIN}$, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

In LIN slave mode [auto baud rate], if 1 Tbit is generated to be $4f_{LIN}$, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be $8f_{LIN}$, the sampling point for received data is at the 7th clock cycle (87.5% position).

Figure 22.13, Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate]) shows an example of data transmission timing.

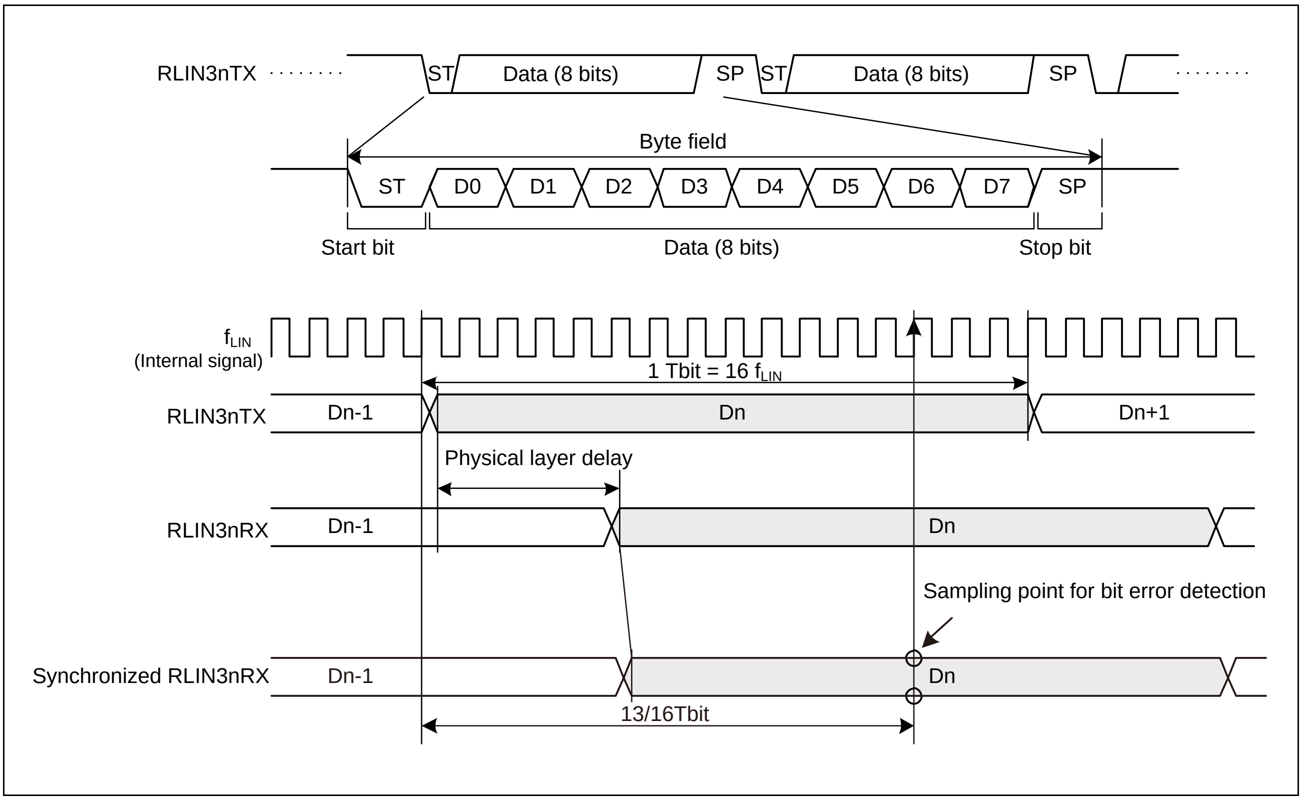


Figure 22.13 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

22.7.3.2 Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function with respect to reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

Figure 22.14, Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate]) shows an example of data reception timing.

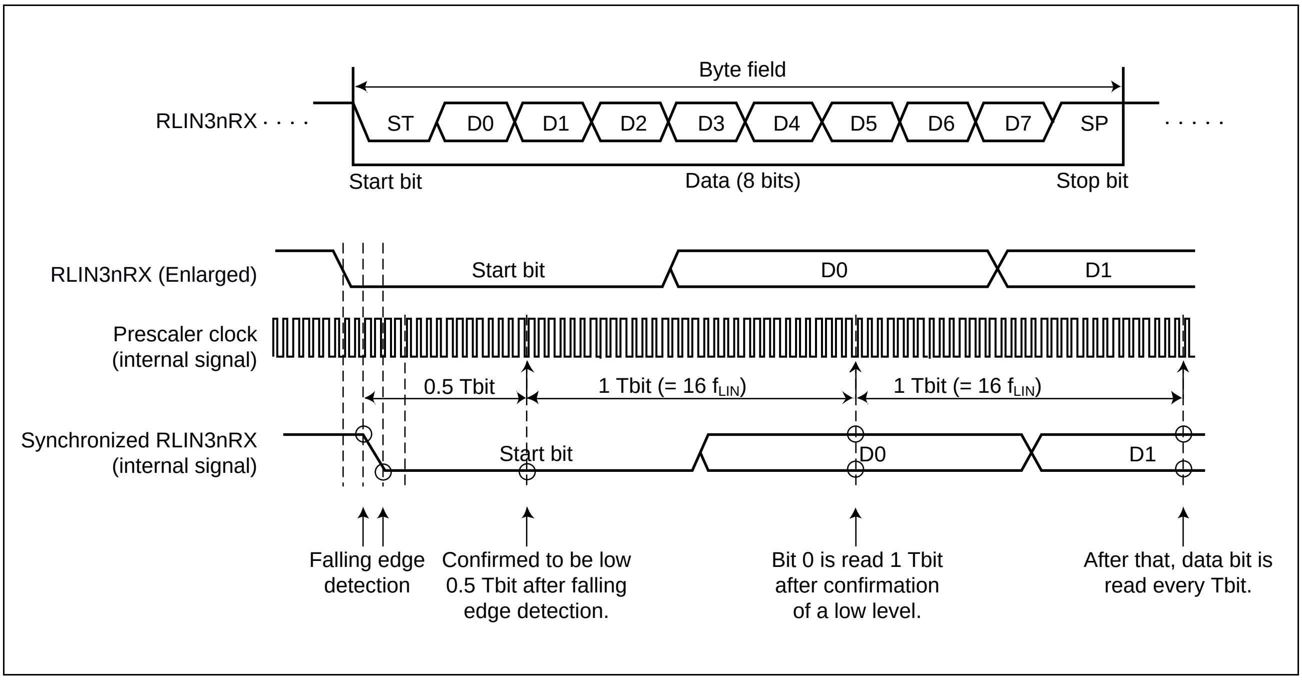


Figure 22.14 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

22.7.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface transmits or receives data continuously.

22.7.4.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data regions 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data regions 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 22.15, LIN Transmission Processing and Corresponding Buffers shows the LIN transmission processing and the corresponding buffers.

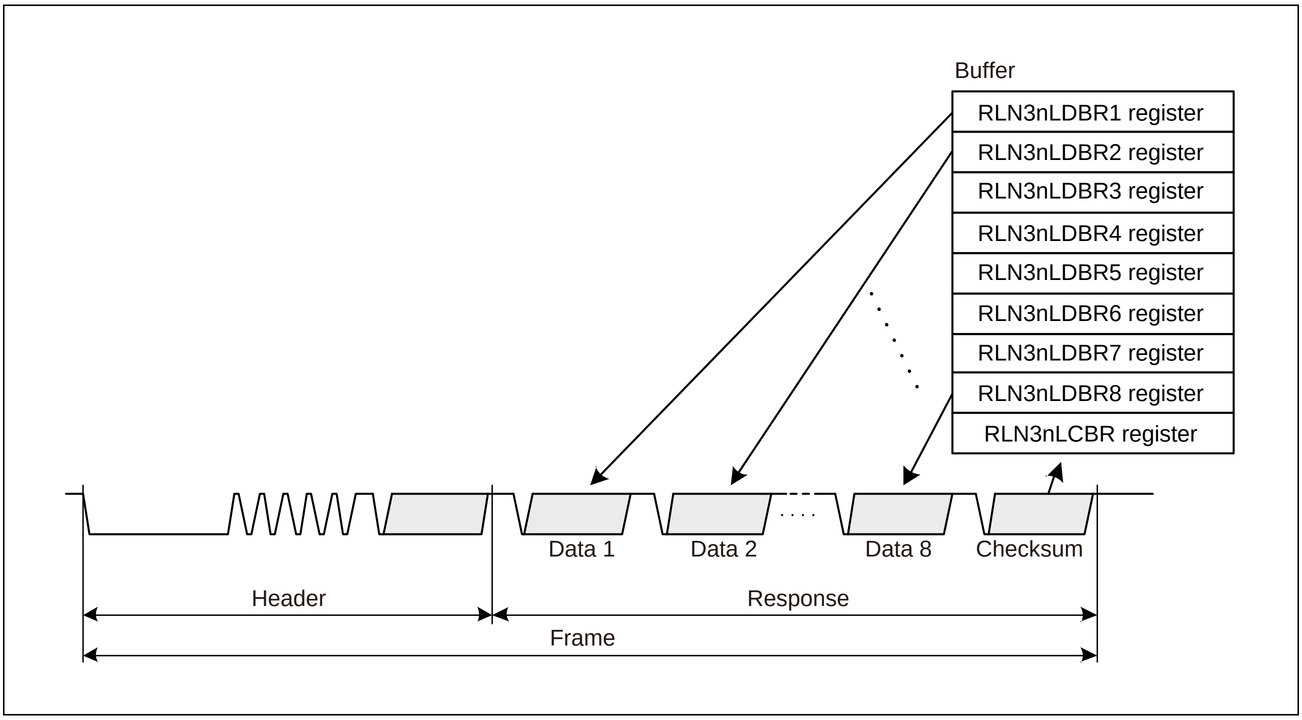


Figure 22.15 LIN Transmission Processing and Corresponding Buffers

[Frame Separate Mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, the header and response are separately transmitted when prompted by respective transmission start requests.

When the transmission of the header is completed, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or greater in LIN master mode.

22.7.4.2 Reception of LIN Frames

For an 8-byte reception, the contents of data regions 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data regions 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 22.16, LIN Reception Processing and Corresponding Buffers depicts the LIN reception processing and the corresponding buffers.

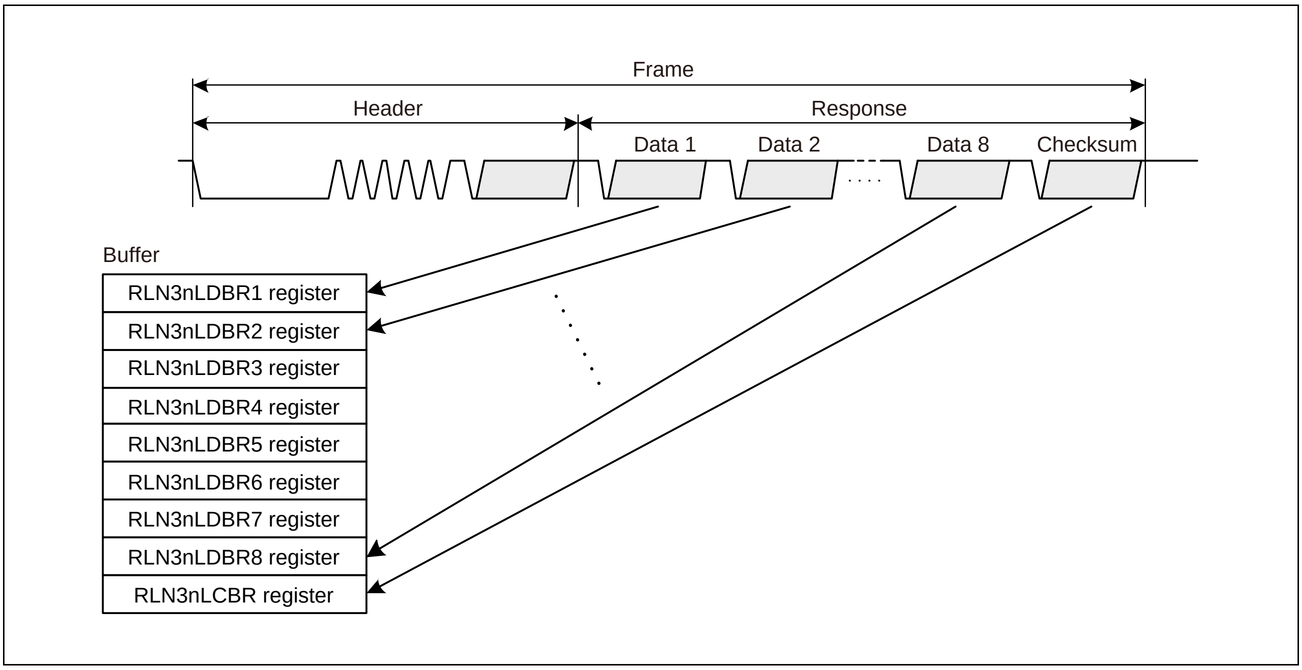


Figure 22.16 LIN Reception Processing and Corresponding Buffers

[Reception of Data 1]

When the reception of the first byte of data is completed, the D1RC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

22.7.4.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or greater can also be transmitted and received.

In such case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is greater than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the final data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the final data group) before transmitting or receiving the data group, and a checksum should be appended to the final data group.

By changing the RFDL bit setting in RLN3nLDFC register when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register to 1 (frame separate mode).

NOTE

In LIN slave mode, the LIN/UART interface can detect a new break field during the transmission or reception of a response.

22.7.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

22.7.5.1 Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission), or setting the RCDS bit in the RLN3nLDFC register to 1 (LIN slave mode: response transmission), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal is set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock (f_{LIN}) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level width is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame response or wake-up transmission) and when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wake-up transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is aborted and the BER flag of the RLN3nLEST register is set to 1 (bit error detected).
When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 22.17, Wake-up Transmission Timing shows the wake-up transmission timing.

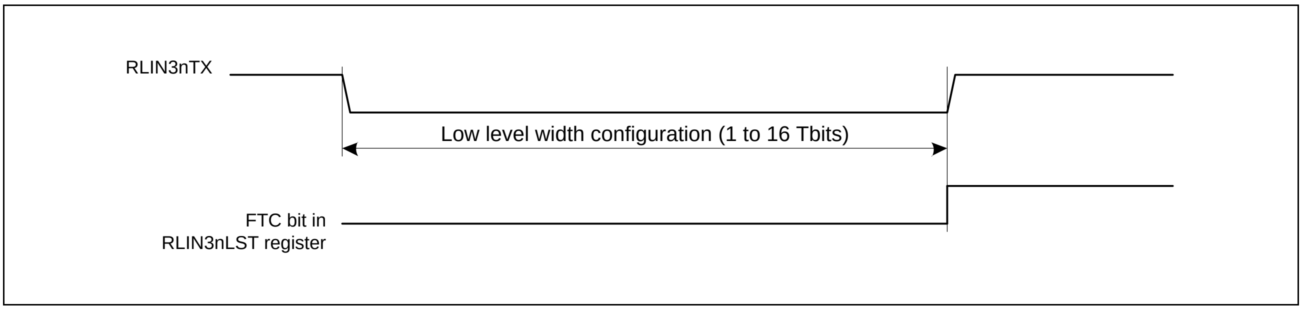


Figure 22.17 Wake-up Transmission Timing

22.7.5.2 Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of f_{LIN} to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation is executed without changing the baud rate generator setting at a transition between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects the LIN system clock (f_{LIN}) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200 bps while fa is selected, an input signal with a low-level width of 130 us or longer regardless of the setting of the LCKS bit in the RLN3nLMD register.

To used the wake-up reception function, in LIN wake-up mode, set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception) or the RCDS bit in the RLN3nLDFC register to 0 (LIN slave mode: response reception), and then set the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame response/wake-up reception) and if the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for RLIN3n successful reception is generated.

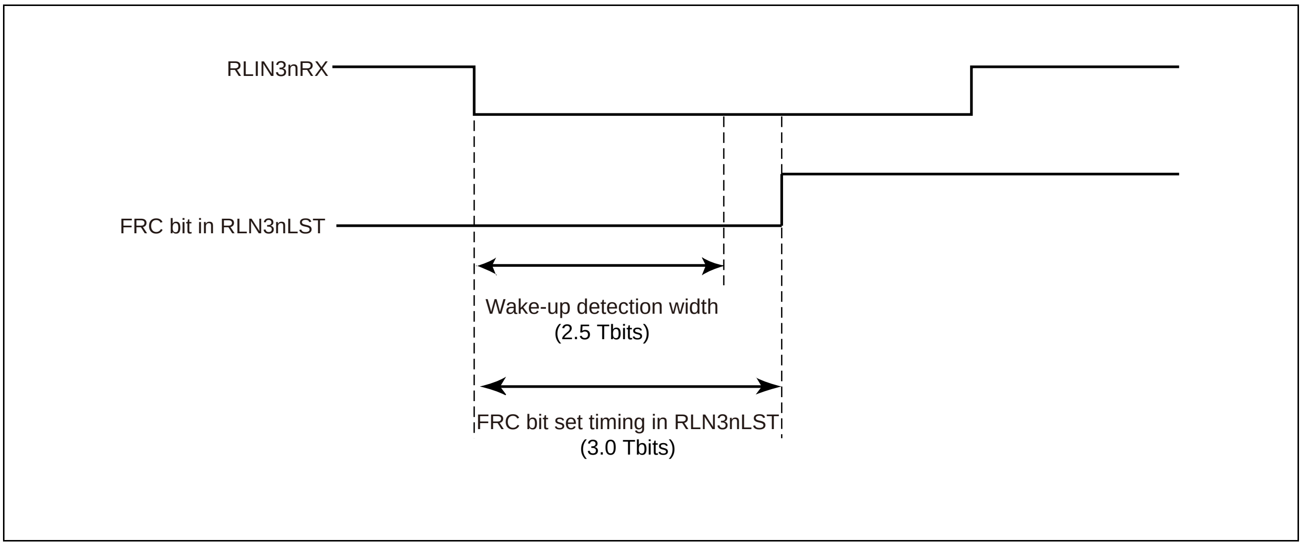


Figure 22.18 Input Signal Low level Count Function

22.7.5.3 Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, however the collision of wakeup signals is not detected in the LIN/UART interface.

22.7.6 Statuses

During LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, successful header transmission/reception, can generate interrupt requests.

Table 22.97, Types of Statuses in LIN Master Mode shows the types of statuses available in LIN master mode. **Table 22.98, Types of Statuses in LIN Slave Mode** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 22.97 Types of Statuses in LIN Master Mode

Status	Status Set Condition	Status Clear Condition	Operating Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if the LIN/UART interface actually enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if the LIN/UART interface actually enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Successful frame/wake-up transmission	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	• When the next communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	✓
Successful frame/wake-up reception	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	• When the next communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	✓
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When the next communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software*1 • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	✓
Successful data 1 reception	When the RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2	• When the next communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Successful header transmission	When a header field is transmitted successfully.	• When the next communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	✓

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

Table 22.98 Types of Statuses in LIN Slave Mode

Status	Status Set Condition	Status Clear Condition	Operating Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit of the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if the LIN/UART interface actually enters LIN operation mode.	● After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if the LIN/UART interface actually enters LIN wake-up mode.	● LIN operation mode ● LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Successful response/wake-up transmission	When a response field, a wake-up signal, or a data group is transmitted successfully.	● When cleared by software ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	FTC flag in RLN3nLST register	✓
Successful response/wake-up reception	When a response field, a wake-up signal, or a data group is received successfully.	● When cleared by software ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	FRC flag in RLN3nLST register	✓
Error detection	If any of the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag of the RLN3nLEST register is set to 1 (error detected).	● When cleared by software*[1] ● After transition to LIN reset mode	● LIN operation mode ● LIN wake-up mode	ERR flag in RLN3nLST register	✓
Successful data 1 reception	When the RCDS bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte for each data group is received successfully.*[2]	● When cleared by software ● After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Successful header reception	When a header field is received successfully.	● When cleared by software ● After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	✓

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, or BER flag of the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

22.7.7 Error Statuses

22.7.7.1 LIN Master Mode

(1) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

All error statuses represent interrupt events.

Table 22.99, Types of Error Statuses in LIN Master Mode shows the types of error statuses.

Table 22.99 Types of Error Statuses in LIN Master Mode

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1,*2	• LIN operation mode • LIN wake-up mode	Aborted	Enabled	BER flag in RLN3nLEST register
Physical bus error	• LIN bus is detected to be high level when transmitting a break • LIN bus is detected to be low level when transmitting a break delimiter • LIN bus is detected to be high level when transmitting a wake-up	• LIN operation mode • LIN wake-up mode	Aborted	Enabled	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time*3	LIN operation mode	Aborted	Enabled	FTER flag in RLN3nLEST register
Framing error	In response field reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	Enabled	FER flag in RLN3nLEST register
Checksum error	In response field reception, checksum test results in an error	LIN operation mode	—	Disabled	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: • The first reception data byte is received after completion of header transmission but before a response transmission/reception request is specified • The first reception data byte is received after the completion of previous data group reception but before a transmission/reception request for the next data group is specified.	LIN operation mode	Aborted	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately after the bit that caused the error is transmitted. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the CSM bit in the RLN3nLDFC register), and can be calculated using the following formula.
When the FSM bit in the RLN3nLDFC register is set to 1 (frame separate mode), the timeout time is that of the 8 data bytes until the RTS bit of the RLN3nLTRC register is set. Once the RTS bit is set, the timeout time is changed to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDFC register).

[Frame timeout]
When classic checksum is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]
The aforementioned timeout time is longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.
[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]
When an error is detected, time-out error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set), when it is cleared by software, or at a transition to LIN reset mode.

(2) Target Time Domain for LIN Error Detection

Figure 22.19, Target Time Domain for LIN Error Detection (LIN Master Mode) shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

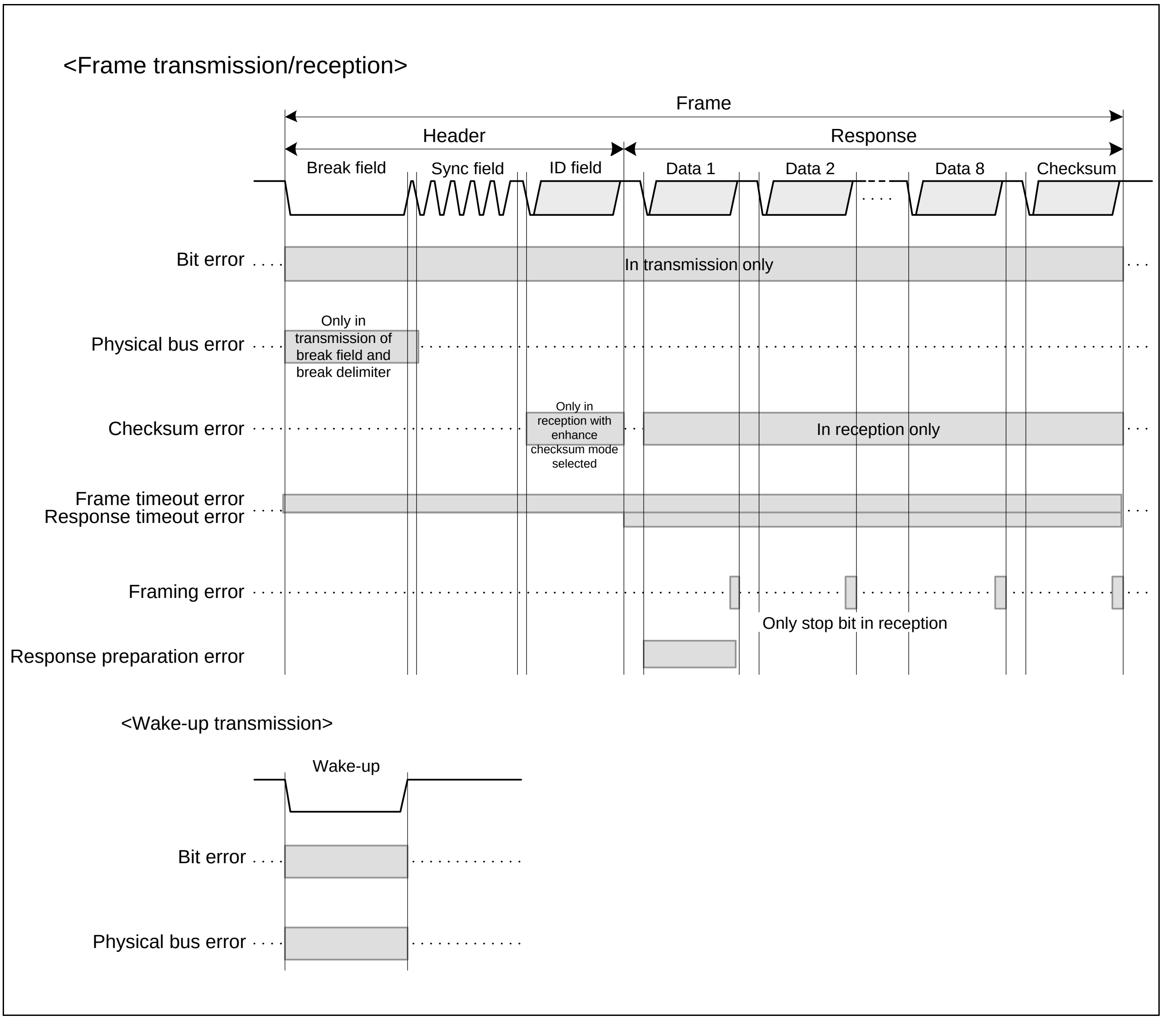


Figure 22.19 Target Time Domain for LIN Error Detection (LIN Master Mode)

22.7.7.2 LIN Slave Mode

(1) Types of Error Statuses

The LIN/UART interface can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

Table 22.100, Types of Error Statuses in LIN Slave Mode shows the types of error statuses.

Table 22.100 Types of Error Statuses in LIN Slave Mode

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match*1 *2	● LIN operation mode ● LIN wake-up mode	Aborted	Enabled	BER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time*3	LIN operation mode	Aborted	Enabled	TER flag in RLN3nLEST register
Framing error	In frame reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	Enabled	FER flag in RLN3nLEST register
Sync field error	If the width of the break low level is greater than the width set by the LBLT bit in the RLN3nLBFC register and the sync field is not 55_H	LIN operation mode	Aborted	Enabled*4	SFER flag in RLN3nLEST register
Checksum error	In response field reception, the checksum test results in an error	LIN operation mode	—*5	Disabled	CSER flag in RLN3nLEST register
ID parity error	If the received ID parity bit does not match the value that is automatically calculated by the LIN/UART interface	LIN operation mode	Aborted	Enabled	IPER flag in RLN3nLEST register
Response preparation error	● After the reception of a header, if the response is not prepared before the first reception data byte is received ● In a multi-byte response reception, if the preparation for the reception of next data group does not complete before the first reception data byte for the next data group is received	LIN operation mode	Aborted	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately after the bit that caused the error is transmitted. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the LCS bit in the RLN3nLDFC register), and this can be calculated according to the following formula. The timeout time is that of 8 data bytes until the RTS bit or the LNRR bit of the RLN3nLTRC register is set. When the RTS bit is set, the timeout time is changed to the time based on the response field data length (RFDL[3:0] bit of the RLN3nLDFC register). When the LNRR bit is set, the timeout function stops.

[Frame timeout]
When classic checksum is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]
When en error is detected, time-out error detection function stops.

Note 4. Only reflection of the result to the SFER flag can be enabled/disabled. Error detection cannot be enabled/disabled.

Note 5. Checksum determination is performed upon completion of response frame reception. In case of an error, the successful reception flag is not set to 1.

The error status is cleared by software or at a transition to LIN reset mode.

(2) Target Time Domain for LIN Error Detection

Figure 22.20, Target Time Domain for LIN Error Detection (LIN Slave Mode) shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

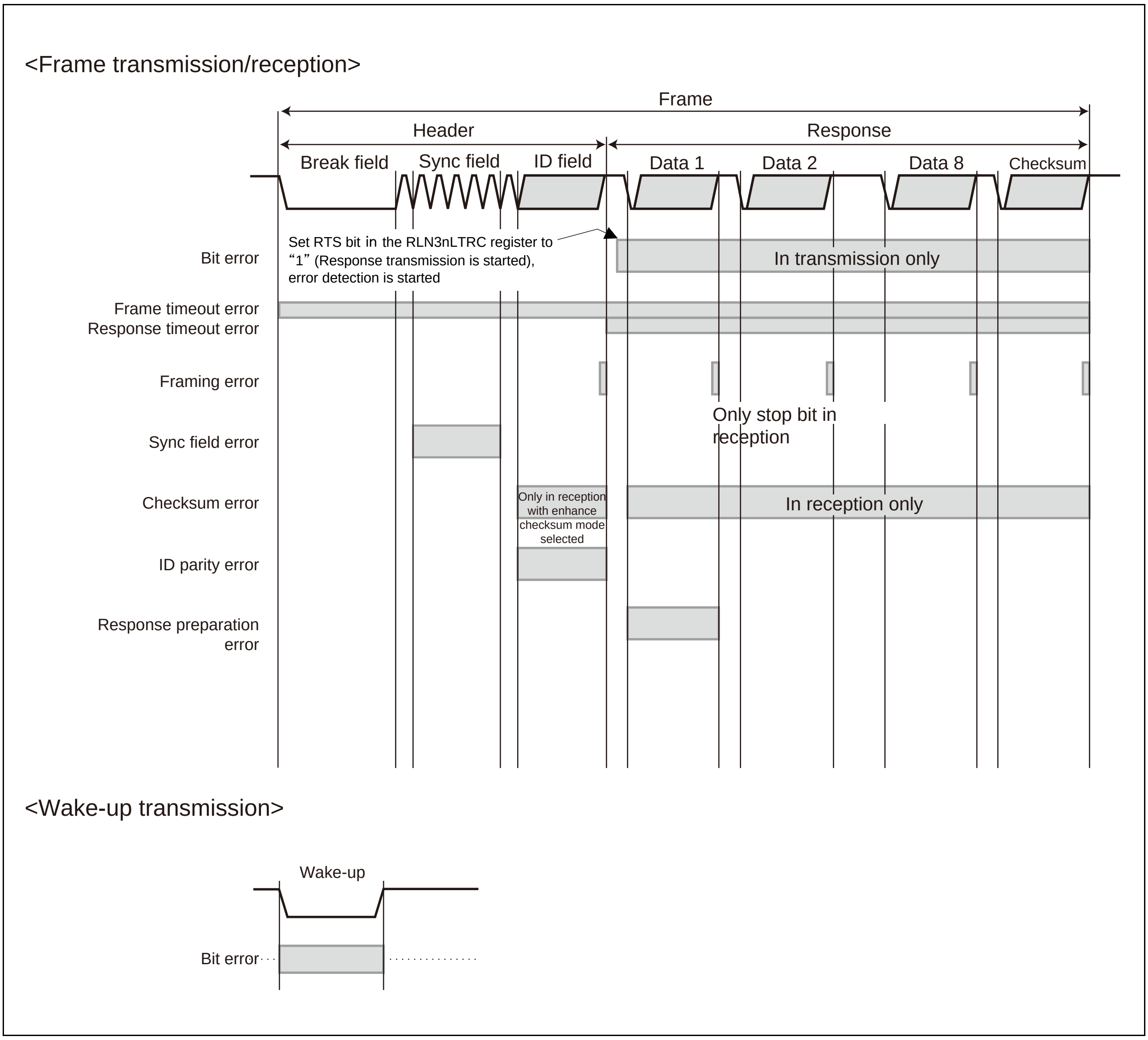


Figure 22.20 Target Time Domain for LIN Error Detection (LIN Slave Mode)

22.8 UART Mode

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

22.8.1 Transmission

Figure 22.21, LIN/UART Interface (in UART Mode) Transmission Operation shows the operation of the LIN/UART interface (in UART mode) during transmission. **Table 22.101, LIN/UART Interface (UART Mode) Transmission Processing** provides processing of the LIN/UART interface (in UART mode) during transmission.

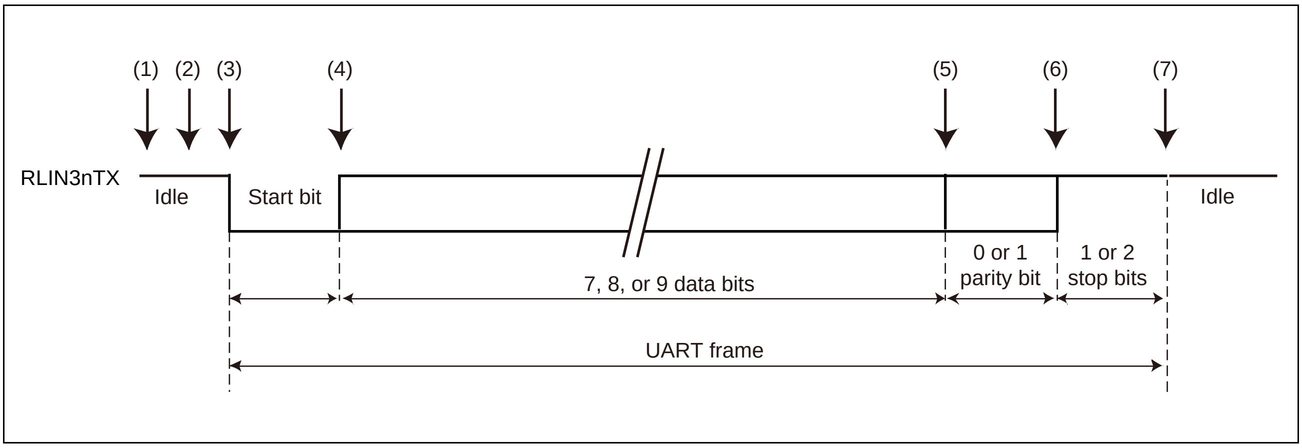


Figure 22.21 LIN/UART Interface (in UART Mode) Transmission Operation

Table 22.101 LIN/UART Interface (UART Mode) Transmission Processing

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Configures data format. • Sets an interrupt generation timing. • Exits from LIN reset mode. • Sets the transmission enable bit (UTOE bit) to 1.	• Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2)	• Specifies the data to be transmitted in the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDR).	• Sets the transmission status flag.
(3)	• Waits for an interrupt request.	• Transmits a start bit (for switching from reception to transmission in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see **Section 22.8.1.4, Transmission Start Wait Function**.) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • Outputs a transmission interrupt.
(4)	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • When transmitting data continuously, specifies the next data to be transmitted in the UART transmission data register (RLN3nLUTDR) and waits for the generation of an interrupt request.	Transmits the data specified in the UART (wait) transmission data register.
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits.

Table 22.101 LIN/UART Interface (UART Mode) Transmission Processing

	Software Processing	LIN/UART Interface Processing
(7)	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] ● If another item of transmission data is set, goes to step (3).	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] ● If another piece of transmission data is set, goes to step (3). ● If the next data to be transmitted is not specified, clears the transmission status flag.
	[When the UTIGTS bit is 1 (a transmission interrupt is output upon completion of transmission)] ● When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 1 (a transmission interrupt is output upon completion of transmission)] ● Generates RLIN3n transmission interrupt request. ● Clears the transmission status flag.

22.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 22.22, Operation Examples of LIN/UART Interface (UART Mode) Continuous Transmission** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the completion of transmission.

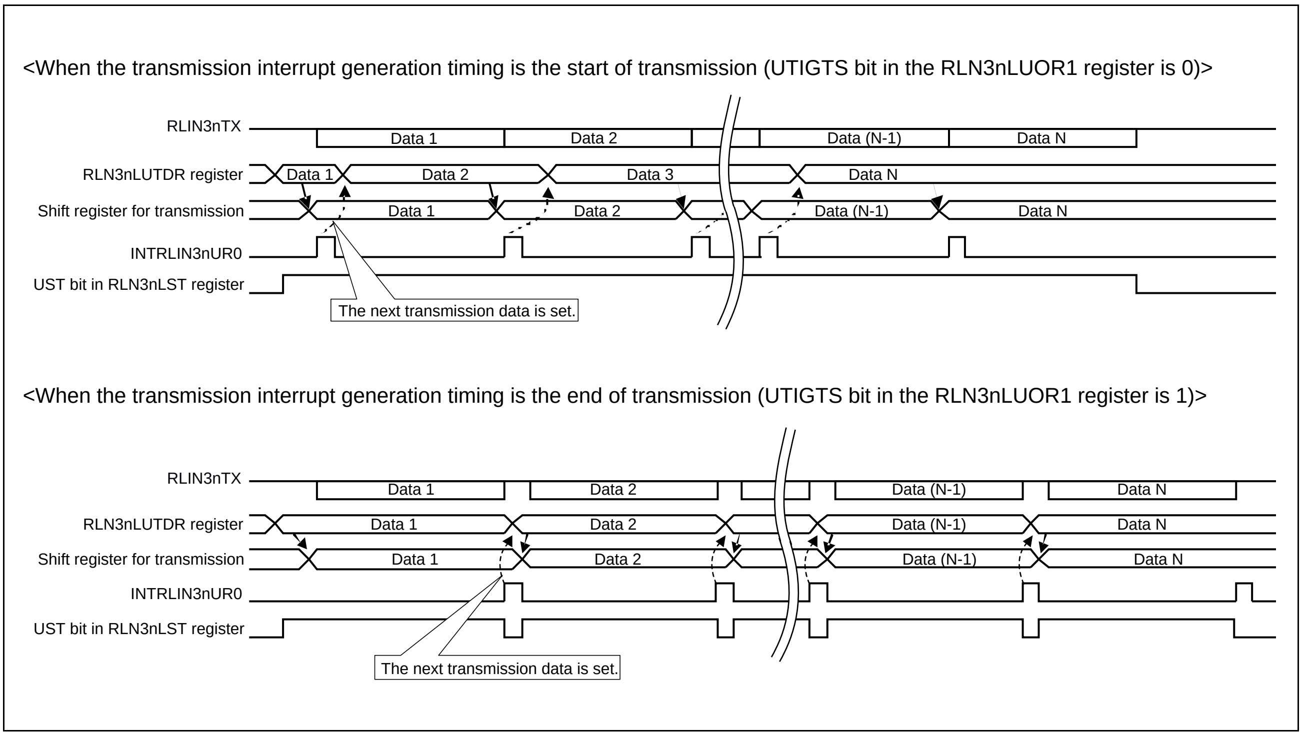


Figure 22.22 Operation Examples of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the completion of a transmission by changing the UTIGTS bit in the RLN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the completion of transmission of final data needs to be known.

22.8.1.2 UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers and it is capable of performing continuous transmissions through the use of UART buffers.

Figure 22.23, UART Buffer Transmission in LIN/UART Interface (in UART Mode) shows the UART buffer transmission operation of the LIN/UART interface (in UART mode).
Table 22.102, UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) shows the UART buffer transmission processing.

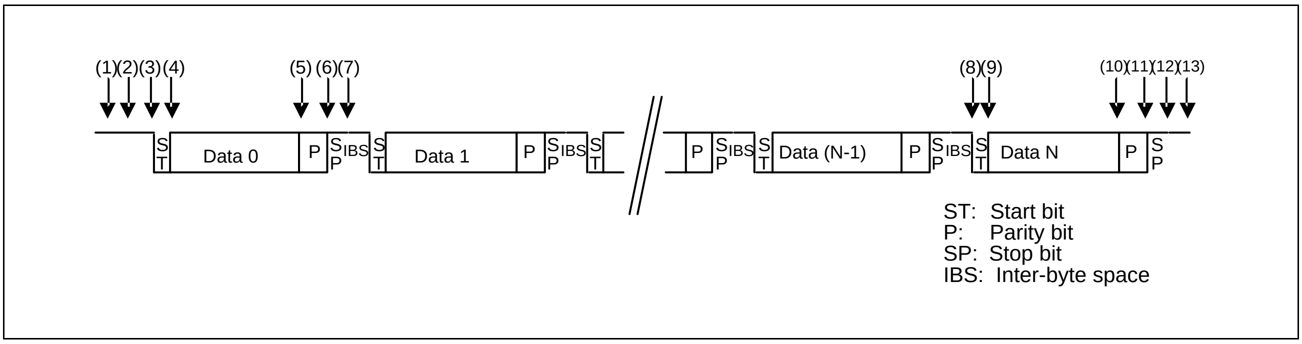


Figure 22.23 UART Buffer Transmission in LIN/UART Interface (in UART Mode)

Table 22.102 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables error detection • Configures data format • Sets the interrupt generation timing to the completion of transmission. • Exits from LIN reset mode. • Sets the transmission enable bit (UTOE bit) to 1	• Waits for a transmission trigger (RTS bit) by software
(2)	• Sets the UART buffer data length and whether the start of transmission must be waited. • Specifies the data to be transmitted in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b =1 to 8) • Sets the UART buffer transmission start bit (RTS).	• Sets the transmission status flag.
(3)	Waits for an interrupt request.	Transmits a start bit. (For switching from reception to transmission in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see **Section 22.8.1.4, Transmission Start Wait Function**.)
(4)		Transmits the data specified in the UART data buffer 0 register (RLN3nLUDB0) or the LIN/UART data buffer b register (RLN3nLDBRb).
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).)
(7)		Transmits an inter-byte space (idle).
		Repeats steps (3) to (7) until the number of data set in the UART buffer data length select bits -1 is reached.

Table 22.103 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode)

	Software Processing	LIN/UART Interface Processing
(8)	Waits for an interrupt request.	Transmits a start bit.
(9)		Transmits the data specified in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)		Transmits a parity bit when parity is used.
(11)		Transmits 1 or 2 stop bits.
(12)		● Sets the successful buffer transmission flag. ● Clears the UART buffer transmission start bit (RTS). ● Generates a transmission interrupt request signal. ● Clears the transmission status flag.
(13)	● Checks the RLN3nLST register, and clears flags ● When continuously transmitting data, goes to step (2).	

(1) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data regions 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is specified. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending on the data length. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data regions 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between each transmitted data can be specified in the IBS bit of the RLN3nLSC register.

Figure 22.24, UART Buffer and Transmission Processing (for 9-Byte Transmission) shows a 9-byte UART buffer and the transmission processing.

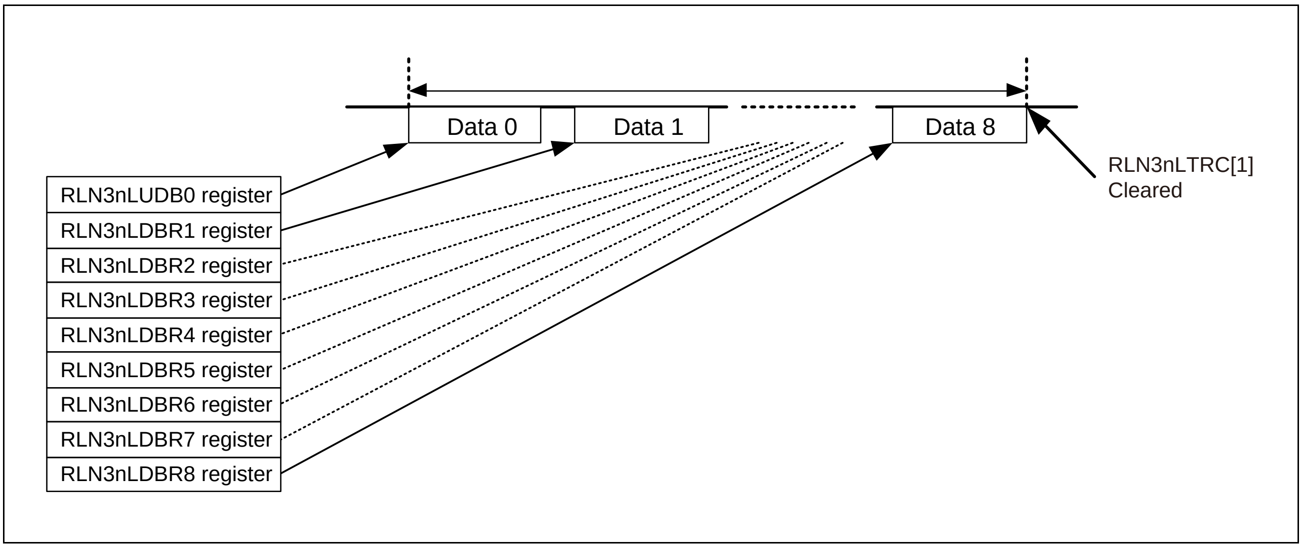


Figure 22.24 UART Buffer and Transmission Processing (for 9-Byte Transmission)

22.8.1.3 Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag of the RLN3nLEST register (see **Section 22.7.7, Error Statuses**). The timing at which the input pin is sampled during data transmission can vary depending on the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 22.104, Error Detection Timing in UART Mode**.

Table 22.104 Error Detection Timing in UART Mode

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 samplings) is shown in **Figure 22.25, Example of Data Transmission Timing (When 1 Tbit = 16 samplings).**

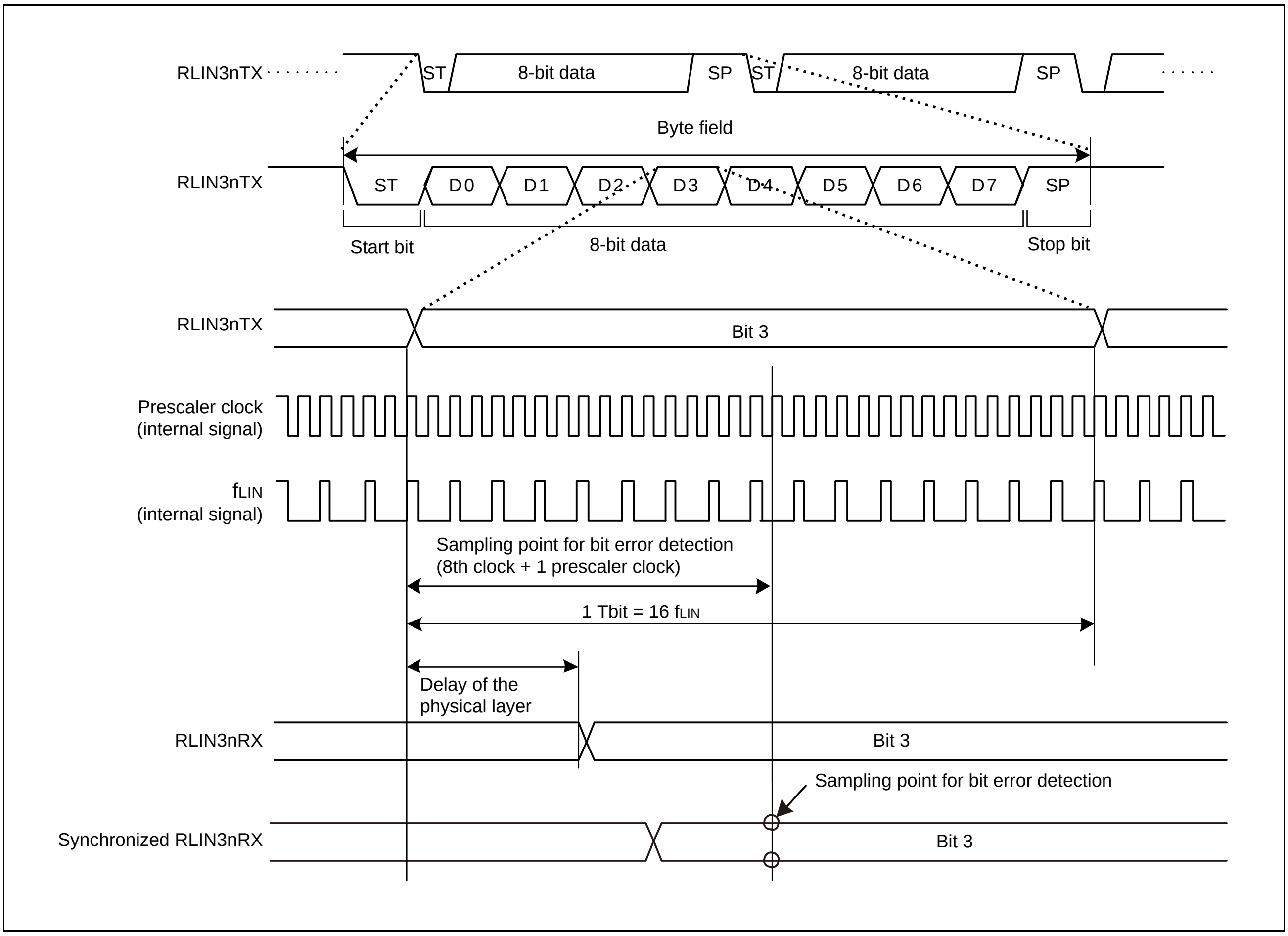


Figure 22.25 Example of Data Transmission Timing (When 1 Tbit = 16 samplings)

22.8.1.4 Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, specify the transmission data in the RLN3nLUWTDR register, which is used only for the wait function, instead of specifying it in the RLN3nLUTDR register as a transmission start request. When transmitting from the UART buffer, set the RTS bit in the RLN3nLTRC register to 1 (UART buffer transmission start) while the UTSW bit in the RLN3nLDFC register is set to 1.

In this case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

It should be noted that even if the UART stop bit length select bit (USBLS) in RLN3nLBFC register is 1 (stop bits = 2 bits), there is only a 1-bit delay.

Figure 22.26, Transmission Wait Function (Case When Transmit Data Is Set While Stop Bits Are Being Received) shows the operation of transmission wait function.

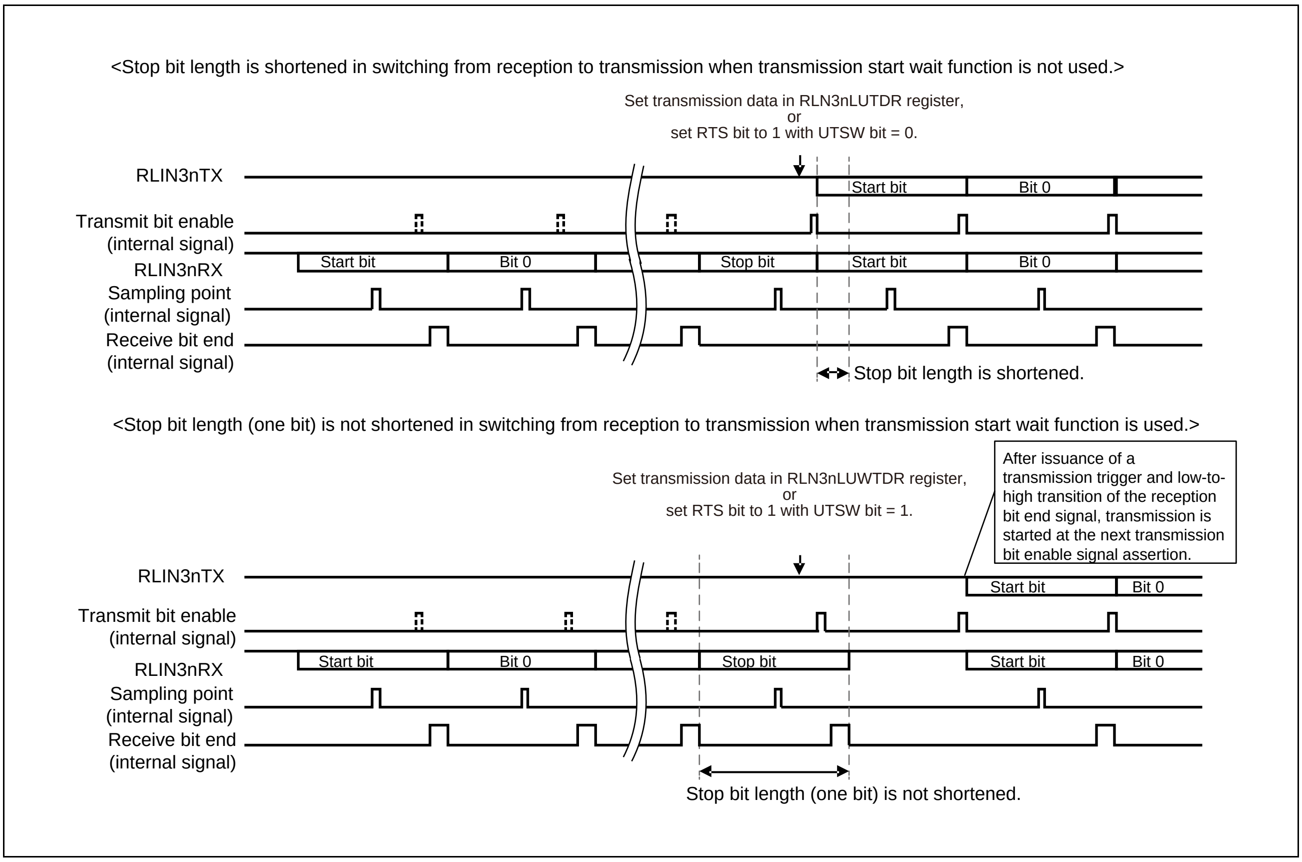


Figure 22.26 Transmission Wait Function (Case When Transmit Data Is Set While Stop Bits Are Being Received)

22.8.2 Reception

Figure 22.27, LIN/UART Interface (in UART Mode) Reception Operation shows the LIN/UART interface (in UART mode) reception operation. **Table 22.105, LIN/UART Interface (in UART Mode) Reception Processing** shows the LIN/UART interface (in UART mode) reception processing.

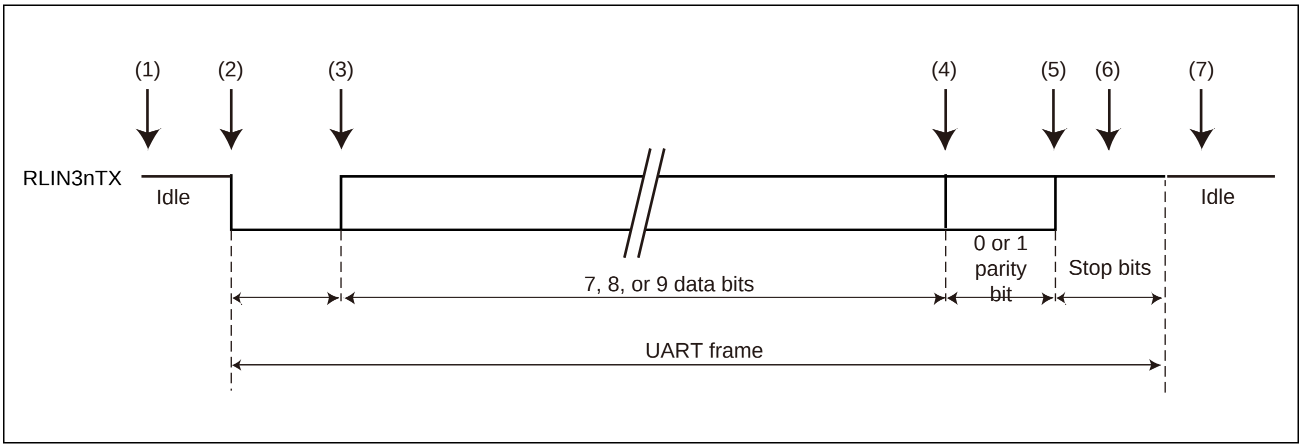


Figure 22.27 LIN/UART Interface (in UART Mode) Reception Operation

Table 22.105 LIN/UART Interface (in UART Mode) Reception Processing

	Software Processing	LIN/UART Interface Processing
(1)	● Sets a baud rate. ● Sets noise filter ON/OFF. ● Enables error detection. ● Configures data format. ● Exits from LIN reset mode. ● Sets the reception enable bit (UROE bit) to 1.	● Waits for the reception to be enabled by software. ● Waits for detection of a start bit.
(2)	Waits for an interrupt request.	● Waits for a falling edge from the reception pin, and detects a start bit. ● Sets the reception status flag.
(3)		Receives data.
(4)		Receives a parity bit when parity is used.
(5)		Receives only 1 stop bit.
(6)		● Generates a RLIN3n successful reception interrupt request. ● Clears the reception status flag.
(7)	Checks the RLN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

22.8.2.1 Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and {(the number of sampling + 1) / 2} / (the number of sampling) Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filter function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-sampling majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 22.28, Example of Data Reception Timing (When 1 Tbit = 16 samplings) shows an example of data reception timing.

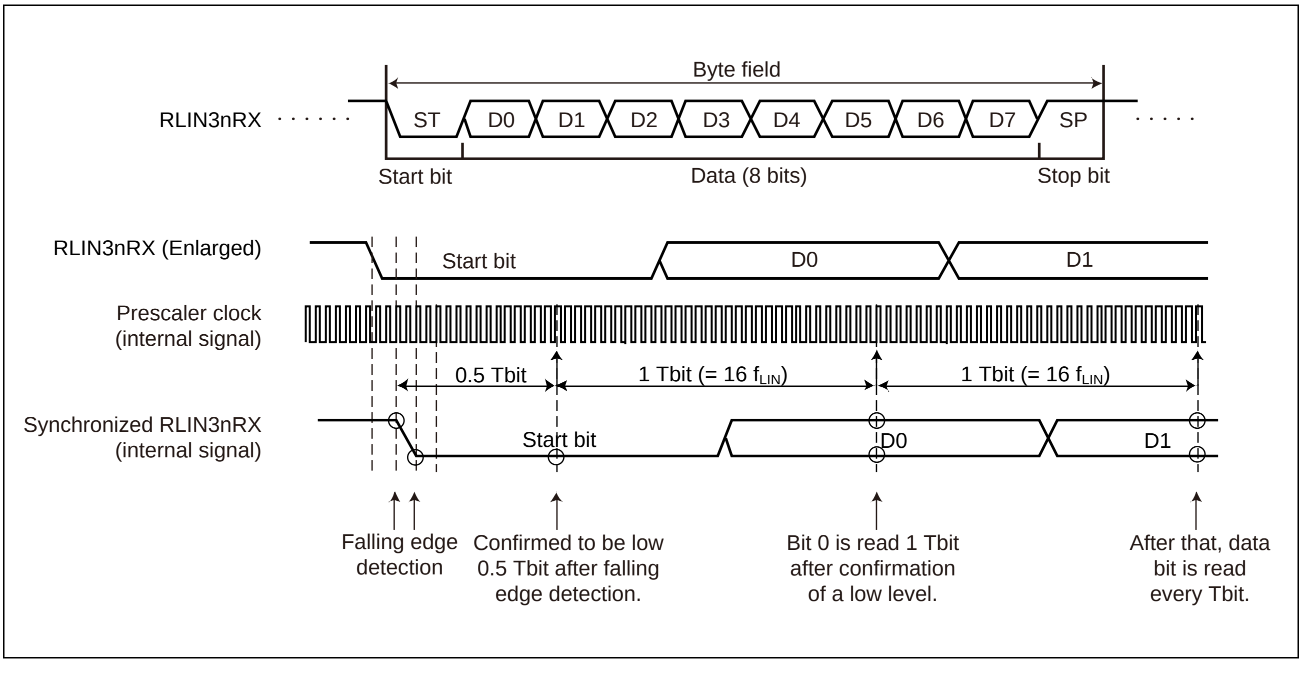


Figure 22.28 Example of Data Reception Timing (When 1 Tbit = 16 samplings)

22.8.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

22.8.3.1 Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR) when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1.

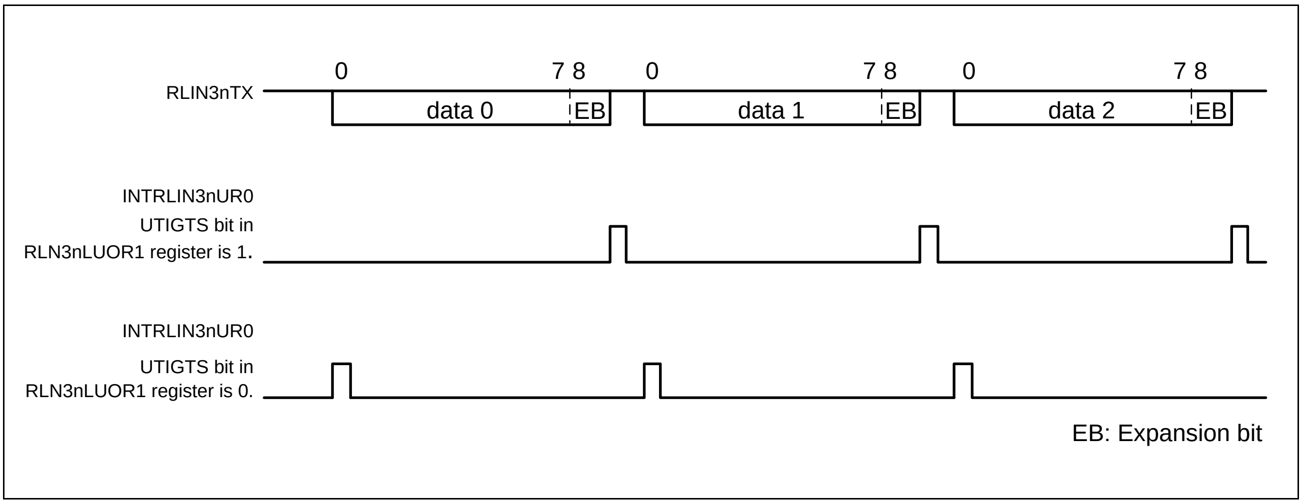


Figure 22.29 Transmission Example When Expansion Bit is Enabled (LSB First)

22.8.3.2 Expansion Bit Reception

The LIN/UART interface (in UART mode) can always receive 9-bit data without comparing the expansion bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Regardless of the expansion bit detection level select bit (UEBDL) setting in the UART option register 1 (RLN3nLUOR1), a RLIN3n successful reception interrupt is generated when 9-bit data is received.

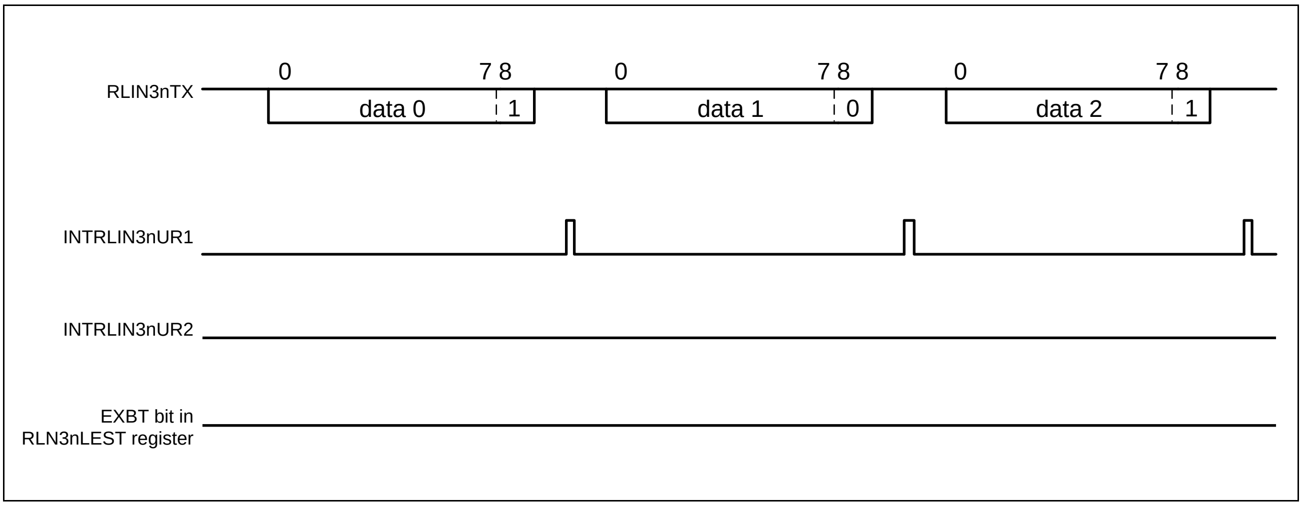


Figure 22.30 Expansion Bit Reception Example (LSB First)

22.8.3.3 Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level select bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, RLIN3n successful reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 22.31, Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0) shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

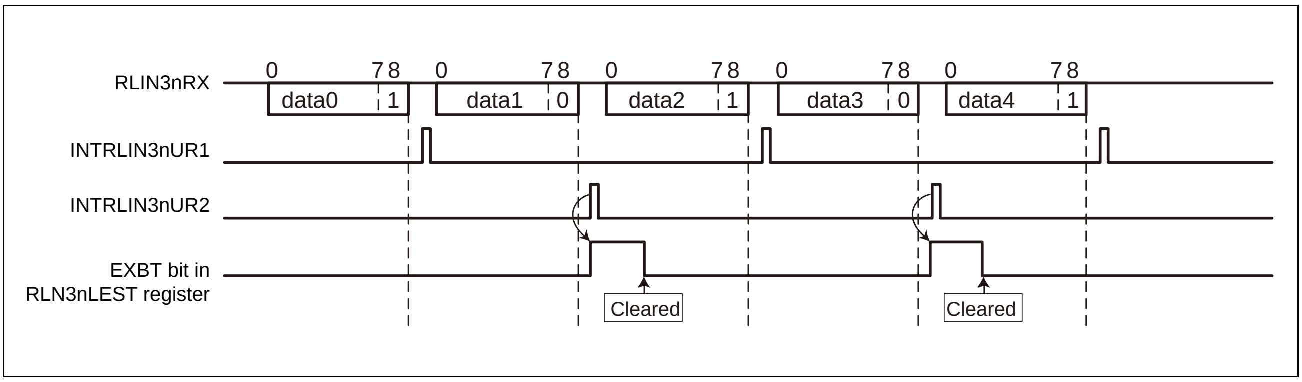


Figure 22.31 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a RLIN3n successful reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

22.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, when the level that was set by the expansion bit detection level select bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits of the received data excluding the expansion bit, with the a preset RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

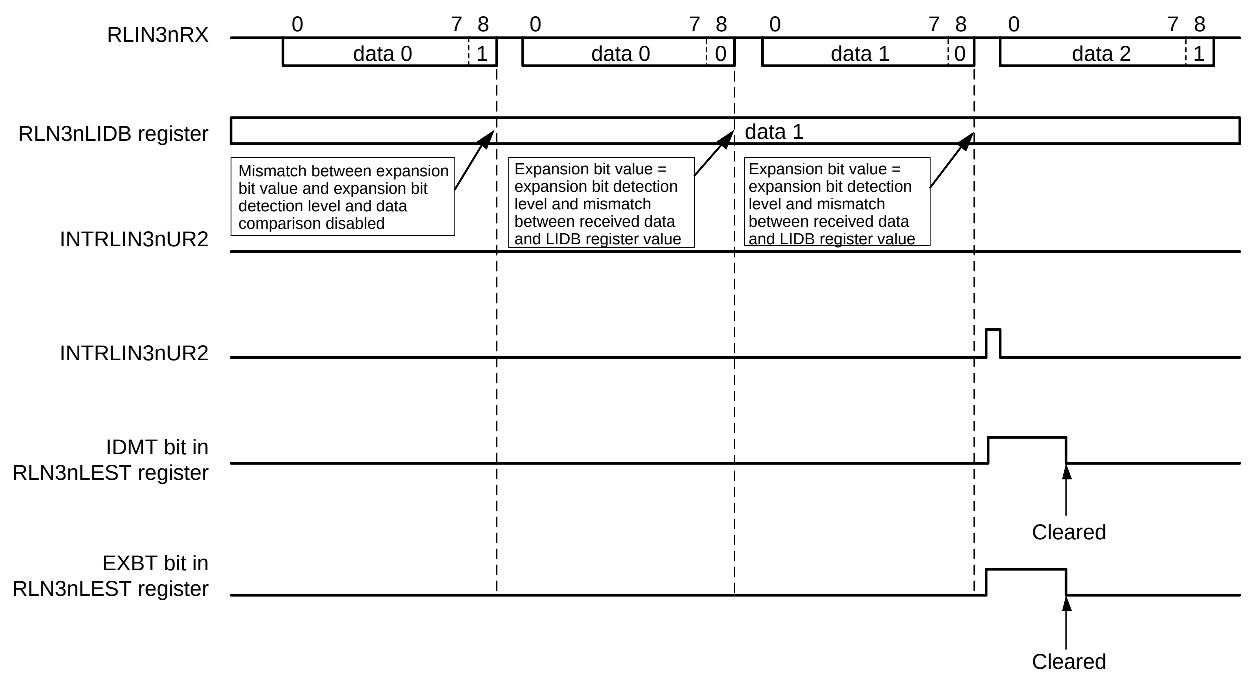
- Generates an RLIN3n status interrupt
- Sets the expansion bit detection flag (EXBT)
- Sets the ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n successful reception interrupt is not generated.

If the result of the comparison is not a match, no RLIN3n successful reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of the next data is completed.

Figure 22.32, Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0) shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.



Note: If a reception error (parity error, framing error, or overrun error) occurs, an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs and the result of comparison is a match, the EXBT and IDMT are also set to 1.

Figure 22.32 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

22.8.4 Statuses

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 22.106, Types of Statuses in UART Mode shows the types of statuses available in UART mode.

Table 22.106 Types of Statuses in UART Mode

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
LIN reset mode	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	• When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), if the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register) is started. • When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon completion of transmission), if the transmission of the data length specified by the MDL bit in the RLN3nLDFC register is completed.	• When cleared by software • After transition to LIN reset mode	FTC flag in RLN3nLST register	✓
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When cleared by software*[1] • After transition to LIN reset mode	ERR flag in RLN3nLST register	✓
Transmission status	• When data is written to the RLN3nLUTDR or RLN3nLUWTDR register. • When 1 is written to the RTS bit in the RLN3nLTRC register.	• When the transmission of the data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed, but the next transmission data is not specified. • When the transmission of the data in the UART buffer is completed and the RTS bit in the RLN3nLTRC register is cleared. • After transition to LIN reset mode	UTS flag in RLN3nLST register	—
Reception status	When a start bit is detected.	• When a sampling point for stop bits is detected • After transition to LIN reset mode	URS flag in RLN3nLST register	—

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when not in the LIN reset mode sets the ERR flag in the RLN3nLST register to 0.

22.8.5 Error Statuses

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

Table 22.107, Types of Statuses in UART Mode lists applicable status types.

Table 22.107 Types of Statuses in UART Mode

Status	Error Detection Condition	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the reception pin do not match*[1]	Continues until the transmission of the specified transmission data is completed.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, the next data is received before the previous data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is completed by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is completed by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is completed.	Disabled*[2]	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit received data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

22.9 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode. When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and they are internally connected within the LIN/UART interface. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following four modes:

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception
- LIN slave self-test mode (transmission): Header reception and response transmission
- LIN slave self-test mode (reception): Header reception and response reception

In LIN self-test mode, the operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps]. (The NSPB bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- LIN slave mode (Auto baud rate)
- Frame/response timeout error

Do not use these functions.

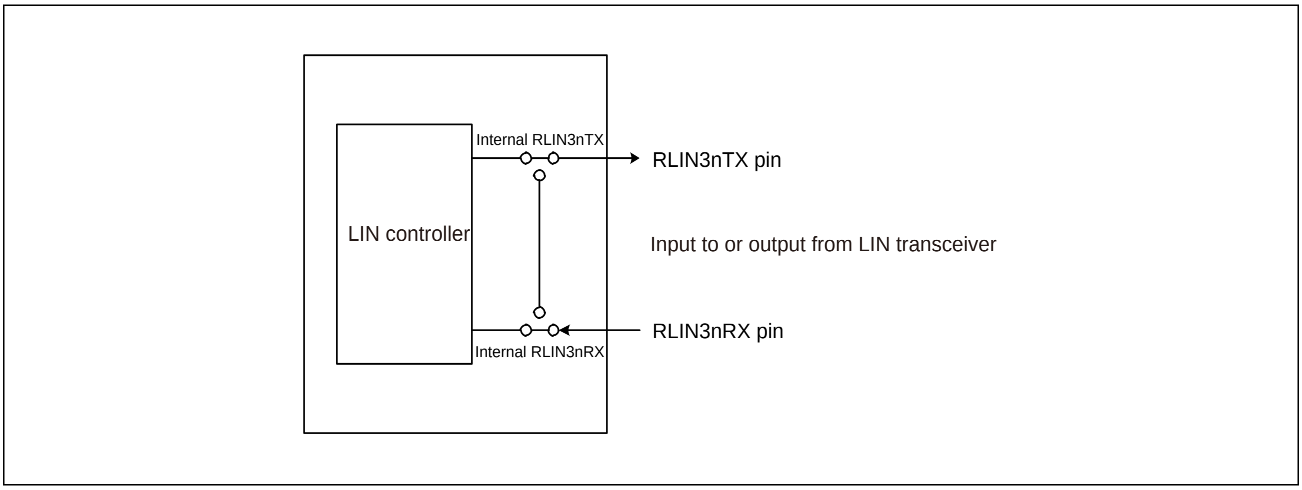


Figure 22.33 Connection in LIN Reset Mode, LIN Mode, and UART Mode

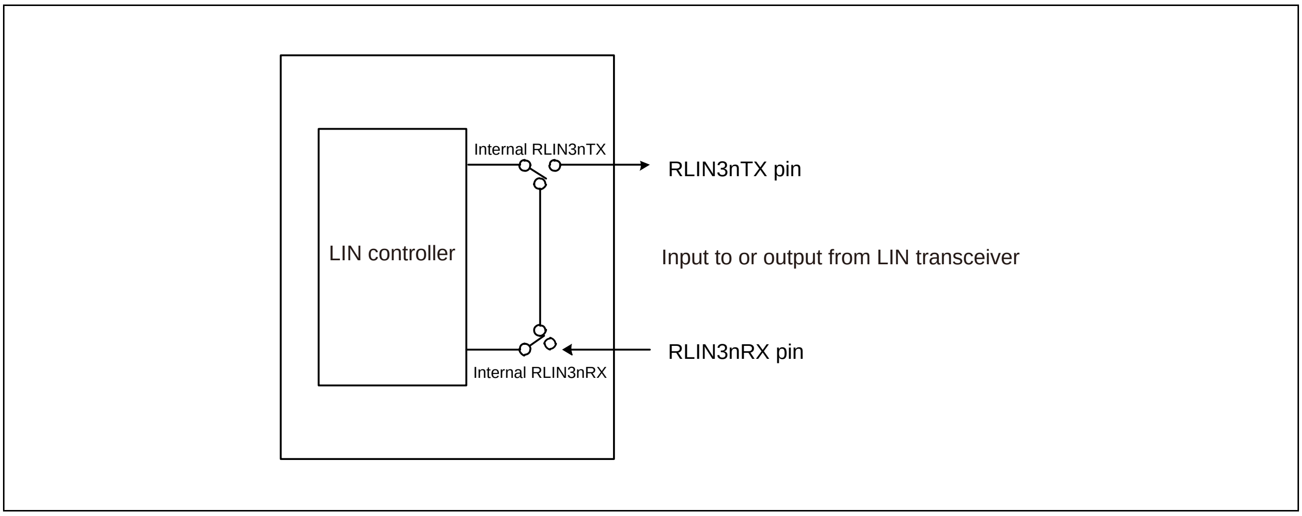


Figure 22.34 Connection in LIN Self-Test Mode

22.9.1 Transition to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN self-test mode.

When transitioning to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be consecutively written three times to the LIN self-test control register, as follows:

- Transition to LIN reset mode
  Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD bits in RLN3nLMD = 00_B (LIN master mode) or 11_B (LIN slave mode [fixed baud rate])
- 1st write: RLN3nLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN3nLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN3nLSTC register = $0000\ 0001_B$ (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the 1st write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is aborted. The above sequence should be retried from the 1st write step. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also aborted.

22.9.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 xxxx_B*1
  RLN3nLBRP0 register = xxxx xxxx_B*1
  RLN3nLBRP1 register = xxxx xxxx_B*1
  RLN3nLMD register = 00xx xx00_B*1
- Configure the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 xxxx_B*2
  RLN3nLEDE register = x000 x0xx_B
- Configure the break field and space related registers.
  RLN3nLBFC register = 00xx xxxx_B
  RLN3nLSC register = 00xx 0xxx_B
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B.
- Configure the transmission frame related registers.
  RLN3nLDFC register = 00x1 xxxx_B
  RLN3nLIDB register = xxxx xxxx_B
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx_B
- Start header transmission → response transmission
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started). The LIN master self-test mode (transmission) is executed, interrupt is generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/UART interface. To abort the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, configuration of these registers is not necessary.

Note 2. If necessary, configure the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up transmission flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1)\ [\text{Tbit}]$$

1 Tbit = 1/frequency of LIN communication clock source × 16

22.9.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxxx_B$*[1]
  RLN3nLBRP0 register = xxxx $xxxx_B$*[1]
  RLN3nLBRP1 register = xxxx $xxxx_B$*[1]
  RLN3nLMD register = 00xx $xx00_B$*[1]
- Configure the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*[2]
  RLN3nLEDE register = x000 $x0xx_B$
- Configure the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$*[1]
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B.
- Configure the reception frame related registers.
  RLN3nLDFC register = 00x0 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
  RLN3nCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, perform the calculation and specify the calculated value in the RLN3nLCBR register.
- Start header transmission → response reception
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started). The LIN master self-test mode (reception) is executed, interrupt is generated, and status and error status are also updated. To abort the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the specified value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, configuration of these registers is not necessary.

Note 2. If necessary, configure the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not

completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

22.9.4 Transmission in LIN Slave Self-Test Mode

To execute a self-test on LIN slave transmission, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 xxx0$_B$*1
  RLN3nLBRP0 register = xxxx xxxx$_B$*1
  RLN3nLBRP1 register = xxxx xxxx$_B$*1
  RLN3nLMD register = 00x x0011$_B$
- Configure the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 xxxx$_B$*2
  RLN3nLEDE register = xx0x x00x$_B$
- Configure the break field and space related registers.
  RLN3nLBFC register = 0000 000x$_B$*3
  RLN3nLSC register = 00xx 0001$_B$
- Exit the LIN reset mode.
  Write 11$_B$ to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11$_B$.
- Configure the transmission frame related registers.
  RLN3nLDFC register = 00x1 xxxx$_B$
  RLN3nLIDB register = xxxx xxxx$_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx$_B$
- Header reception → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started).
  (The header reception and the response transmission are executed in this order, without manipulating the RTS bit in the RLN3nLTRC register.)
  The LIN slave self-test mode (transmission) is executed, interrupt is generated, and status and error status are also updated.
  The checksum is automatically calculated by the LIN/UART interface. To abort the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, and the RLN3nLBRP1 register. Therefore, configuration of these registers is not necessary.

Note 2. If necessary, configure the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response transmission interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).

The time from setting of the successful header reception flag to setting of the successful response/wake-up transmission flag is calculated by using the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

22.9.5 Reception in LIN Slave Self-Test Mode

To execute a self-test on LIN slave reception, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 xxx0$_B$*[1]
  RLN3nLBRP0 register = xxxx xxxx$_B$*[1]
  RLN3nLBRP1 register = xxxx xxxx$_B$*[1]
  RLN3nLMD register = 00xx 0011$_B$
- Configure the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 xxxx$_B$*[2]
  RLN3nLEDE register = xx0x x00x$_B$
- Configure the break field and space related registers.
  RLN3nLBFC register = 0000 000x$_B$*[3]
  RLN3nLSC register = 00xx 0001$_B$*[1]
- Exit the LIN reset mode.
  Write 11$_B$ to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11$_B$.
- Configure the reception frame related registers.
  RLN3nLDFC register = 00x0 xxxx$_B$
  RLN3nLIDB register = xxxx xxxx$_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx$_B$
  RLN3nCBR register = xxxx xxxx$_B$
  Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header reception → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started).
  (Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response reception are executed in this order.)
  The LIN slave self-test mode (reception) is executed, interrupt is generated, and status and error status are also updated. To abort the LIN slave self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the specified value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register, and the IBS bit in the RLN3nLSC register.
Therefore, configuration of these registers is not necessary.

Note 2. If necessary, configure the related registers described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**, **Section 7BC, Exception/Interrupts of RH850/F1KM**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response reception interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).
The time required from the set of the successful header reception flag to the set of the successful response/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

22.9.6 Exiting LIN Self-Test Mode

To exit LIN self-test mode, perform the following procedure:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
  If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B, transition to LIN reset mode.
- Verify that LIN/UART interface has exited LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

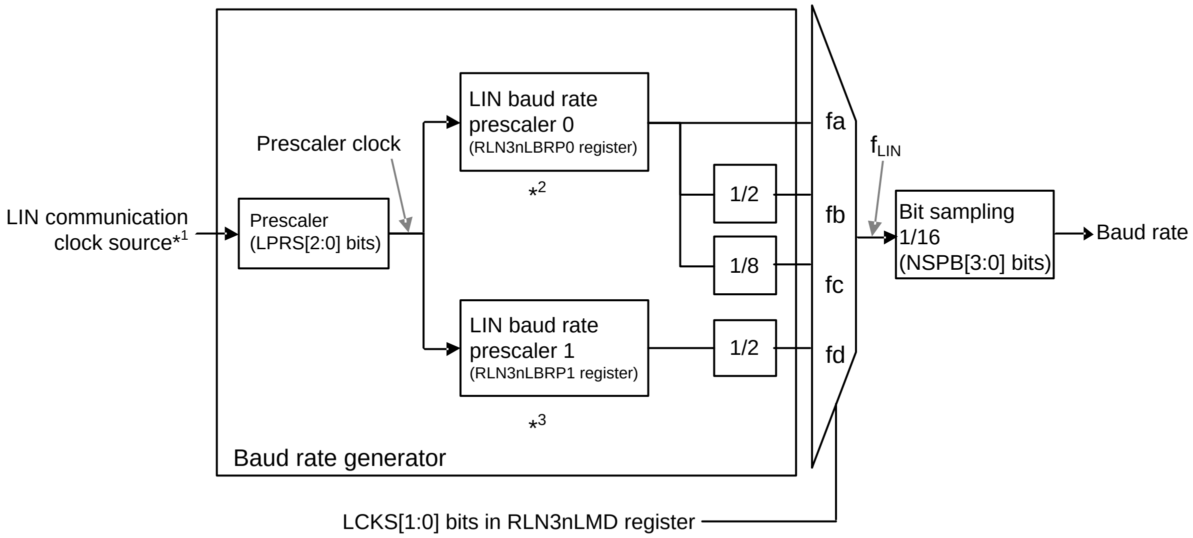
22.10 Baud Rate Generator

The prescaler clock is obtained by dividing the LIN communication clock source frequency by the prescaler, and the LIN system clock (f_{LIN}) is obtained by dividing the prescaler clock frequency by the baud rate generator. The clock obtained by dividing the LIN system clock frequency (f_{LIN}) by the number of samples is the baud rate. The inverse of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generators are switched according to the mode used.

22.10.1 LIN Master Mode

Figure 22.35, Block Diagram of Baud Rate Generation in LIN Master Mode shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see **Section 12AB.5.3, Clock Domain Settings**, **12C.5.3, Clock Domain Settings**.

Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.

Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 22.35 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting bit rates are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These bit rates are frequency-divided by 16 in the bit timing generator, enabling baud rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by configuring the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

Baud rate of LIN master

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

÷ (RLN3nLBRP0+ 1) ÷ 16 [bps] (When fa is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

÷ (RLN3nLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

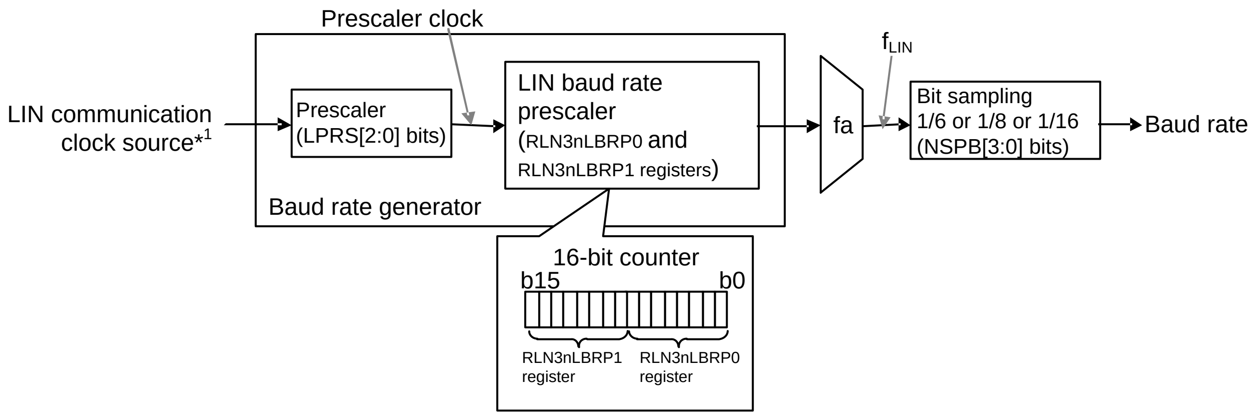
÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for f_{LIN})

22.10.2 LIN Slave Mode

Figure 22.36, Block Diagram of Baud Rate Generation in LIN Slave Mode shows a block diagram of baud rate generation in LIN slave mode.



Note 1. For the LIN communication clock source, see **Section 12AB.5.3, Clock Domain Settings**, **12C.5.3, Clock Domain Settings**.

Figure 22.36 Block Diagram of Baud Rate Generation in LIN Slave Mode

In LIN slave mode (auto baud rate), the baud rate can be specified in the range of 1 kbps to 20 kbps. Configure the prescaler clock as follows according to the target baud rate:

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4 MHz*1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4 MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to “0011_B” (4 samplings).

The formula for baud rate is described below.
Baud rate of LIN slave

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

÷ (RLN3nLBRP01 + 1) ÷ 16 [bps] ([Fixed baud rate])

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selected clock)

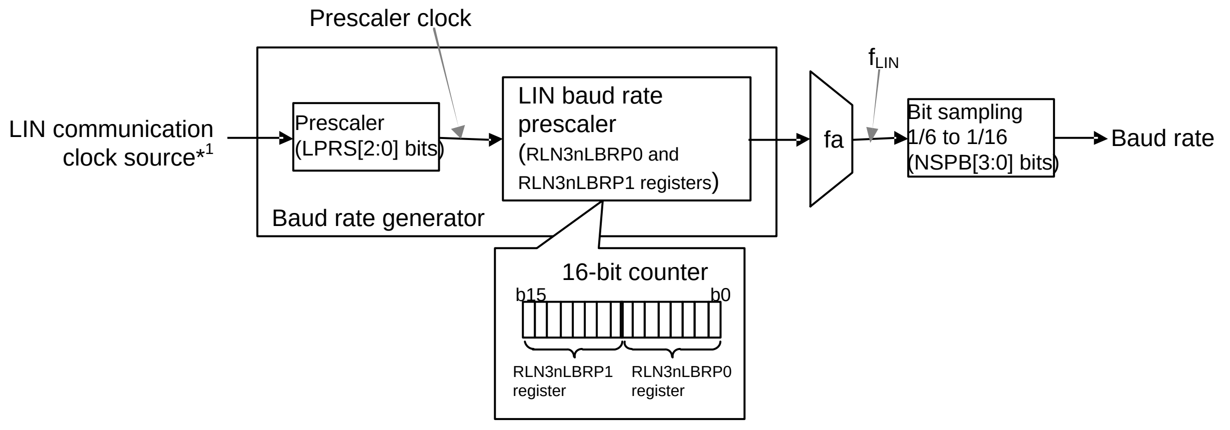
÷ (RLN3nLBRP01 + 1) ÷ 4 or 8 [bps] ([Auto baud rate])

NOTE

For a LIN slave with fixed baud rate, set the NSPB[3:0] bit to “0000$_B$” (16 samplings) or “1111$_B$” (16 samplings). For a LIN slave with auto baud rate, set the NSPB[3:0] bits to “0011$_B$” (4 samplings) or “0100$_B$” (8 samplings).

22.10.3 UART Mode

Figure 22.37, Block Diagram of Baud Rate Generation in UART Mode shows a block diagram of baud rate generation in UART mode.



Note 1. For the LIN communication clock source, see **Section 12AB.5.3, Clock Domain Settings**, **12C.5.3, Clock Domain Settings**.

Figure 22.37 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

UART baud rate

= {LIN communication clock source frequency} × (RLN3nLWBR.LPRS[2:0] selected clock) ÷ (RLN3nLBRP01 + 1) ÷ {RLN3nLWBR.NSPB[3:0] selected count} [bps]

22.11 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 22.38, Configuration of Noise Filter shows the configuration of the noise filter, **Figure 22.39, Example of Noise Filter Circuit** shows an example of a noise filter circuit, and **Figure 22.40, Determination of Received Data when Noise Filter is Used** shows the determination of the received data when the noise filter is used.

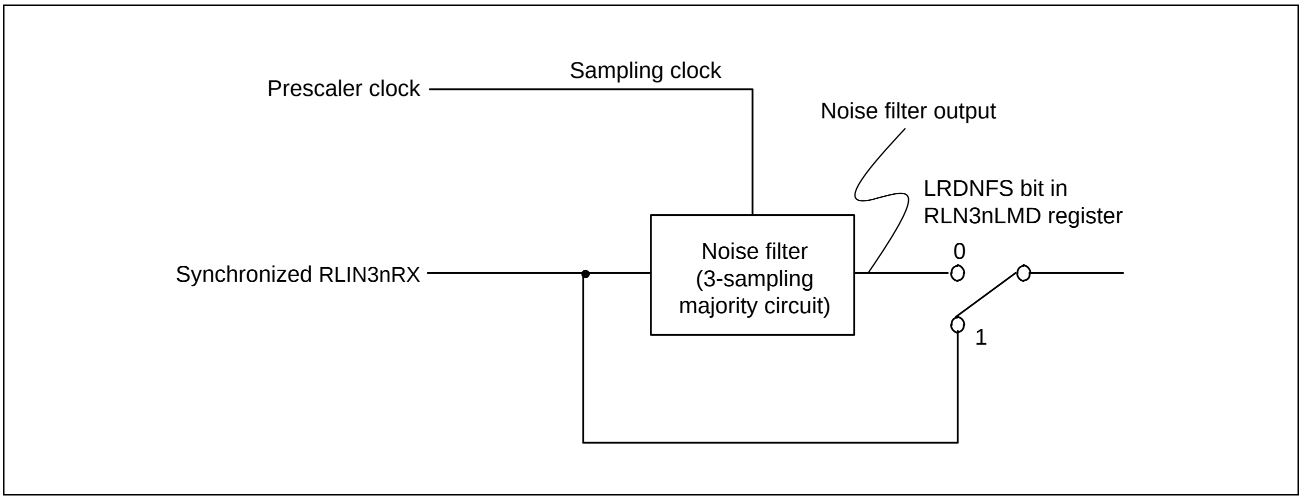


Figure 22.38 Configuration of Noise Filter

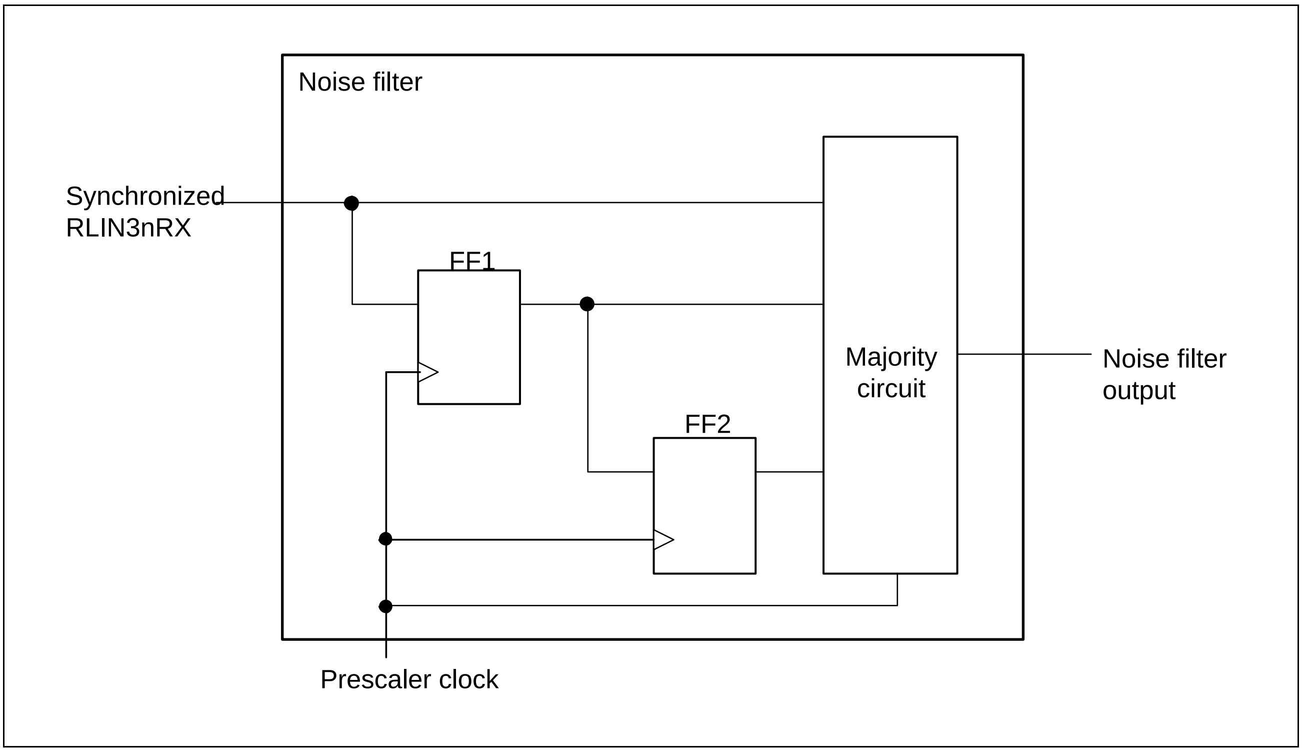


Figure 22.39 Example of Noise Filter Circuit

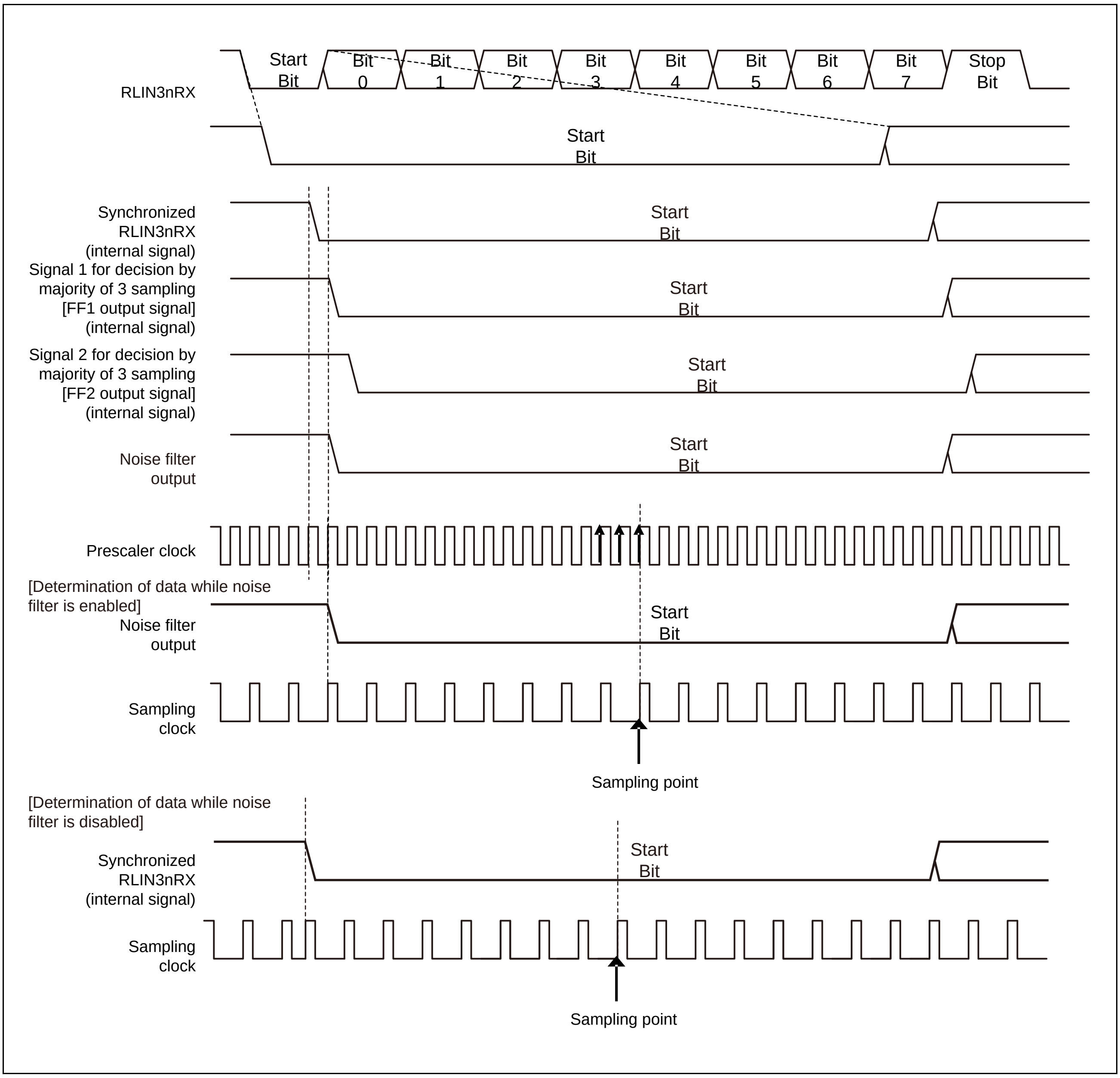


Figure 22.40 Determination of Received Data when Noise Filter is Used

Section 23 I²C Bus Interface (RIIC)

This section contains a generic description of the I²C Bus Interface (RIIC).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RIIC.

23.1 Features of RH850/F1KH, RH850/F1KM RIIC

23.1.1 Number of Units

This microcontroller has the following number of RIIC units.

Each RIIC unit has single channel interface.

Table 23.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	2	2	2
Name	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)

Table 23.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	2	2	2	2	2
Name	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)

Table 23.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	2	2	2	2
Name	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)	RIICn (n = 0, 1)

Table 23.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RIIC units are identified by the index “n”: for example, RIICnCR1 (n = 0, 1) is the I²C bus control register 1.

23.1.2 Register Base Addresses

RIIC base addresses are listed in the following table.

RIIC register addresses are given as offsets from the base addresses.

Table 23.5 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<RIIC0_base>	FFCA 0000_H
<RIIC1_base>	FFCA 0080_H

23.1.3 Clock Supply

The RIIC clock supply is shown in the following table.

Table 23.6 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RIICn	PCLK*[1]	CKSCLK_IIIC	Communication clock
	Register access clock	CPUCLK_L, CKSCLK_IIIC	Bus clock

Note 1. Set the period of PCLK no greater than 1/2 of the width at high level of the SCL clock.

23.1.4 Interrupt Requests

RIIC interrupt requests are listed in the following table.

Table 23.7 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RIIC0			
INTRIICnEE	RIIC communication error / event generation interrupt	77	—
INTRIICnRI	RIIC receive end interrupt	78	20
INTRIICnTI	RIIC transmit data empty interrupt	76	19
INTRIICnTEI	RIIC transmit end interrupt	79	—
RIIC1			
INTRIICnEE	RIIC communication error / event generation interrupt	241	—
INTRIICnRI	RIIC receive end interrupt	242	114
INTRIICnTI	RIIC transmit data empty interrupt	240	113
INTRIICnTEI	RIIC transmit end interrupt	243	—

23.1.5 Reset Sources

RIIC reset sources are listed in the following table. RIIC is initialized by these reset sources.

Table 23.8 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RIICn	All reset sources (ISORES)

23.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

Table 23.9 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RIIC0		
RIICnSCL	Serial clock I/O pin	RIIC0SCL
RIICnSDA	Serial data I/O pin	RIIC0SDA
RIIC1		
RIICnSCL	Serial clock I/O pin	RIIC1SCL
RIICnSDA	Serial data I/O pin	RIIC1SDA

When using these ports, the PBDCn register for the corresponding port and the corresponding bit in the PODCn register must be set to 1.

23.2 Overview

23.2.1 Functional Overview

Communications format

- I^{2}C bus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times according to the specified transfer rate

Transfer rate

Up to 400 kbps

SCL clock

- For master operation, the duty cycle of the SCL clock is selectable in the following range:
  - 0% < Duty < 100%

Issuing and detecting conditions

Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.

Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses and device ID addresses are detectable.

Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching of internal and line levels for transmit data.
- Loss of arbitration due to detection of the start condition while the bus is busy can be detected (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching can be detected.
- Loss of arbitration due to non-matching of internal and line levels for data can be detected in slave transmission.

Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

Noise removal

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.

Interrupt sources

- Four sources:
  - Error in transfer or occurrence of events (detection of arbitration loss, NACK, time-out, a start condition including a restart condition, or a stop condition)
  - Reception complete (including matching with a slave address)
  - Transmit-data-empty (including matching with a slave address)
  - Transmission complete

23.2.2 Block Diagram

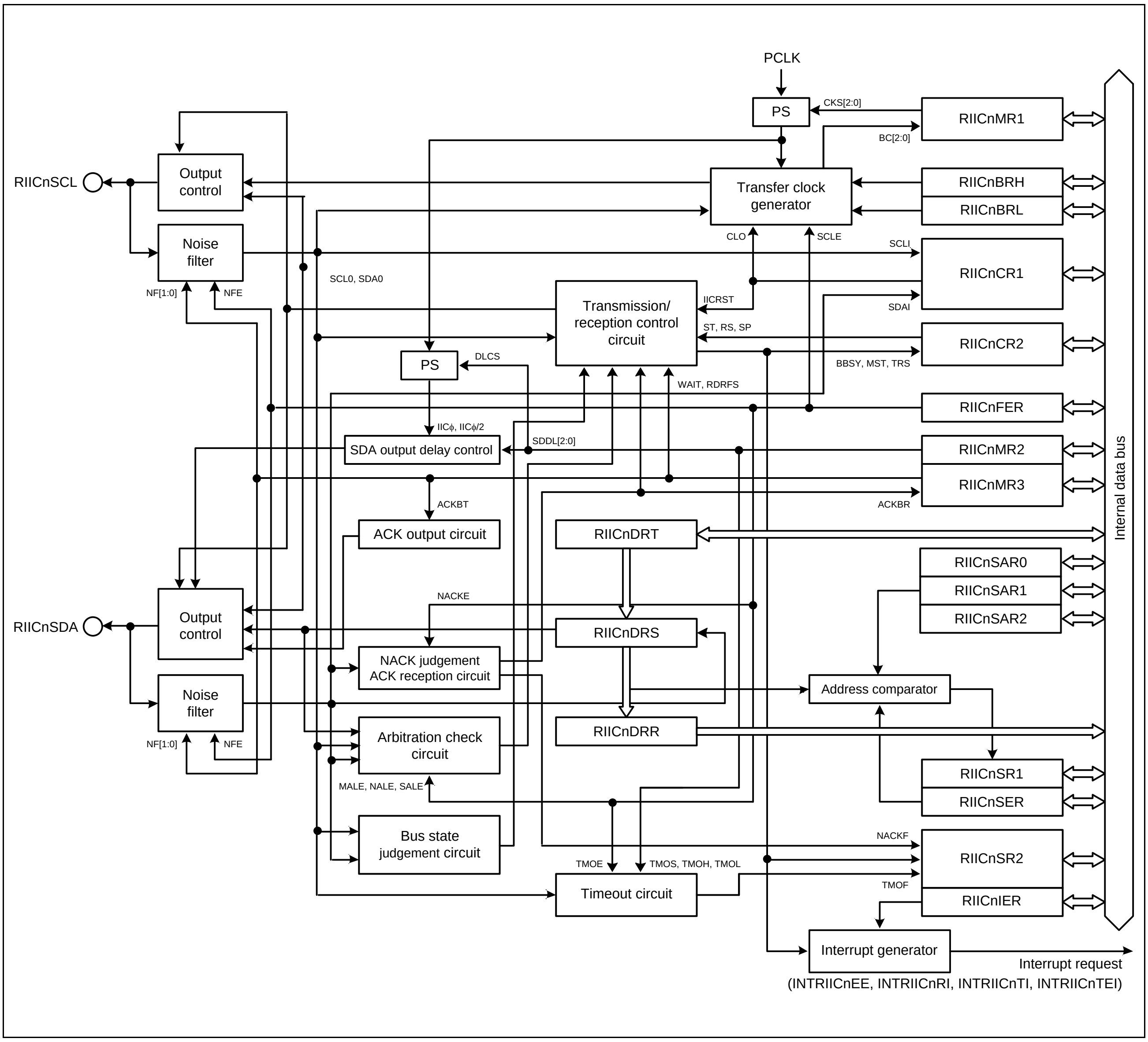


Figure 23.1 Block Diagram of RIIC

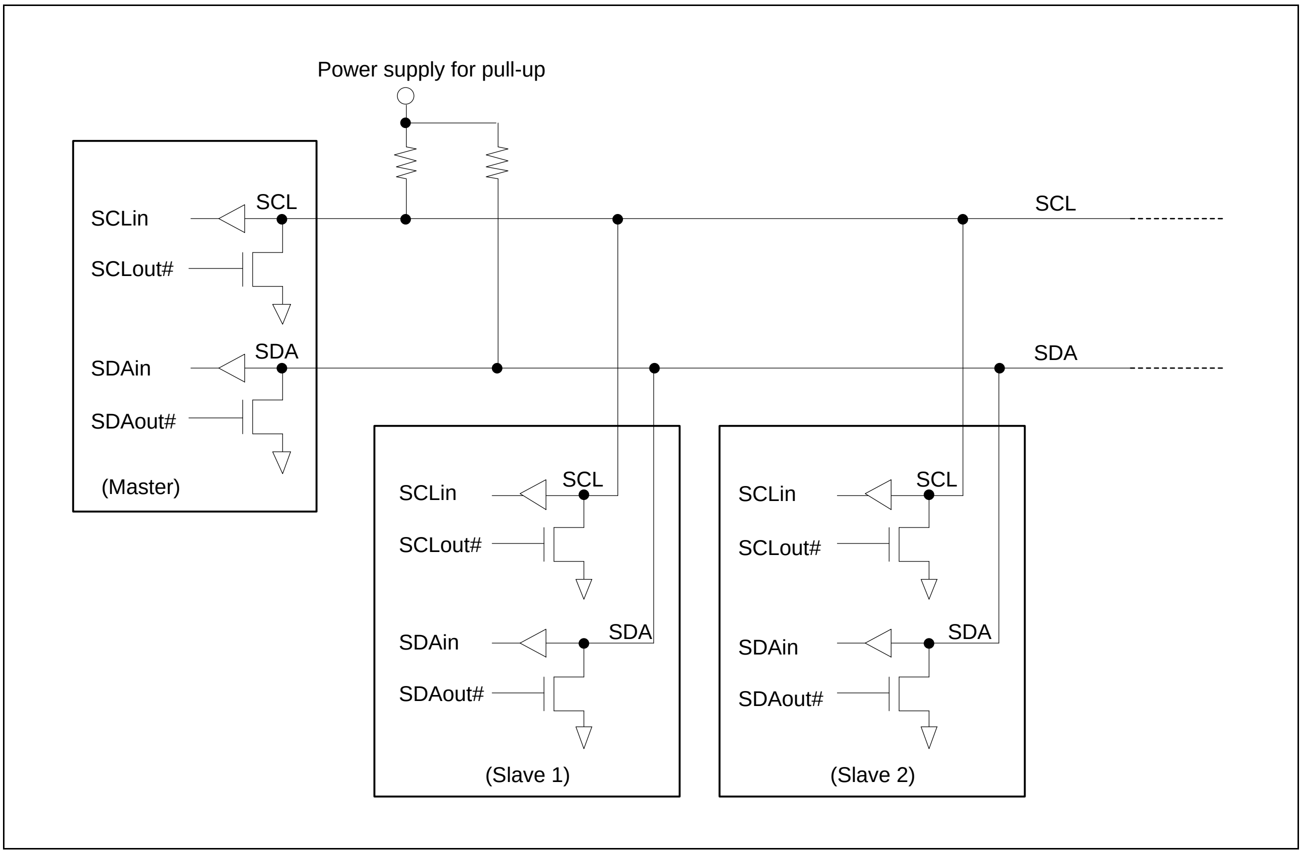


Figure 23.2 Connections to the External Circuit by the I/O Pins (I^{2}C Bus Configuration Example)

23.3 Registers

23.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn_base>, see **Section 23.1.2, Register Base Addresses**.

Table 23.10 List of Registers

Module Name	Register Name	Symbol	Address
RIICn	I^{2}C bus control register 1	RIICnCR1	<RIICn_base> + 0000_H
	I^{2}C bus control register 2	RIICnCR2	<RIICn_base> + 0004_H
	I^{2}C bus mode register 1	RIICnMR1	<RIICn_base> + 0008_H
	I^{2}C bus mode register 2	RIICnMR2	<RIICn_base> + $000C_H$
	I^{2}C bus mode register 3	RIICnMR3	<RIICn_base> + 0010_H
	I^{2}C bus function enable register	RIICnFER	<RIICn_base> + 0014_H
	I^{2}C bus status enable register	RIICnSER	<RIICn_base> + 0018_H
	I^{2}C bus interrupt enable register	RIICnIER	<RIICn_base> + $001C_H$
	I^{2}C bus status register 1	RIICnSR1	<RIICn_base> + 0020_H
	I^{2}C bus status register 2	RIICnSR2	<RIICn_base> + 0024_H
	I^{2}C slave address register 0	RIICnSAR0	<RIICn_base> + 0028_H
	I^{2}C slave address register 1	RIICnSAR1	<RIICn_base> + $002C_H$
	I^{2}C slave address register 2	RIICnSAR2	<RIICn_base> + 0030_H
	I^{2}C bus bit rate low-level register	RIICnBRL	<RIICn_base> + 0034_H
	I^{2}C bus bit rate high-level register	RIICnBRH	<RIICn_base> + 0038_H
	I^{2}C bus transmit data register	RIICnDRT	<RIICn_base> + $003C_H$
	I^{2}C bus receive data register	RIICnDRR	<RIICn_base> + 0040_H
	I^{2}C bus shift register	RIICnDRS	—

23.3.2 RIICnCR1 — I²C Bus Control Register 1

Access: RIICnCR1 register can be read or written in 32-bit units.

RIICnCR1L register can be read or written in 16-bit units.

RIICnCR1LL register can be read or written in 8-bit units.

Address: RIICnCR1: <RIICn_base> + 0000_H

RIICnCR1L: <RIICn_base> + 0000_H

RIICnCR1LL: <RIICn_base> + 0000_H

Value after reset: $0000\ 001F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	W	R/W	R/W	R	R

Table 23.11 RIICnCR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ICE	I²C Bus Interface Enable 0: Disabled (the RIICnSCL and RIICnSDA pins are not driven). 1: Enabled (the RIICnSCL and RIICnSDA pins are driven). (This bit selects an RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I²C Bus Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: Bits SCLO and SDAO can be written. 1: Bits SCLO and SDAO are protected. (This bit is read as 1.)
3	SCLO	SCL Output Control/Monitor ● Read: 0: The RIIC has driven the RIICnSCL pin low. 1: The RIIC has released the RIICnSCL pin. ● Write: 0: The RIIC drives the RIICnSCL pin low. 1: The RIIC releases the RIICnSCL pin.

Table 23.11 RIICnCR1 Register Contents

Bit Position	Bit Name	Function
2	SDAO	SDA Output Control/Monitor ● Read: 0: The RIIC has driven the RIICnSDA pin low. 1: The RIIC has released the RIICnSDA pin. ● Write: 0: The RIIC drives the RIICnSDA pin low. 1: The RIIC releases the RIICnSDA pin.
1	SCLI	SCL Line Monitor 0: RIICnSCL line is low. 1: RIICnSCL line is high.
0	SDAI	SDA Line Monitor 0: RIICnSDA line is low. 1: RIICnSDA line is high.

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the RIICnSDA and RIICnSCL signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see **Section 23.13.2, Extra SCL Clock Cycle Output Function**.

IICRST Bit (I^{2}C Bus Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 23.12, RIIC Resets** lists the types of RIIC reset.

The RIIC reset resets all registers (except ICE and IICRST) including the RIICnCR2.BBSY flag and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I^{2}C bus shift register (RIICnDRS), and the I^{2}C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 23.14, Reset Function of RIIC**.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error, etc.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 23.12 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

ICE Bit (I^{2}C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see **Table 23.12, RIIC Resets**.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 stops driving of the RIICnSCL and RIICnSDA pins.

CAUTION

Though the output from RIICnSDA or RIICnSCL is disabled while the ICE bit is 0, the input to RIICnSDA or RIICnSCL is enabled. The RIICnSCL and RIICnSDA pin functions should not be assigned to the RIIC. If assigned, it causes the slave addresses to be compared.

23.3.3 RIICnCR2 — I^{2}C Bus Control Register 2

Access: RIICnCR2 register can be read or written in 32-bit units.
RIICnCR2L register can be read or written in 16-bit units.
RIICnCR2LL register can be read or written in 8-bit units.

Address: RIICnCR2: <RIICn_base> + 0004$_H$
RIICnCR2L: <RIICn_base> + 0004$_H$
RIICnCR2LL: <RIICn_base> + 0004$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R

Table 23.13 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I^{2}C bus is released (the bus is free). 1: The I^{2}C bus is occupied (the bus is busy).
6	MST*[1]	Master/Slave Mode 0: Slave mode 1: Master mode
5	TRS*[1]	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. When the RIICnMR1.MTWP bit is set to 1, the MST and TRS bits can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see **Section 23.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).

Note that arbitration may be lost as the start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see **Section 23.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the RS bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see **Section 23.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode (1 or 0) by the following conditions.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)
- When 0 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode (1 or 0) by the following conditions.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I^{2}C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

CAUTION

- When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
- When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

23.3.4 RIICnMR1 — I^{2}C Bus Mode Register 1

Access:	RIICnMR1 register can be read or written in 32-bit units. RIICnMR1L register can be read or written in 16-bit units. RIICnMR1LL register can be read or written in 8-bit units.
Address:	RIICnMR1: <RIICn_base> + 0008$_H$ RIICnMR1L: <RIICn_base> + 0008$_H$ RIICnMR1LL: <RIICn_base> + 0008$_H$
Value after reset:	0000 0008$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 23.14 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MTWP	MST/TRS Write Protect 0: Disables writing to the RIICnCR2.MST and TRS bits. 1: Enables writing to the RIICnCR2.MST and TRS bits.
6 to 4	CKS[2:0]	Internal Reference Clock Selection (IICϕ) b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock
3	BCWP*1	BC Write Protect 0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.) 1: Protects the BC[2:0] bits.
2 to 0	BC[2:0]	Bit Counter b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames. When setting any value other than 000_B, set the value while the SCL line is at a low level.

[Clearing conditions]

- When 1 is written to the RIICnCR1.IICRST bit and a RIIC reset or internal reset is initiated.
- Data transfer including the acknowledge bit being completed.
- A start condition including a restart condition being detected.

23.3.5 RIICnMR2 — I^{2}C Bus Mode Register 2

Access: RIICnMR2 register can be read or written in 32-bit units.

RIICnMR2L register can be read or written in 16-bit units.

RIICnMR2LL register can be read or written in 8-bit units.

Address: RIICnMR2: <RIICn_base> + 000C$_H$

RIICnMR2L: <RIICn_base> + 000C$_H$

RIICnMR2LL: <RIICn_base> + 000C$_H$

Value after reset: 0000 0006$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 23.15 RIICnMR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1
6 to 4	SDDL[2:0]	SDA Output Delay Counter ● When RIICnMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles ● When RIICnMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles

Table 23.15 RIICnMR2 Register Contents

Bit Position	Bit Name	Function
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOH	Timeout H Count Control 0: Disables counting while the SCL line is at a high level. 1: Enables counting while the SCL line is at a high level.
1	TMOL	Timeout L Count Control 0: Disables counting while the SCL line is at a low level. 1: Enables counting while the SCL line is at a low level.
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see **Section 23.13.1, Timeout Function**.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see **Section 23.7, Facility for Delaying SDA Output**.

CAUTION

Set the SDA output delay time to meet the I^{2}C bus standard (within the data enable time/acknowledge enable time*[1]). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [Fm])

23.3.6 RIICnMR3 — I^{2}C Bus Mode Register 3

Access: RIICnMR3 register can be read or written in 32-bit units.

RIICnMR3L register can be read or written in 16-bit units.

RIICnMR3LL register can be read or written in 8-bit units.

Address: RIICnMR3: <RIICn_base> + 0010$_H$

RIICnMR3L: <RIICn_base> + 0010$_H$

RIICnMR3LL: <RIICn_base> + 0010$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 23.16 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	WAIT*2	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS*2	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP*1	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT*1	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Digital noise Filter Stage Selection b1 b0 0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Digital noise Filter Stage Selection)

These bits are used to select the number of stages of the digital noise filter.

CAUTION

Set the noise range to be filtered out by the noise filter within a range less than the SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock synchronized (IICϕ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit after ACKWP reading while the ACKWP bit is set to 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

CAUTION

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

CAUTION

When the value of the WAIT bit is cleared to 0, be sure to read the RIICnDRR beforehand.

23.3.7 RIICnFER — I^{2}C Bus Function Enable Register

Access:	RIICnFER register can be read or written in 32-bit units. RIICnFERL register can be read or written in 16-bit units. RIICnFERLL register can be read or written in 8-bit units.
Address:	RIICnFER: <RIICn_base> + 0014$_H$ RIICnFERL: <RIICn_base> + 0014$_H$ RIICnFERLL: <RIICn_base> + 0014$_H$
Value after reset:	0000 0072$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.17 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) If 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode, the TRS bit is not cleared.
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see **Section 23.13.1, Timeout Function**.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to select enabling or disabling of the arbitration-lost detection function. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether the detection of ACK during transmission of NACK in reception (such as when slaves with the same address are present on the bus and each is transmitting different data, or when two or more masters select the same slave device simultaneously with different numbers of bytes for reception) is judged to represent a loss in arbitration.

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with a rising or falling edge on the SCL line. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

23.3.8 RIICnSER — I^{2}C Bus Status Enable Register

Access:	RIICnSER register can be read or written in 32-bit units. RIICnSERL register can be read or written in 16-bit units. RIICnSERLL register can be read or written in 8-bit units.
Address:	RIICnSER: <RIICn_base> + 0018_H RIICnSERL: <RIICn_base> + 0018_H RIICnSERLL: <RIICn_base> + 0018_H
Value after reset:	0000 0009_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 23.18 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000$_B$ + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100$_B$) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 23.9.3, Device-ID Address Detection**.

23.3.9 RIICnIER — I^{2}C Bus Interrupt Enable Register

Access: RIICnIER register can be read or written in 32-bit units.
RIICnIERL register can be read or written in 16-bit units.
RIICnIERLL register can be read or written in 8-bit units.

Address: RIICnIER: <RIICn_base> + 001C$_H$
RIICnIERL: <RIICn_base> + 001C$_H$
RIICnIERLL: <RIICn_base> + 001C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.19 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt request (INTRIICnTI) is disabled. 1: Transmit data empty interrupt request (INTRIICnTI) is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt request (INTRIICnTEI) is disabled. 1: Transmit end interrupt request (INTRIICnTEI) is enabled.
5	RIE	Receive Complete Interrupt Enable 0: Receive complete interrupt request (INTRIICnRI) is disabled. 1: Receive complete interrupt request (INTRIICnRI) is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the RIICnSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the RIICnSR2.AL flag is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the RIICnSR2.START flag is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the RIICnSR2.STOP flag is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the RIICnSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Complete Interrupt Enable)

This bit is used to enable or disable receive complete interrupt requests (INTRIICnRI) when the RIICnSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTRIICnTEI) when the RIICnSR2.TEND flag is set to 1. An INTRIICnTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTRIICnTI) when the RIICnSR2.TDRE flag is set to 1.

23.3.10 RIICnSR1 — I^{2}C Bus Status Register 1

Access: RIICnSR1 register can be read or written in 32-bit units.
RIICnSR1L register can be read or written in 16-bit units.
RIICnSR1LL register can be read or written in 8-bit units.

Address: RIICnSR1: <RIICn_base> + 0020$_H$
RIICnSR1L: <RIICn_base> + 0020$_H$
RIICnSR1LL: <RIICn_base> + 0020$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(/W)*1	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 23.20 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0_B + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100_B) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address (1111 100_B) plus 1[R] has matched while the setting of the RIICnSER.DIDE bit is 1 (device ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100_B)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100_B) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

23.3.11 RIICnSR2 — I^{2}C Bus Status Register 2

Access: RIICnSR2 register can be read or written in 32-bit units.
RIICnSR2L register can be read or written in 16-bit units.
RIICnSR2LL register can be read or written in 8-bit units.

Address: RIICnSR2: <RIICn_base> + 0024_H
RIICnSR2L: <RIICn_base> + 0024_H
RIICnSR2LL: <RIICn_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 23.21 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.

TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (the RIICnCR2.BBSY flag is 1) in master mode (the RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (the RIICnSR1 register is not 00_H) and the bus is busy (the RIICnCR2.BBSY bit is 1) in slave mode (the RIICnCR2.MST bit is 0).
- The bus is free (the RIICnCR2.BBSY flag is 0) while generation of a start condition is requested (the RIICnCR2.ST bit is 1).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 23.22 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

RIICnFER			RIICnSR2	Error	Arbitration-Lost Generation Source
MALE	NALE	SALE	AL		
1	—	—	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
—	1	—	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
—	—	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

Remark: —: Don’t care

START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Complete)

[Setting conditions]

- When receive data has been transferred from RIICnDRS to RIICnDRR
  At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches the address enabled in RIICnSER after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0
- In master mode, transition to master reception while the R/W# bit appended to the slave address is set to 1

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition is detected
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When the received slave address matches the address enabled in RIICnSER after a start condition including a restart condition is detected with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

23.3.12 RIICnSARy — I^{2}C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy register can be read or written in 32-bit units.

RIICnSARyL register can be read or written in 16-bit units.

RIICnSARyLL and RIICnSARyLH registers can be read or written in 8-bit units.

Address: RIICnSAR0: <RIICn_base> + 0028$_H$

RIICnSAR0L: <RIICn_base> + 0028$_H$

RIICnSAR0LL: <RIICn_base> + 0028$_H$,
RIICnSAR0LH: <RIICn_base> + 0029$_H$

RIICnSAR1: <RIICn_base> + 002C$_H$

RIICnSAR1L: <RIICn_base> + 002C$_H$

RIICnSAR1LL: <RIICn_base> + 002C$_H$,
RIICnSAR1LH: <RIICn_base> + 002D$_H$

RIICnSAR2: <RIICn_base> + 0030$_H$

RIICnSAR2L: <RIICn_base> + 0030$_H$

RIICnSAR2LL: <RIICn_base> + 0030$_H$,
RIICnSAR2LH: <RIICn_base> + 0031$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—	SVA[9:1]									SVA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.23 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. ● When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are valid and form a 7-bit slave address. ● When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. ● When the FSy bit is 0 (7-bit address format), this bit is invalid. ● When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

23.3.13 RIICnBRL — I^{2}C Bus Bit Rate Low-Level Register

Access:	RIICnBRL register can be read or written in 32-bit units. RIICnBRLL register can be read or written in 16-bit units. RIICnBRLLL register can be read or written in 8-bit units.
Address:	RIICnBRL: <RIICn_base> + 0034_H RIICnBRLL: <RIICn_base> + 0034_H RIICnBRLLL: <RIICn_base> + 0034_H
Value after reset:	$0000\ 00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 23.24 RIICnBRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see **Section 23.10, Automatic Low-Hold Function for SCL**); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.

RIICnBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

Note 1. Data setup time (t_{SU}: DAT)
250 ns (up to 100 kbps: standard mode [Sm])
100 ns (up to 400 kbps: fast mode [Fm])

23.3.14 RIICnBRH — I^{2}C Bus Bit Rate High-Level Register

Access: RIICnBRH register can be read or written in 32-bit units.

RIICnBRHL register can be read or written in 16-bit units.

RIICnBRHLL register can be read or written in 8-bit units.

Address: RIICnBRH: <RIICn_base> + 0038$_H$

RIICnBRHL: <RIICn_base> + 0038$_H$

RIICnBRHLL: <RIICn_base> + 0038$_H$

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 23.25 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

The I^{2}C transfer rate and the SCL clock duty are calculated using the following expression.

(1) When RIICnFER.SCLE = 0
Transfer rate = $1/\{[(\text{RIICnBRH} + 1) + (\text{RIICnBRL} + 1)]/\text{IIC}\phi^{*1} + tr + tf\}$
Duty cycle = $\{tr + [(\text{RIICnBRH} + 1)/\text{IIC}\phi]\}/\{tr + tf + [(\text{RIICnBRH} + 1) + (\text{RIICnBRL} + 1)]/\text{IIC}\phi\}$

(2) When RIICnFER.SCLE = 1, RIICnFER.NFE = 0, IICϕ = PCLK
Transfer rate = $1/\{[(\text{RIICnBRH} + 3) + (\text{RIICnBRL} + 3)]/\text{IIC}\phi^{*1} + tr + tf\}$
Duty cycle = $\{tr + [(\text{RIICnBRH} + 3)/\text{IIC}\phi]\}/\{tr + tf + [(\text{RIICnBRH} + 3) + (\text{RIICnBRL} + 3)]/\text{IIC}\phi\}$

(3) When RIICnFER.SCLE = 1, RIICnFER.NFE = 1, IICϕ = PCLK
Transfer rate = $1/\{[(\text{RIICnBRH} + 3 + nf) + (\text{RIICnBRL} + 3 + nf)]/\text{IIC}\phi^{*1} + tr + tf\}$
Duty cycle = $\{tr + [(\text{RIICnBRH} + 3 + nf)/\text{IIC}\phi]\}/\{tr + tf + [(\text{RIICnBRH} + 3 + nf) + (\text{RIICnBRL} + 3 + nf)]/\text{IIC}\phi\}$

(4) RIICnFER.SCLE = 1, RIICnFER.NFE = 0, IICϕ < PCLK
Transfer rate = $1/\{[(\text{RIICnBRH} + 2) + (\text{RIICnBRL} + 2)]/\text{IIC}\phi^{*1} + tr + tf\}$
Duty cycle = $\{tr + [(\text{RIICnBRH} + 2)/\text{IIC}\phi]\}/\{tr + tf + [(\text{RIICnBRH} + 2) + (\text{RIICnBRL} + 2)]/\text{IIC}\phi\}$

(5) When RIICnFER.SCLE = 1, RIICnFER.NFE = 1, $\mathrm{IIC}\phi < \mathrm{PCLK}$

$\text{Transfer rate} = 1/\{[(\mathrm{RIICnBRH} + 2 + nf) + (\mathrm{RIICnBRL} + 2 + nf)\ ]/\mathrm{IIC}\phi^{*1} + tr + tf\}$

$\text{Duty cycle} = \{tr + [(\mathrm{RIICnBRH} + 2 + nf)/\ \mathrm{IIC}\phi]\}/\{\ tr + tf + [(\mathrm{RIICnBRH} + 2 + nf) + (\mathrm{RIICnBRL} + 2 + nf)]/\mathrm{IIC}\phi\}$

tf: SCL line falling time [ns]*2

tr: SCL line rising time [ns]*2

nf: Digital noise filter stage

Duty cycle: 0% < Duty < 100%

Note 1. As for IICϕ, see CKS[2:0] in **Section 23.3.4, RIICnMR1 — I2C Bus Mode Register 1**.

Note 2. The SCL line rising time [tr] and SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I²C bus standard from NXP Semiconductors.

Table 23.26, Examples of RIICnBRH/RIICnBRLSettings for Transfer Rate lists examples of the RIICnBRH and RIICnBRL register settings when the SCL synchronization circuit is not used.

Table 23.26 Examples of RIICnBRH/RIICnBRLSettings for Transfer Rate

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	8					10					12.5				
	CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL	
		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	100_B	22	$F6_H$	25	$F9_H$	101_B	13	ED_H	15	EF_H	101_B	16	$F0_H$	20	$F4_H$
50	010_B	16	$F0_H$	19	$F3_H$	010_B	21	$F5_H$	24	$F8_H$	011_B	12	EC_H	15	EF_H
100	001_B	15	EF_H	18	$F2_H$	001_B	19	$F3_H$	23	$F7_H$	001_B	24	$F8_H$	29	FD_H
400	000_B	4	$E4_H$	10	EA_H	000_B	5	$E5_H$	12	EC_H	000_B	7	$E7_H$	16	$F0_H$

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	16					20					25				
	CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL	
		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	101_B	22	$F6_H$	25	$F9_H$	110_B	13	ED_H	15	EF_H	110_B	16	$F0_H$	20	$F4_H$
50	011_B	16	$F0_H$	19	$F3_H$	011_B	21	$F5_H$	24	$F8_H$	100_B	12	EC_H	15	EF_H
100	010_B	15	EF_H	18	$F2_H$	010_B	19	$F3_H$	23	$F7_H$	010_B	24	$F8_H$	29	FD_H
400	000_B	9	$E9_H$	20	$F4_H$	000_B	11	EB_H	25	$F9_H$	001_B	7	$E7_H$	16	$F0_H$

	PCLK Frequency (MHz)									
	30					33				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
Transfer Rate (kbps)	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	110_B	20	$F4_H$	24	$F8_H$	110_B	22	$F6_H$	26	FA_H
50	100_B	15	EF_H	18	$F2_H$	100_B	17	$F1_H$	20	$F4_H$
100	011_B	14	EE_H	17	$F1_H$	011_B	16	$F0_H$	19	$F3_H$
400	001_B	8	$E8_H$	19	$F3_H$	001_B	9	$E9_H$	21	$F5_H$

CAUTION

CBRH/ICBRL settings in these tables are calculated using the following values:
SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I^{2}C bus standard from NXP Semiconductors.

23.3.15 RIICnDRT — I^{2}C Bus Transmit Data Register

Access: RIICnDRT register can be read or written in 32-bit units.
RIICnDRTL register can be read or written in 16-bit units.
RIICnDRTLL register can be read or written in 8-bit units.

Address: RIICnDRT: <RIICn_base> + 003C$_H$
RIICnDRTL: <RIICn_base> + 003C$_H$
RIICnDRTLL: <RIICn_base> + 003C$_H$

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I^{2}C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTRIICnTI) request is generated. When writing to bit 31 to 8, write the value after reset.

23.3.16 RIICnDRR — I^{2}C Bus Receive Data Register

Access: RIICnDRR register is a read-only register that can be read in 32-bit units.

RIICnDRRL register is a read-only register that can be read in 16-bit units.

RIICnDRRLL register is a read-only register that can be read in 8-bit units.

Address: RIICnDRR: <RIICn_base> + 0040$_H$

RIICnDRRL: <RIICn_base> + 0040$_H$

RIICnDRRLL: <RIICn_base> + 0040$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I^{2}C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive complete interrupt (INTRIICnRI) request is generated.

If RIIC receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

23.3.17 RIICnDRS — I^{2}C Bus Shift Register

Access: This register is not accessible.

Address: —

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is a shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

23.4 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive complete, transmit data empty, and transmit end.

Table 23.27, Interrupt Sources lists details of the several interrupt requests. The receive complete and transmit data empty interrupt request are both capable of launching data transfer by the DMA.

Table 23.27 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMA Launching	Interrupt Condition
INTRIICnTI	Transmit Data Empty	TDRE	Possible	TDRE = 1 and TIE = 1
INTRIICnTEI	Transmit End	TEND	Not possible	TEND = 1 and TEIE = 1
INTRIICnRI	Receive Complete	RDRF	Possible	RDRF = 1 and RIE = 1
INTRIICnEE	Transfer Error/Event Generation	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1

Clear or mask the each flag during interrupt handling.

CAUTIONS

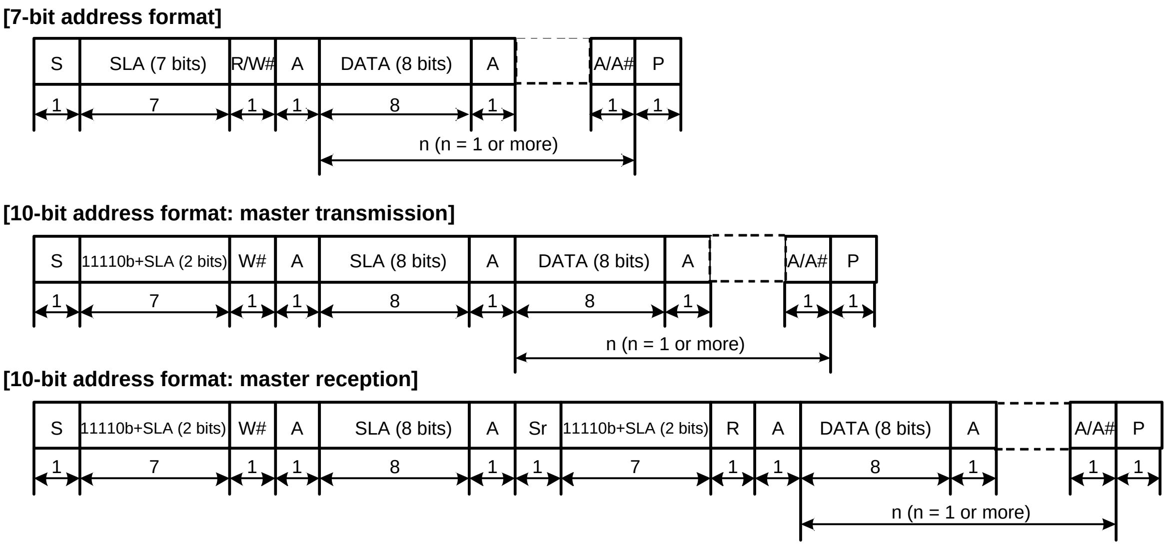
1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since INTRIICnTI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.TDRE flag (a condition for INTRIICnTI) is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).
3. Since INTRIICnRI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.RDRF flag (a condition for INTRIICnRI) is automatically cleared to 0 when data are read from RIICnDRR.
4. When using the INTRIICnTEI interrupt, clear the RIICnSR2.TEND flag in the INTRIICnTEI interrupt processing. Note that the RIICnSR2.TEND flag is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).

23.5 Operation

23.5.1 Communication Data Format

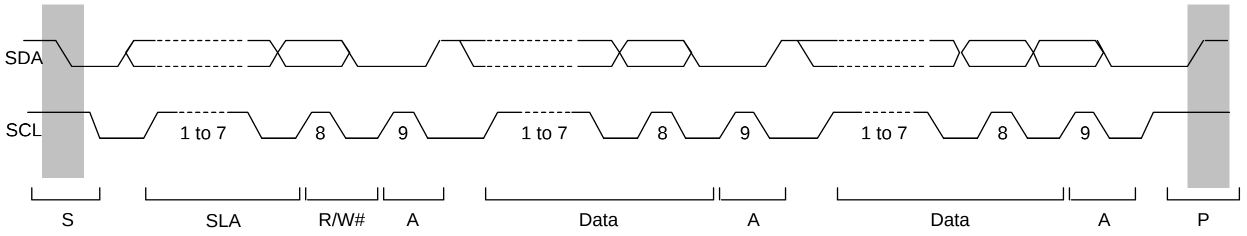
The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 23.3, I2C Bus Format shows the I²C bus format, and **Figure 23.4, I2C Bus Timing (SLA = 7 Bits)** shows the I²C bus timing.



Note: n: Number of transfer frames

Figure 23.3 I²C Bus Format



Note: S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

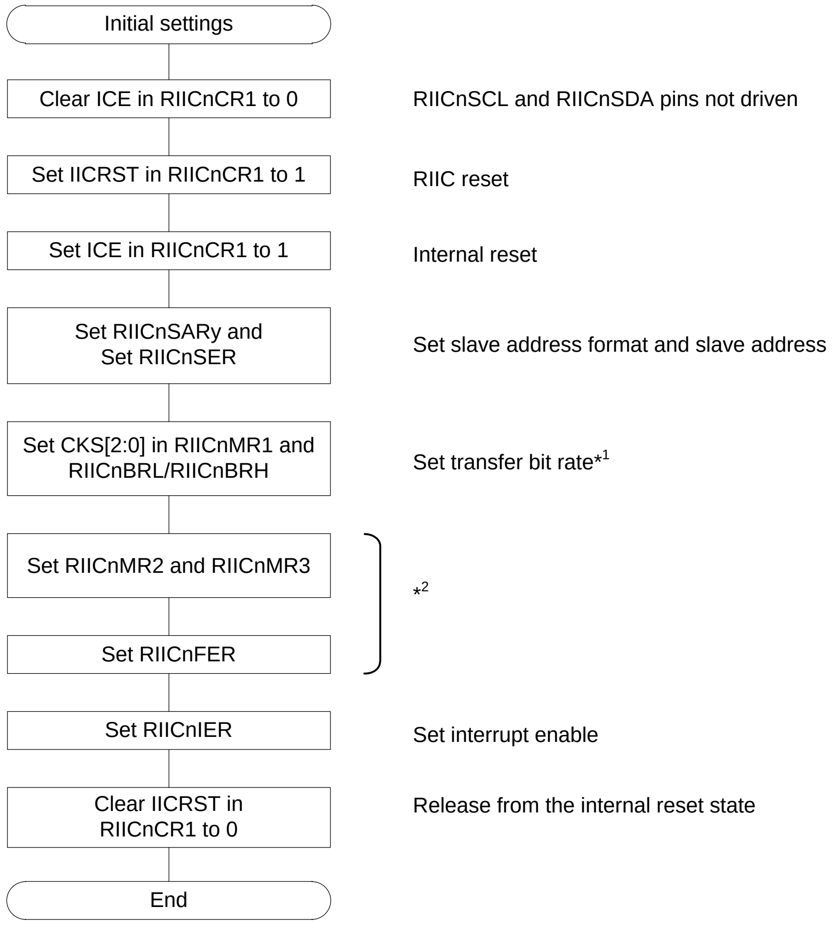
DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

Figure 23.4 I²C Bus Timing (SLA = 7 Bits)

23.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 23.5, Example of RIIC Initialization Flowchart**. Make initial settings for the RIIC once when starting the RIIC.



Note: y = 0 to 2

Note 1. When the RIIC is used only in slave mode, set the RIICnBRL register to a value longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 23.5 Example of RIIC Initialization Flowchart

23.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 23.6, Example of Master Transmission Flowchart** shows an example of usage of master transmission and **Figure 23.7, Master Transmit Operation Timing (1) (7-Bit Address Format)** to **Figure 23.9, Master Transmit Operation Timing (3)** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 23.5, Example of RIIC Initialization Flowchart**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS and MST bits to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0_B, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After all bytes of data for transmission have been written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

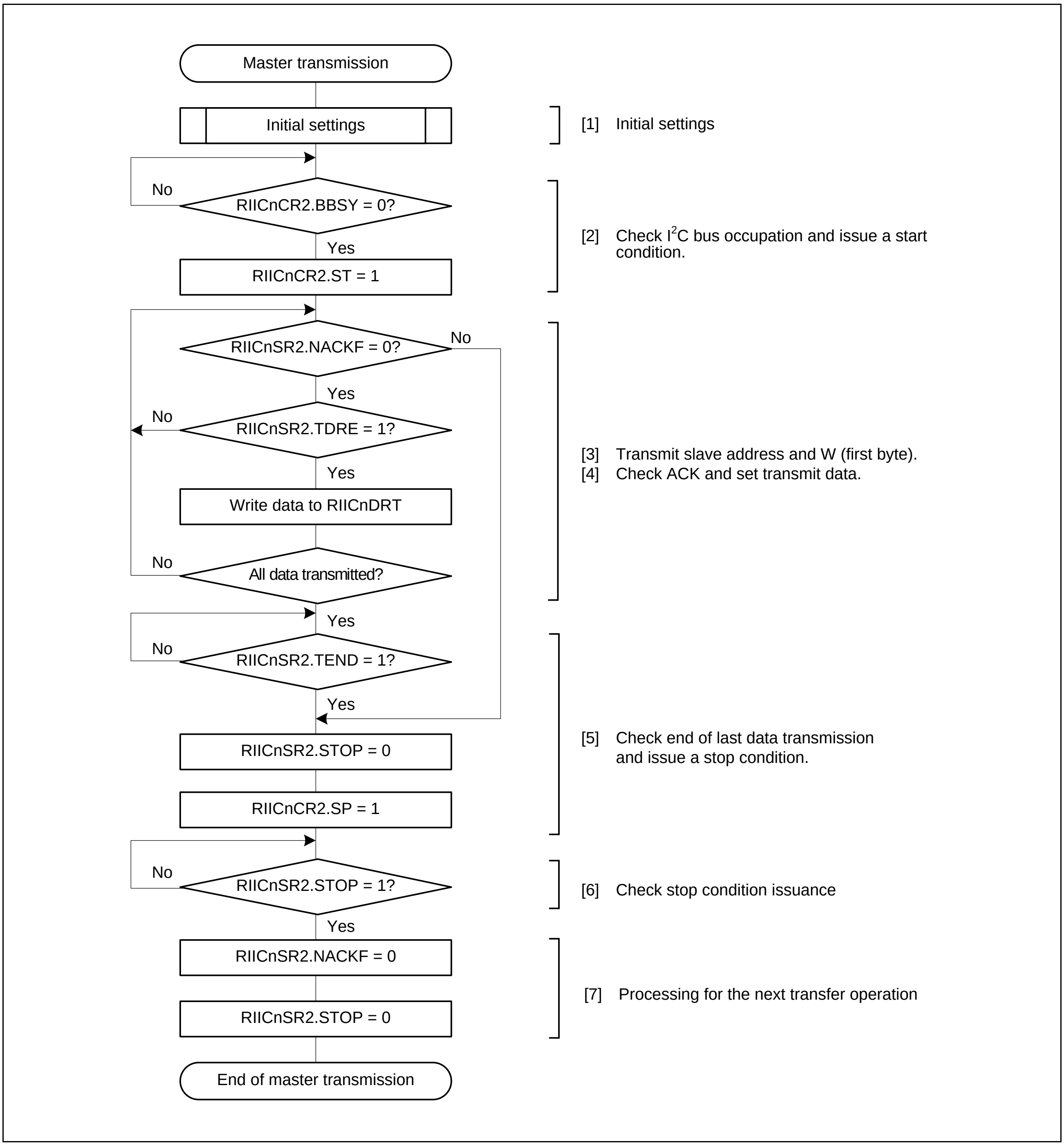


Figure 23.6 Example of Master Transmission Flowchart

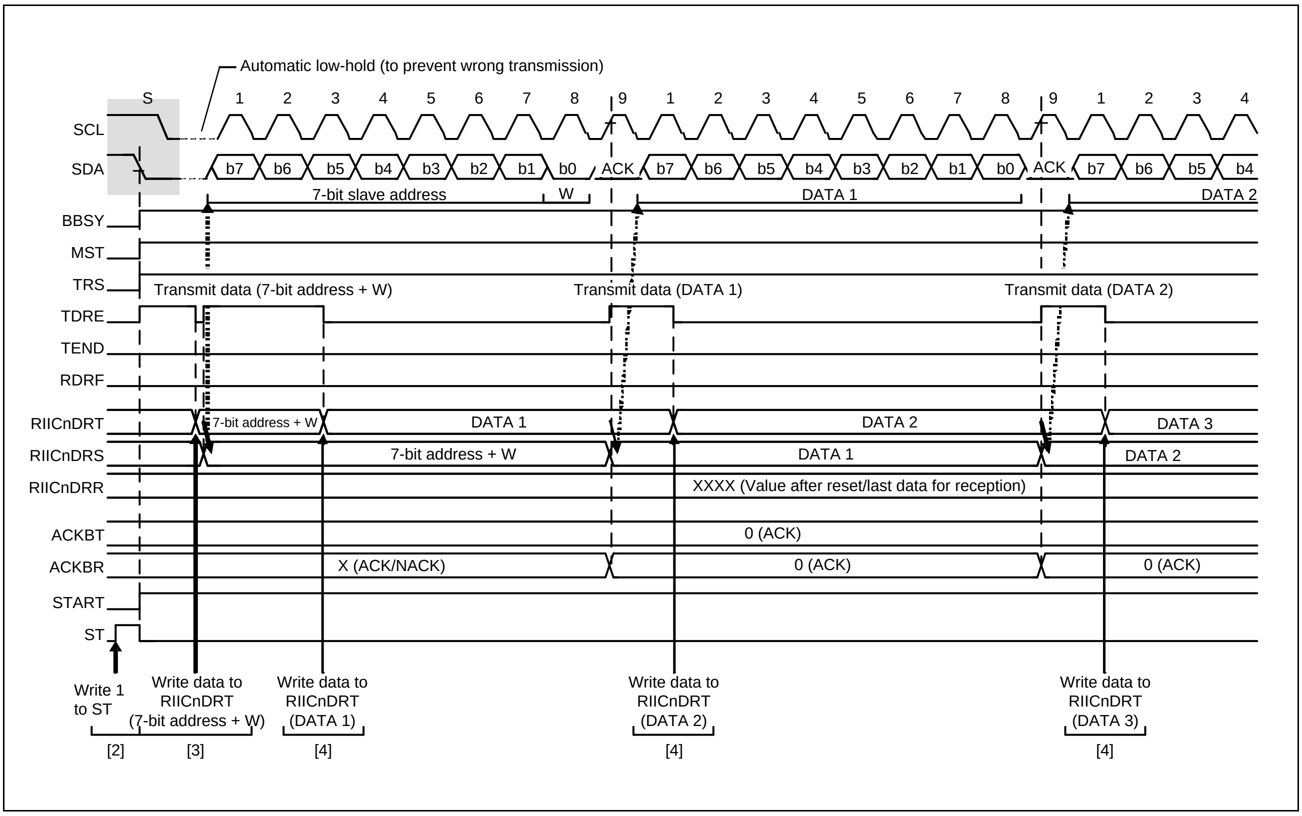


Figure 23.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

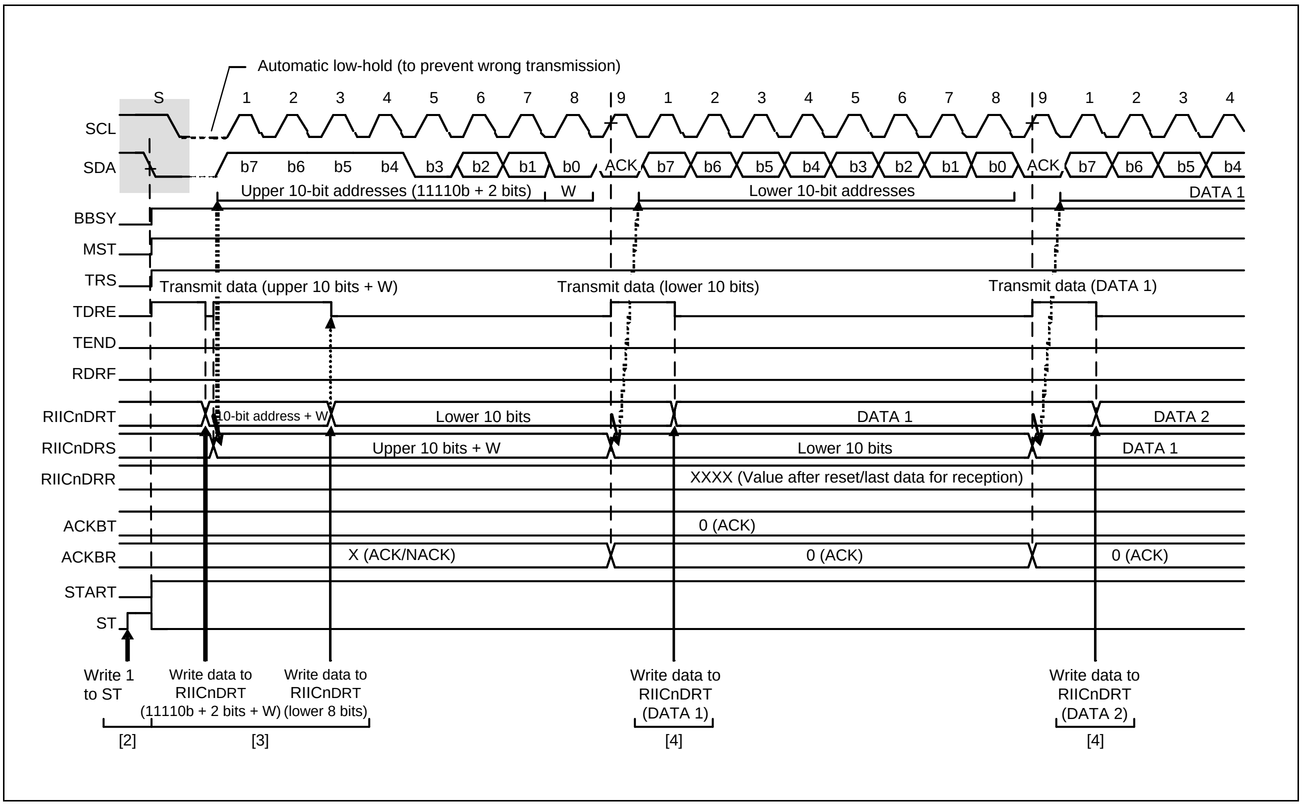


Figure 23.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

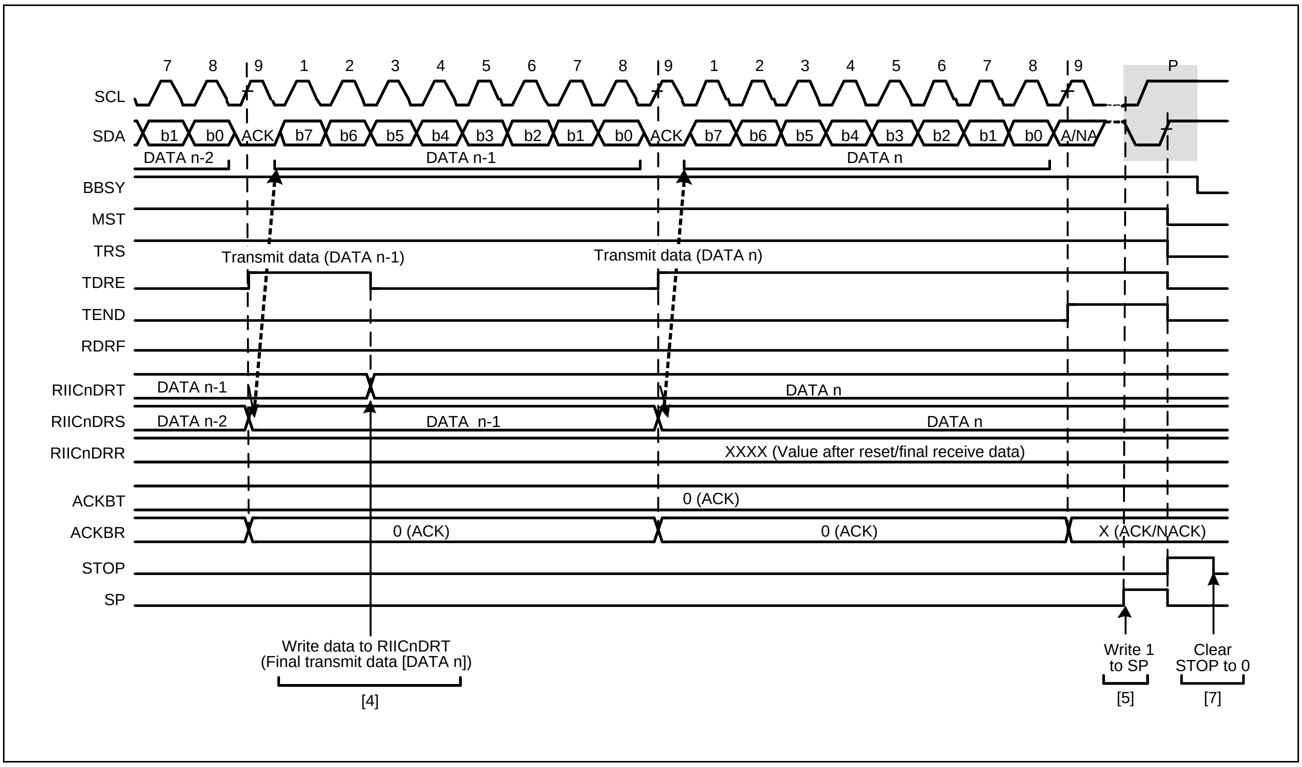


Figure 23.9 Master Transmit Operation Timing (3)

23.5.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 23.10, Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes) shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 23.11, Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 23.12, Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)** to **Figure 23.14, Master Receive Operation Timing (3) (when RDRFS = 0)** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 23.5, Example of RIIC Initialization Flowchart**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to transmit the two higher-order bits of the slave address and then the eight lower-order bits of the slave address, and issue a restart condition following generation of the transmission end interrupt (or after TEND = 1) (see **Figure 23.13, Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)** for operation timing). After that, transmitting 1111 0_B plus the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

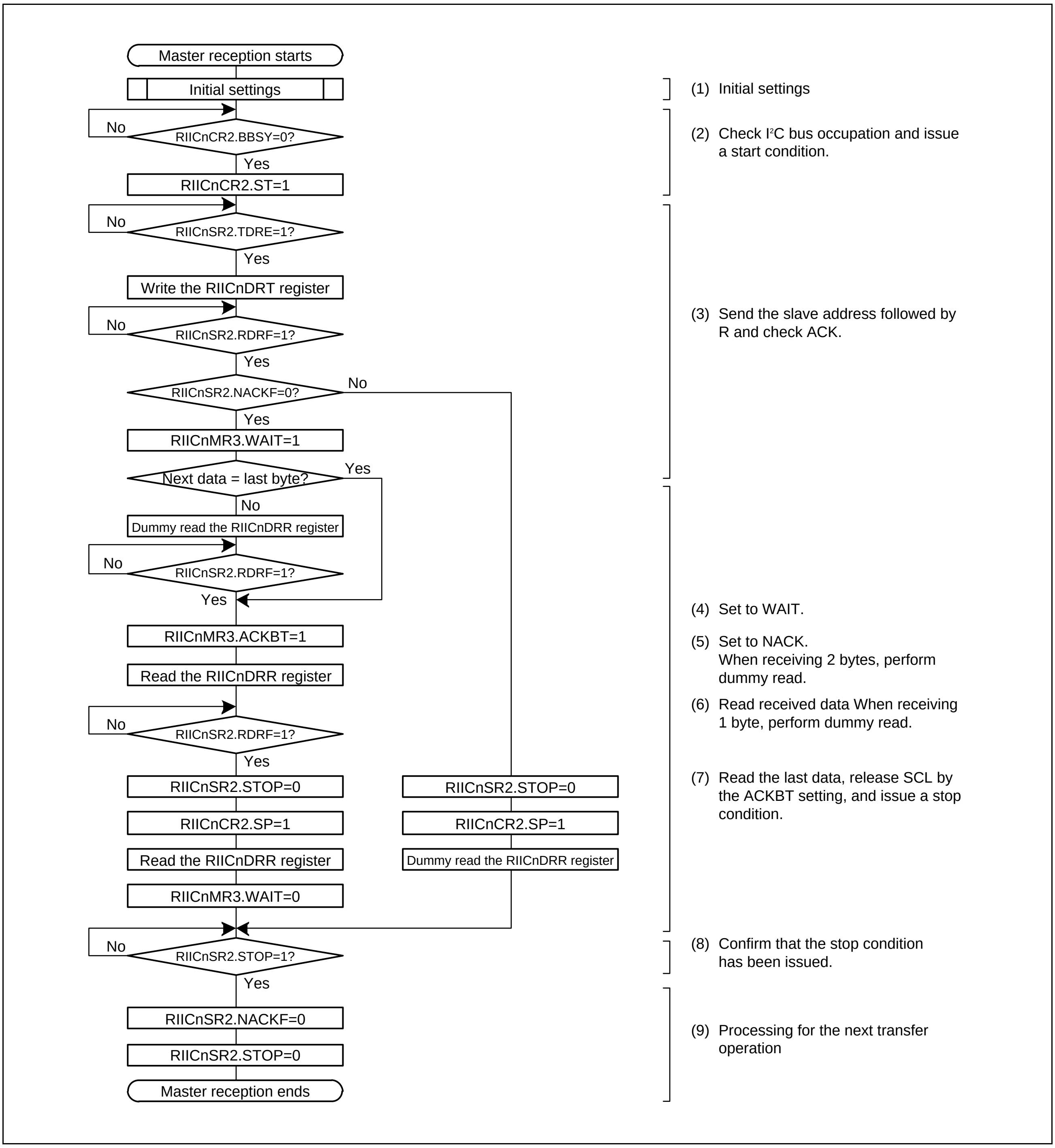


Figure 23.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)

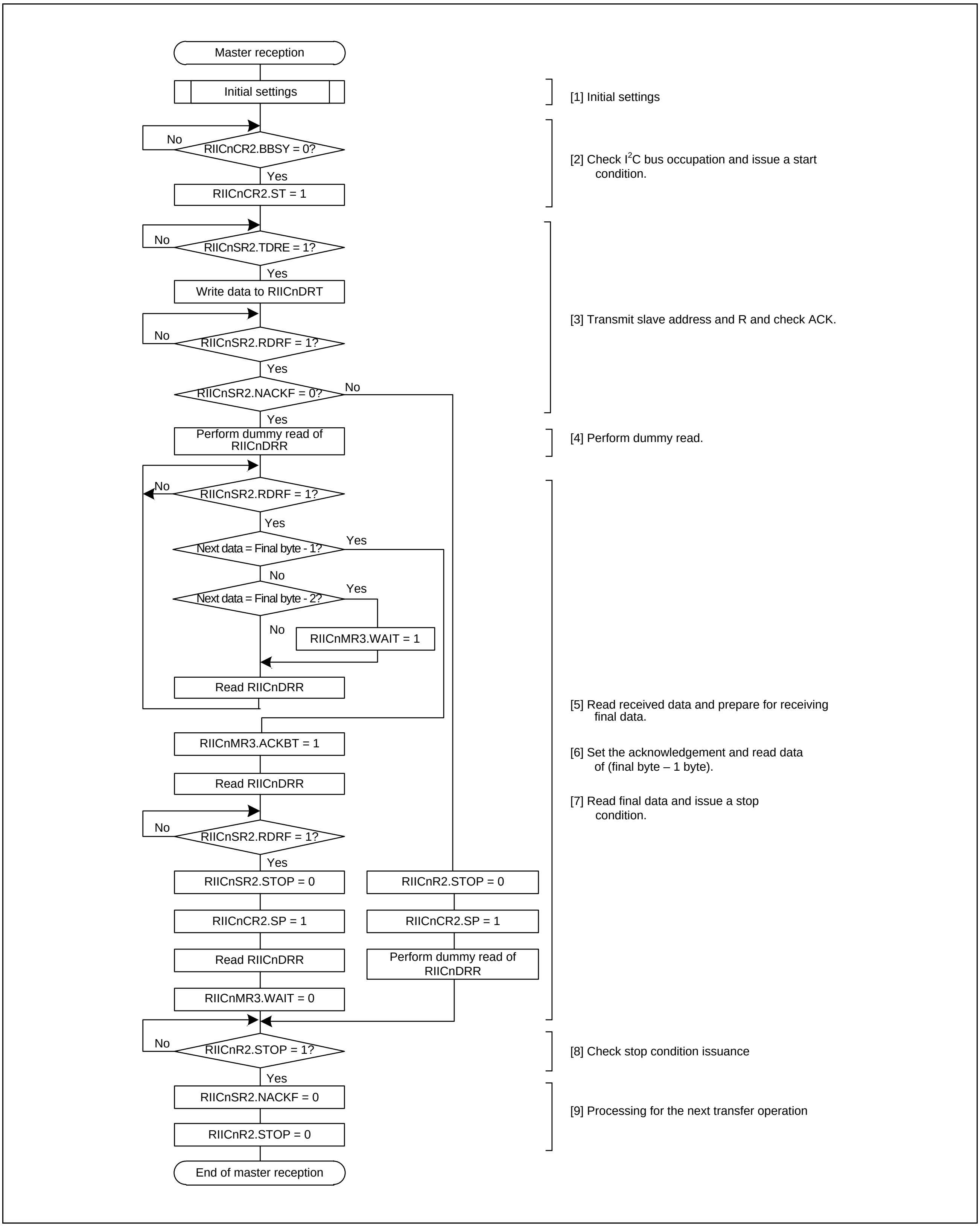


Figure 23.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

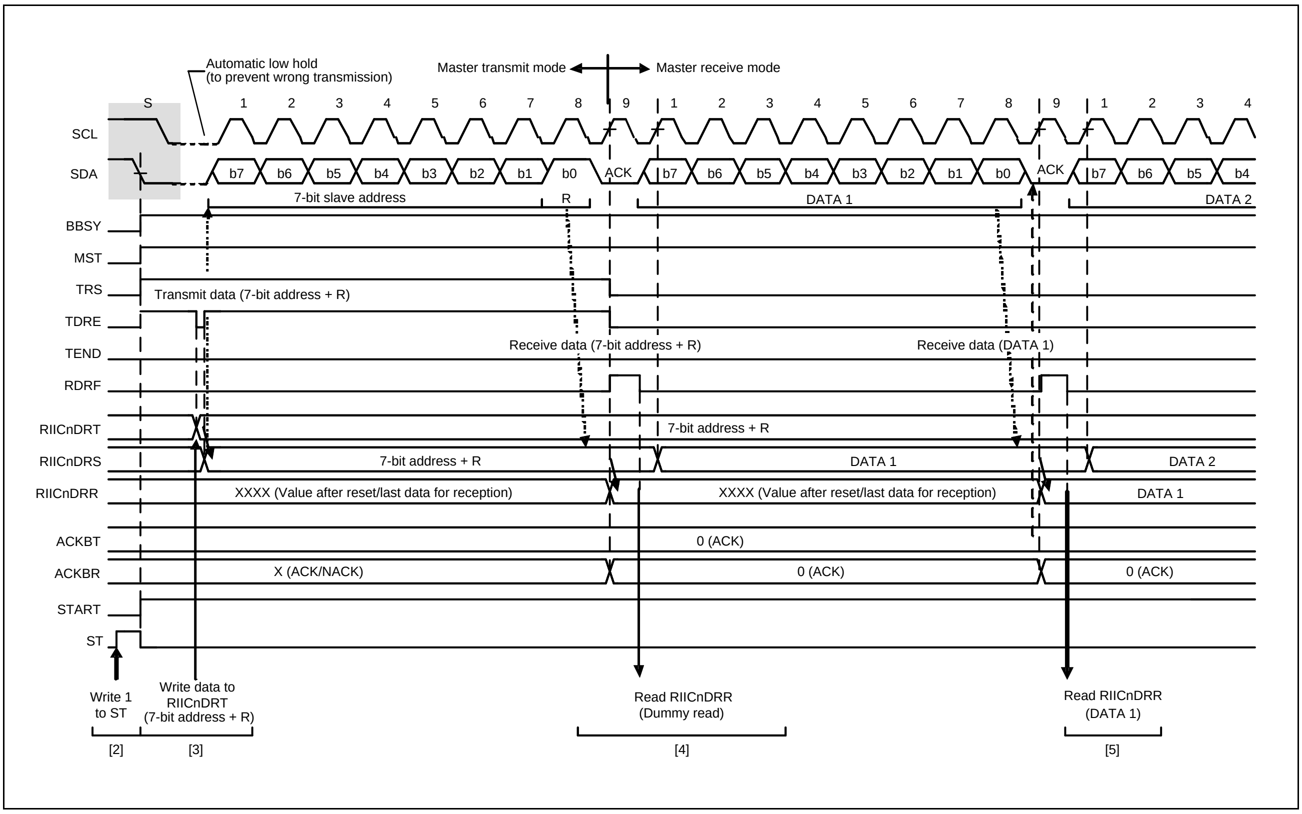


Figure 23.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

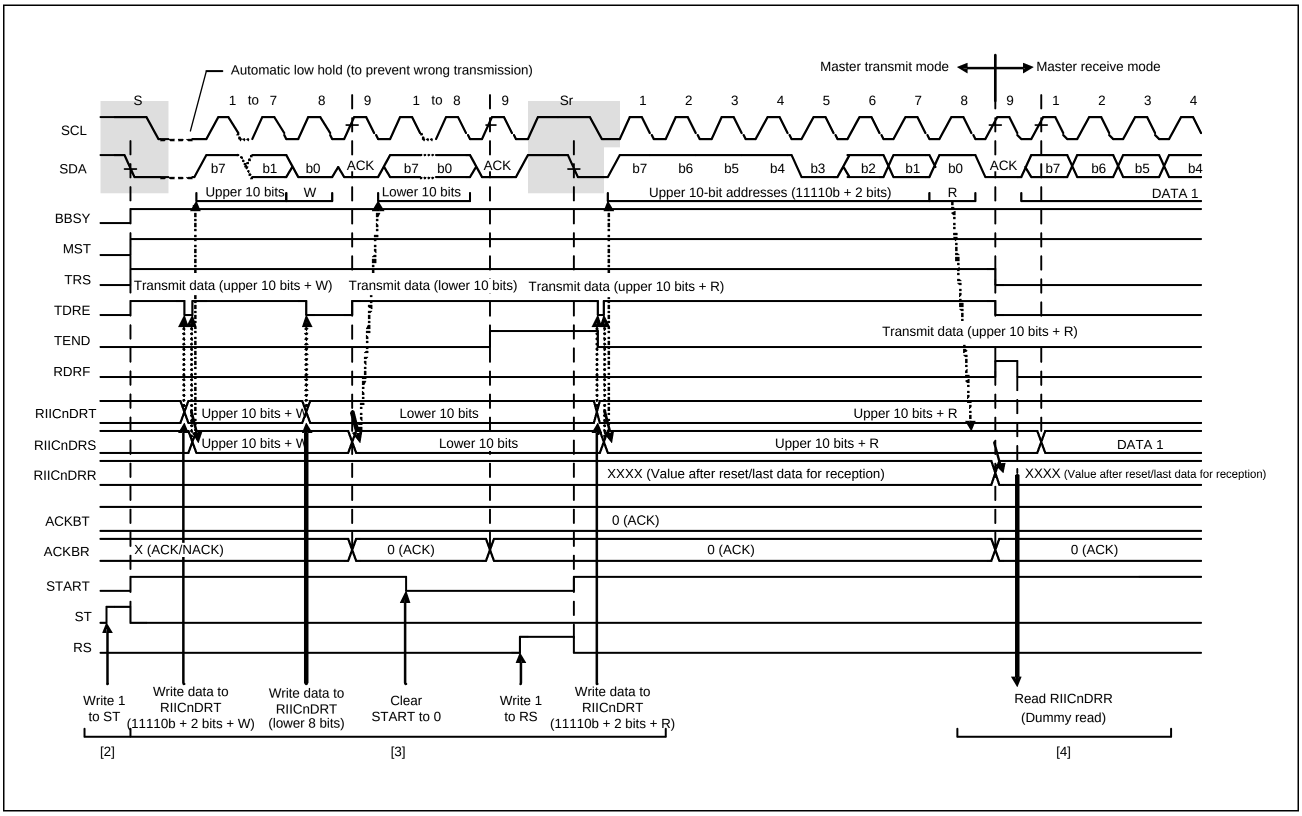


Figure 23.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

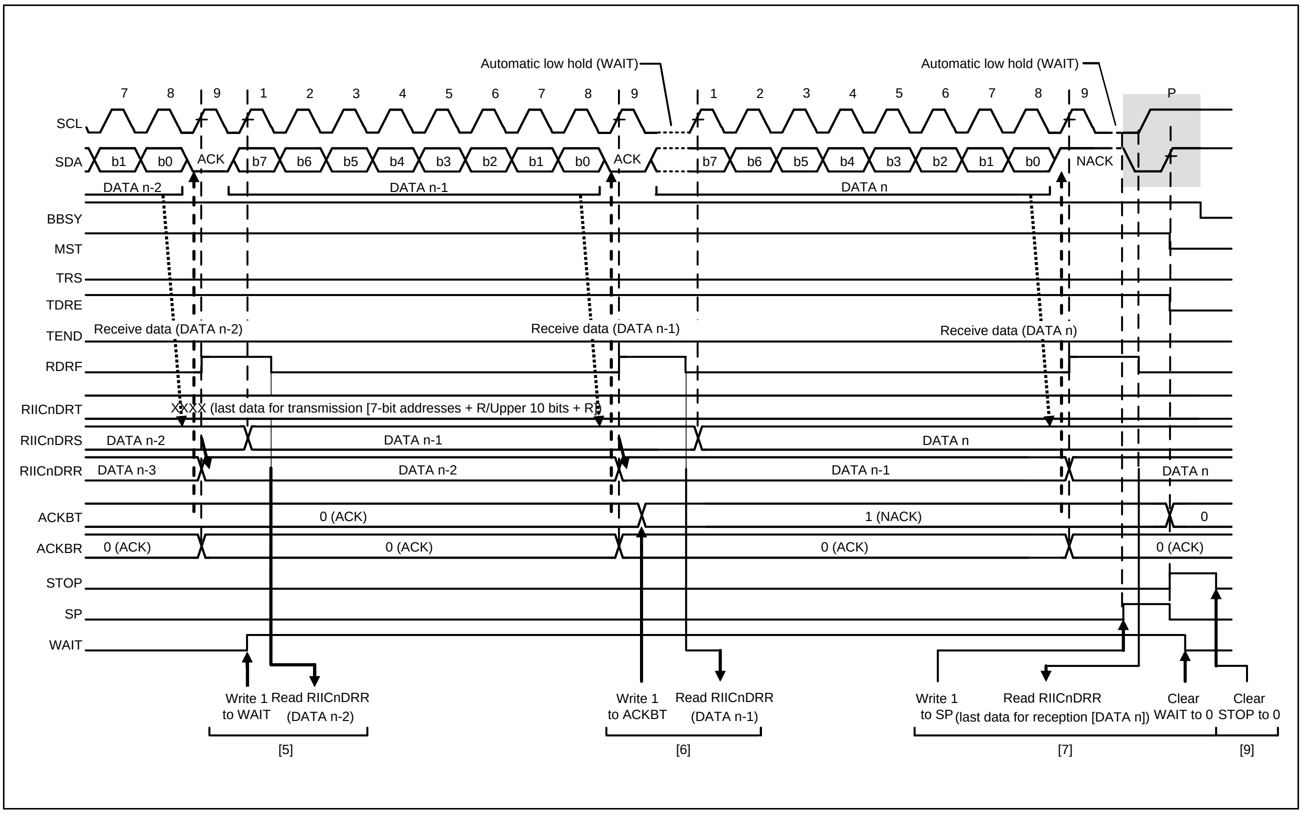


Figure 23.14 Master Receive Operation Timing (3) (when RDRFS = 0)

23.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 23.15, Example of Slave Transmission Flowchart shows an example of usage of slave transmission and **Figure 23.16, Slave Transmit Operation Timing (1) (7-Bit Address Format)** and **Figure 23.17, Slave Transmit Operation Timing (2)** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 23.5, Example of RIIC Initialization Flowchart** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

(3) After the RIICnSR2.TDRE flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

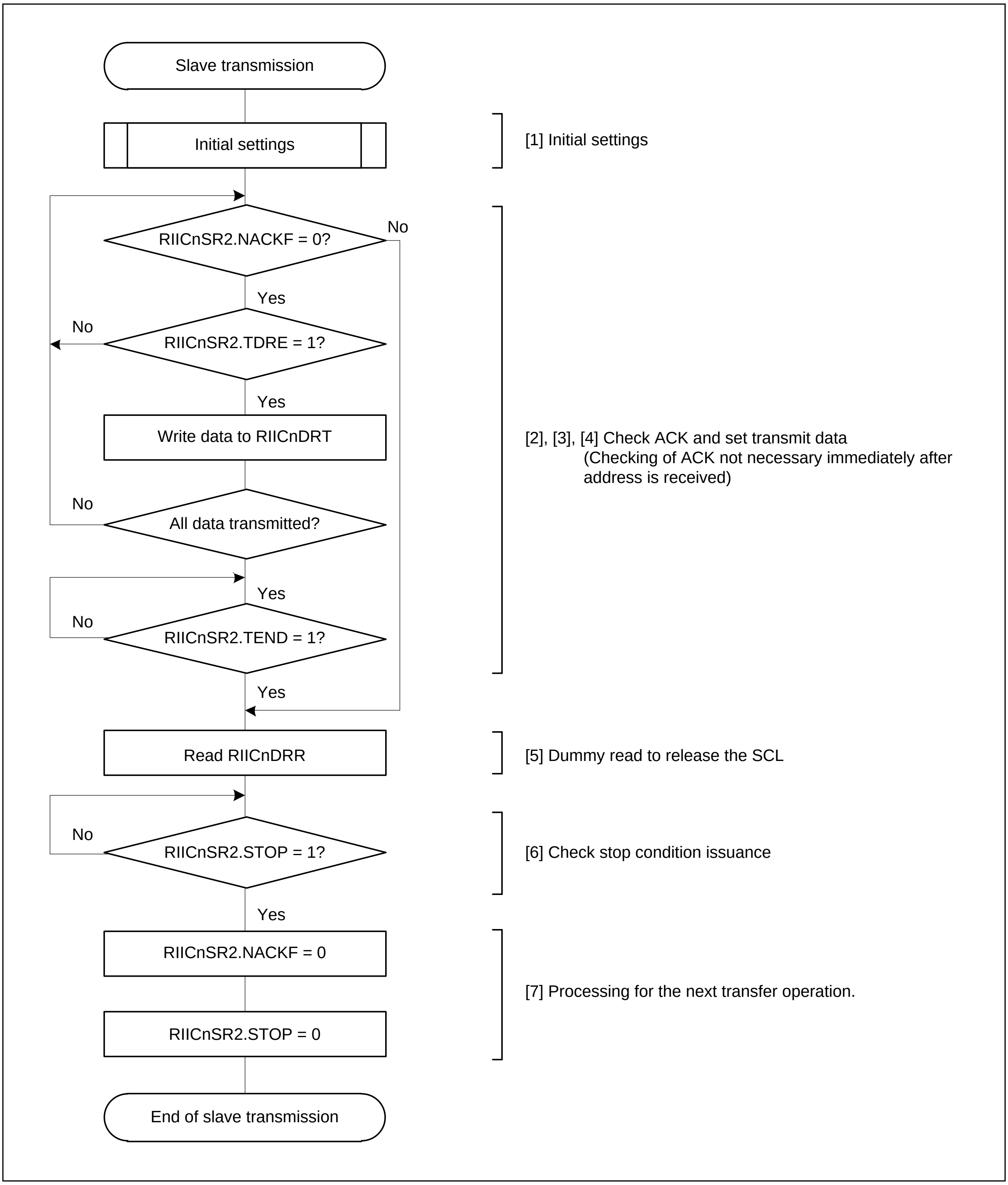


Figure 23.15 Example of Slave Transmission Flowchart

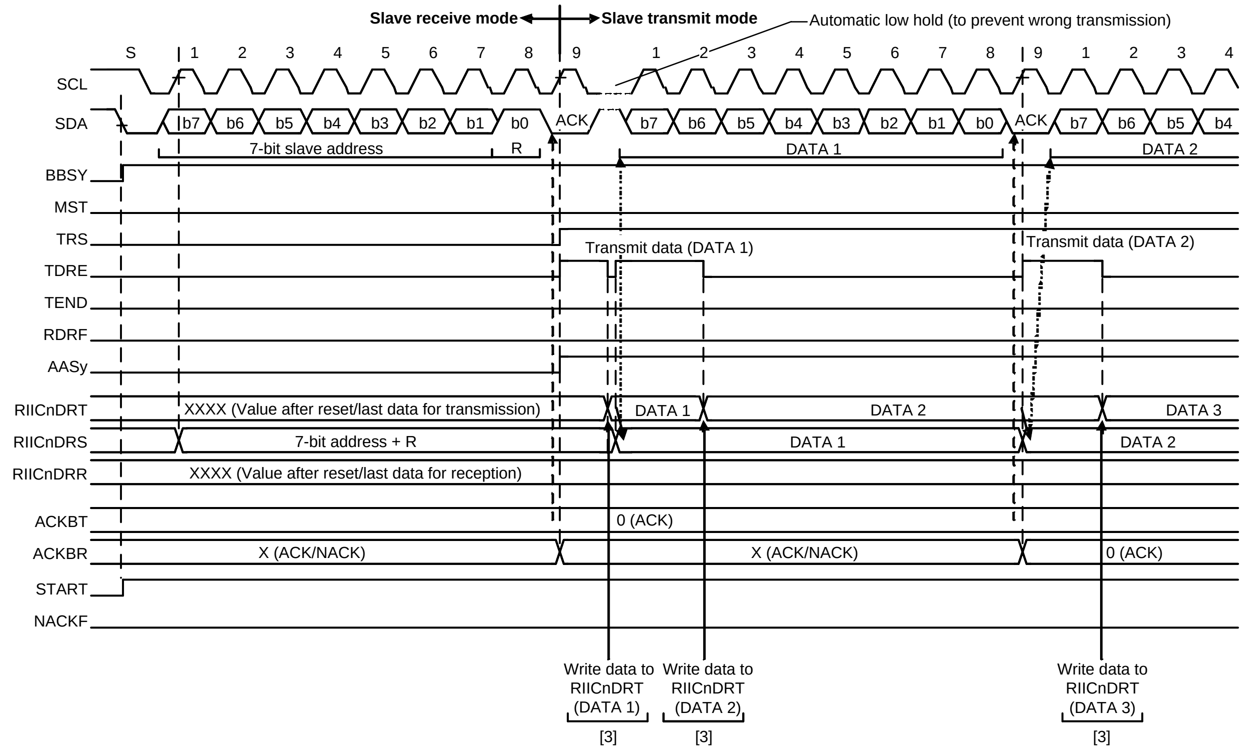


Figure 23.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

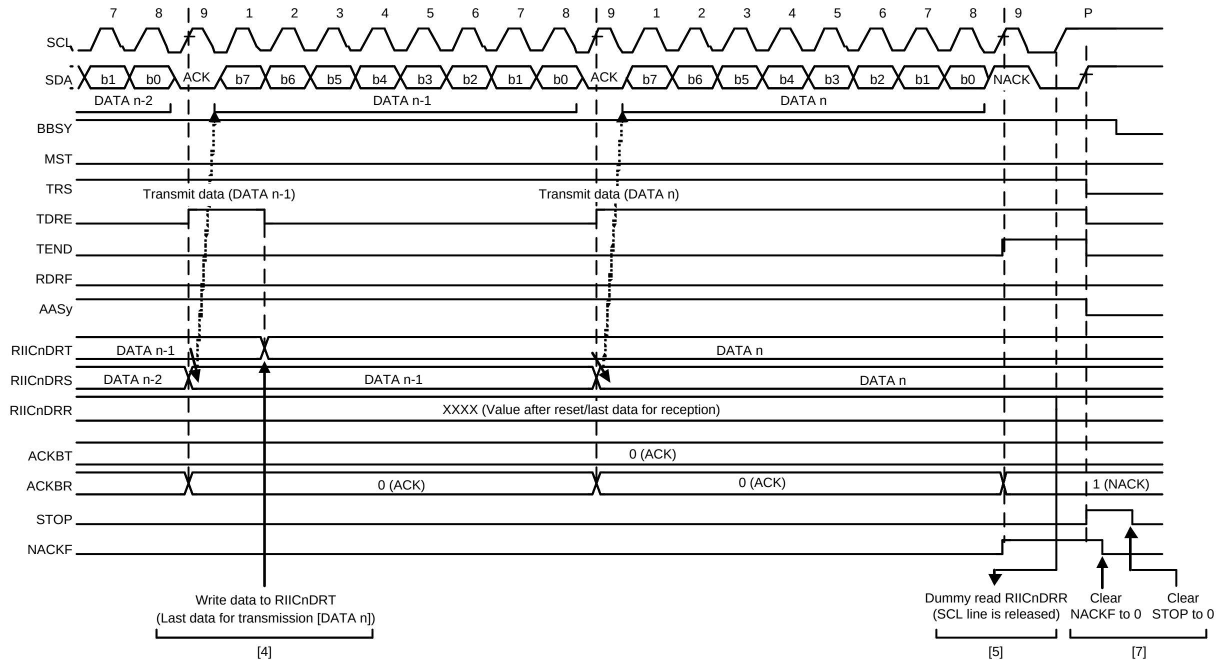


Figure 23.17 Slave Transmit Operation Timing (2)

23.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 23.18, Example of Slave Reception Flowchart shows an example of usage of slave reception and **Figure 23.19, Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)** and **Figure 23.20, Slave Receive Operation Timing (2) (when RDRFS = 0)** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 23.5, Example of RIIC Initialization Flowchart** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1. GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR as the first read operation (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 0. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

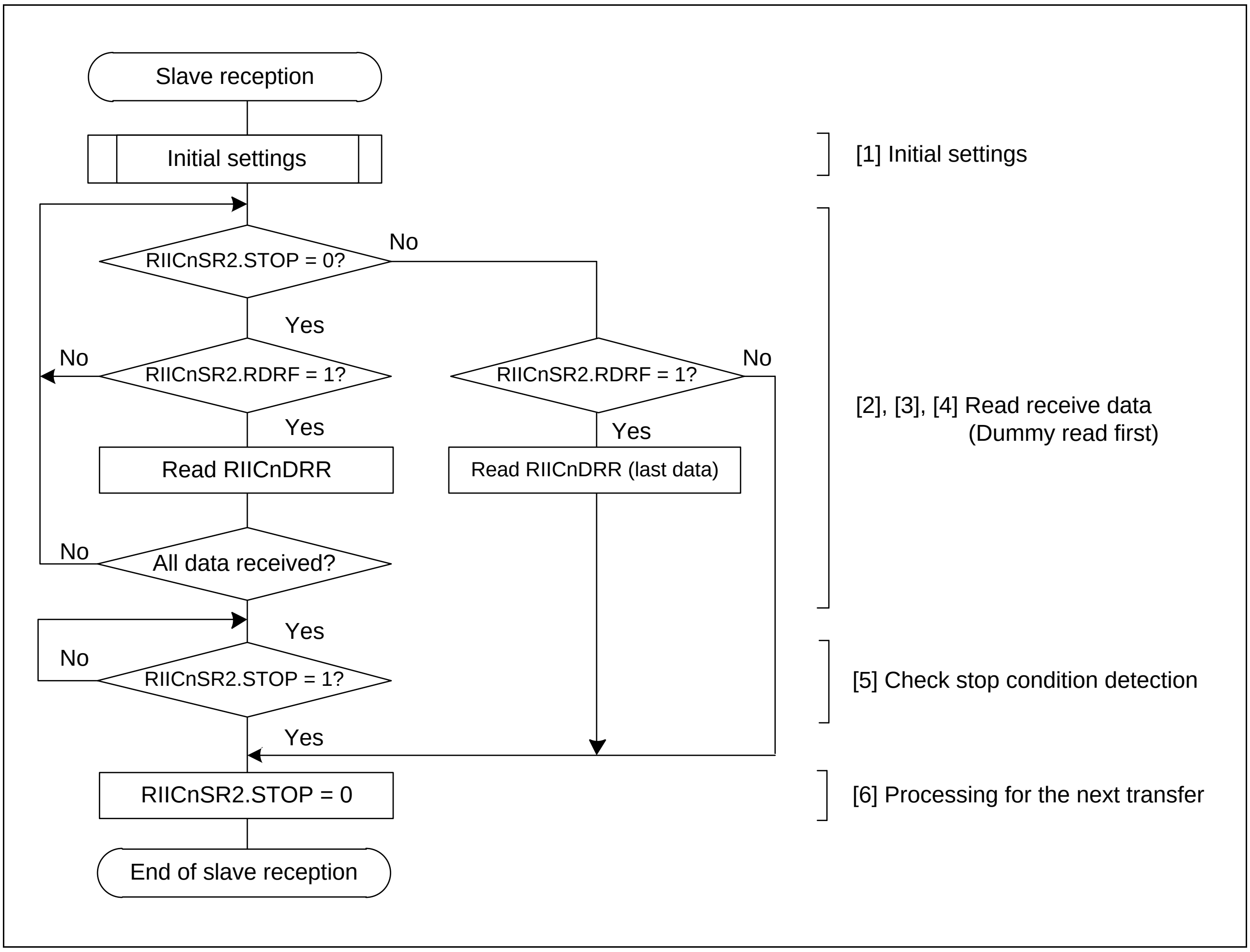


Figure 23.18 Example of Slave Reception Flowchart

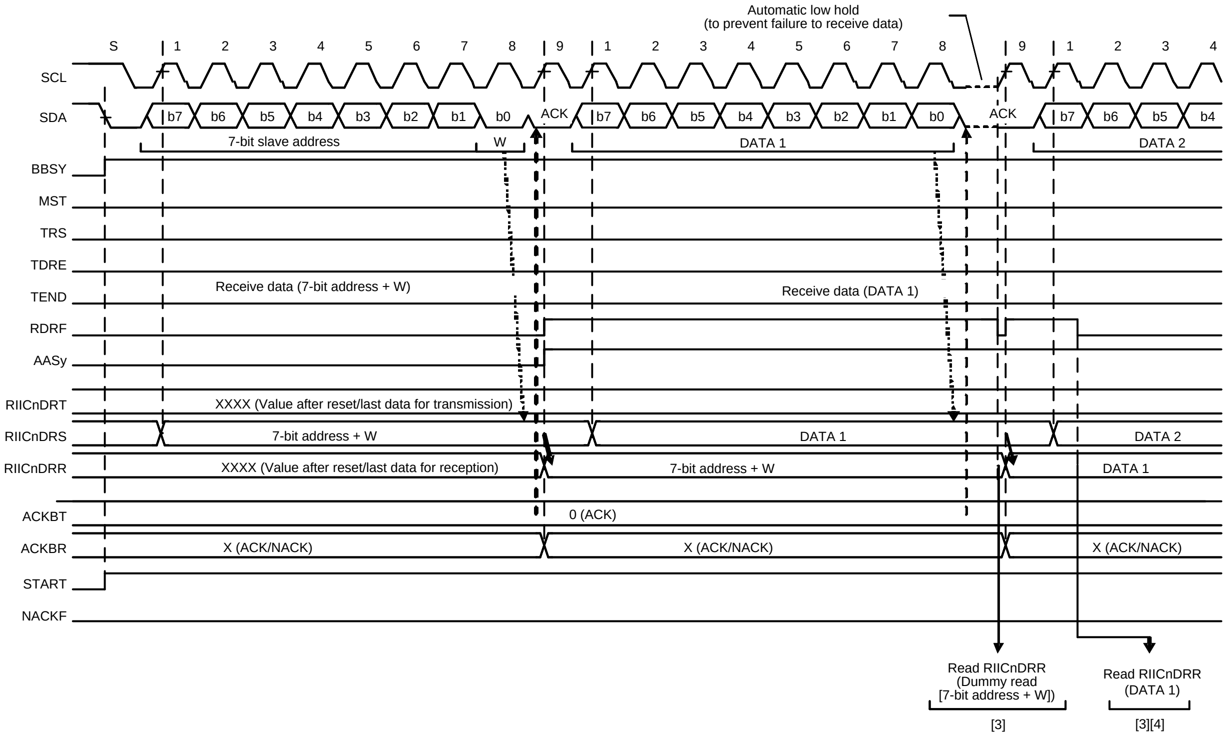


Figure 23.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

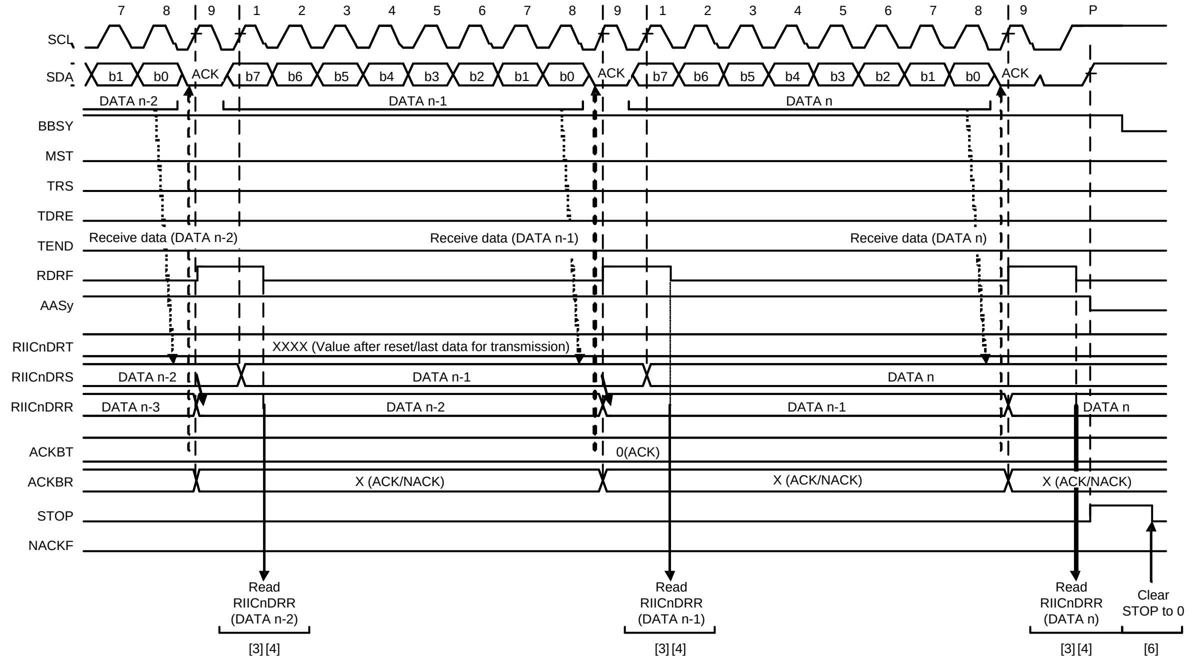


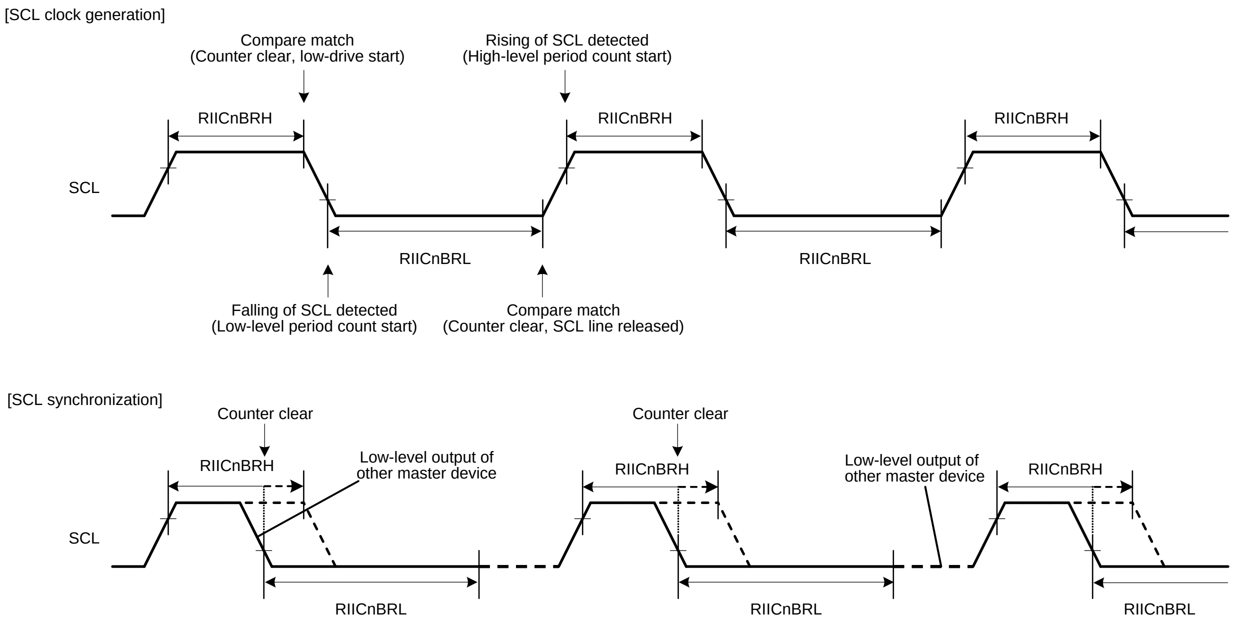
Figure 23.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

23.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I^{2}C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.



Note: RIICnBRH: I^{2}C bus bit rate high-level register (SCL clock high-level period counter)
RIICnBRL: I^{2}C bus bit rate low-level register (SCL clock low-level period counter)

Figure 23.21 Generation and Synchronization of the SCL Signal from the RIIC

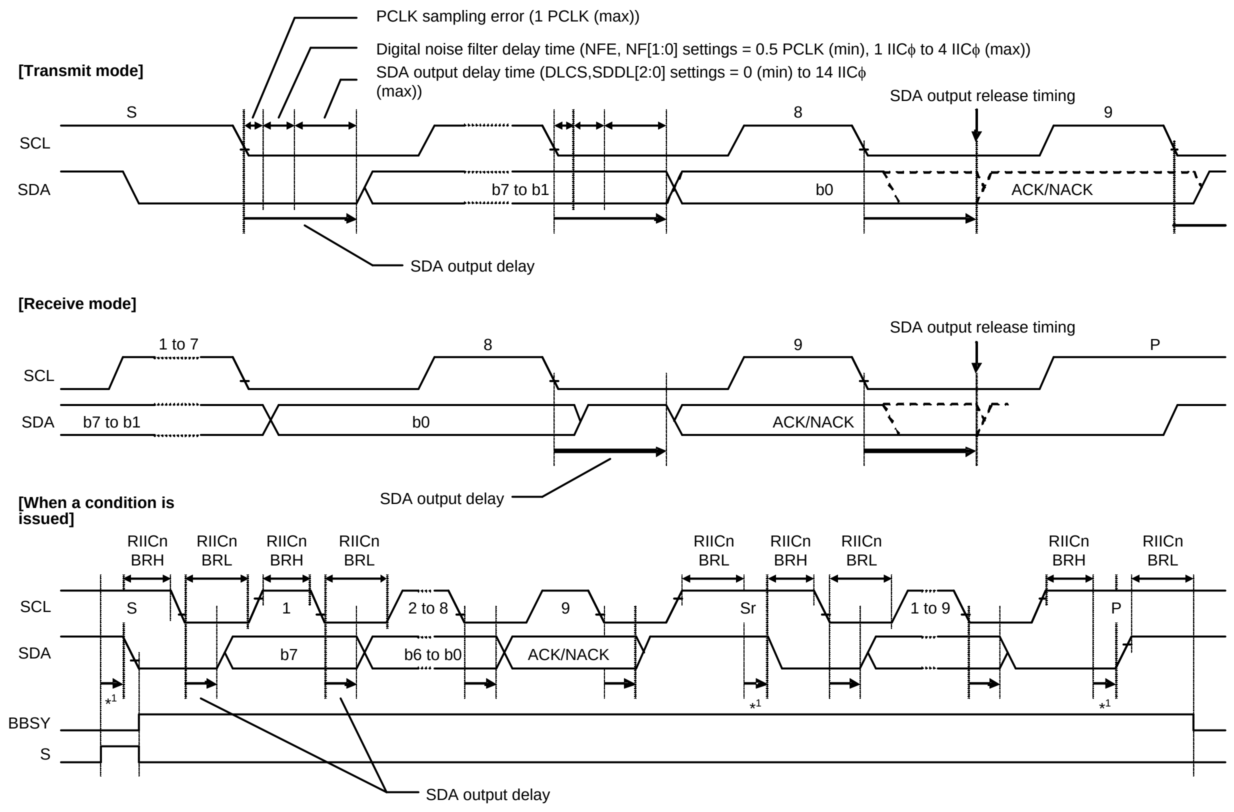
23.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000_B, and disabled by setting the same bits to 000_B.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits are set to any value other than 000_B), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay facility is set by the SDDL[2:0] bits when a start (S), restart (Sr), or stop (P) condition is issued.

Figure 23.22 SDA Output Delay Facility

23.8 Digital Noise-Filter Circuits

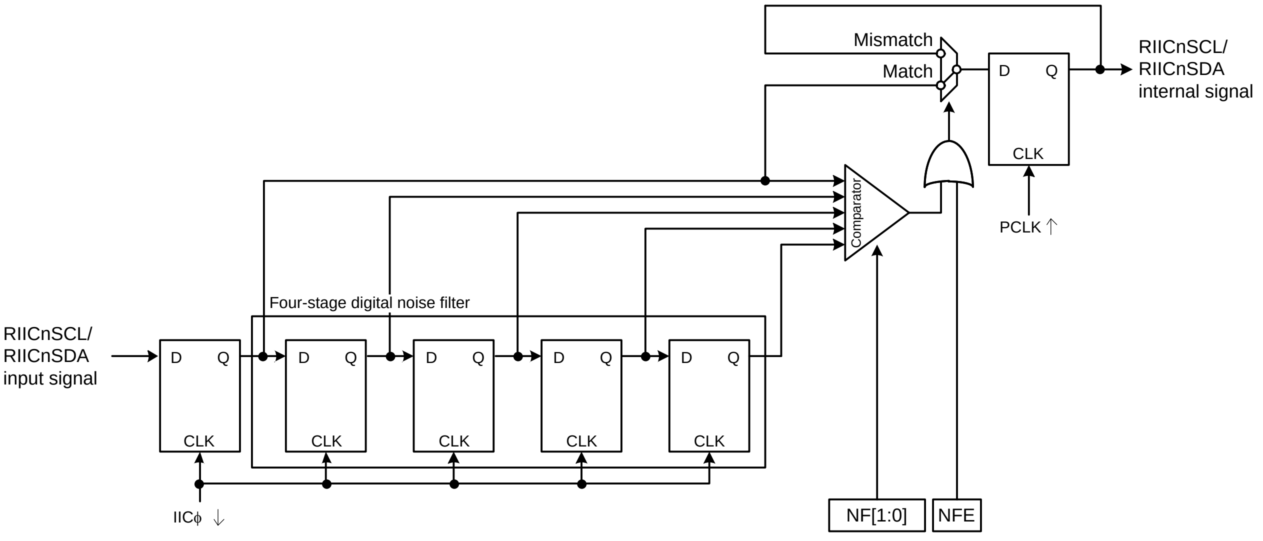
The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 23.23, Block Diagram of Digital Noise Filter Circuit** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed as an internal signal. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small, the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the RIICnFER.NFE bit to 0).



Note: NFE: Digital noise filter circuit enable bit
NF[1:0]: Digital noise filter stage selection bit

Figure 23.23 Block Diagram of Digital Noise Filter Circuit

23.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and device ID address, and also can set 7-bit or 10-bit slave addresses.

23.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. The RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive complete interrupt (INTRIICnRI) or transmit data empty interrupt (INTRIICnTI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 23.24, AASy Flag Set Timing with 7-Bit Address Format Selected to **Figure 23.26, AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed** show the AASy flag set timing in three cases.

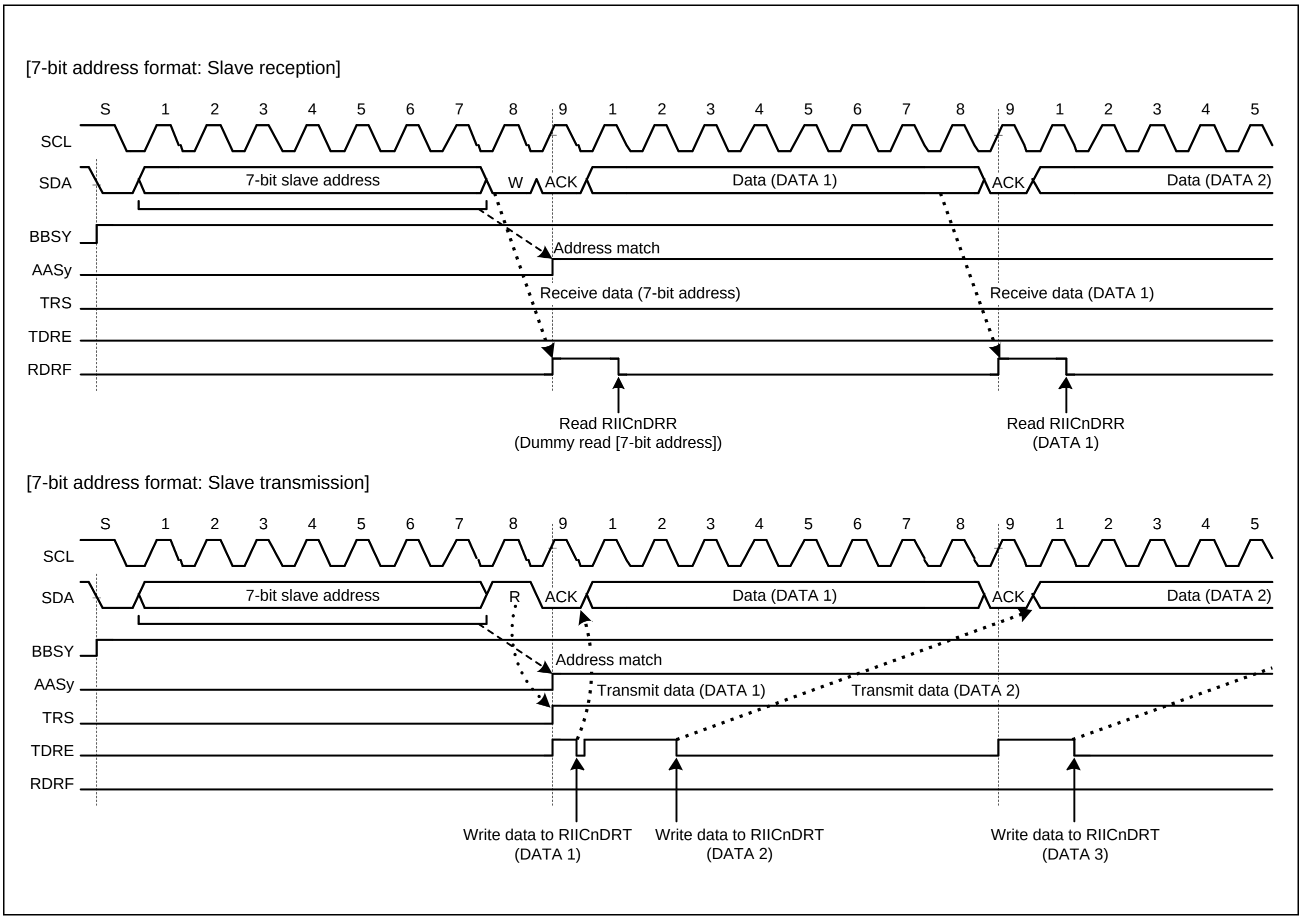


Figure 23.24 AASy Flag Set Timing with 7-Bit Address Format Selected

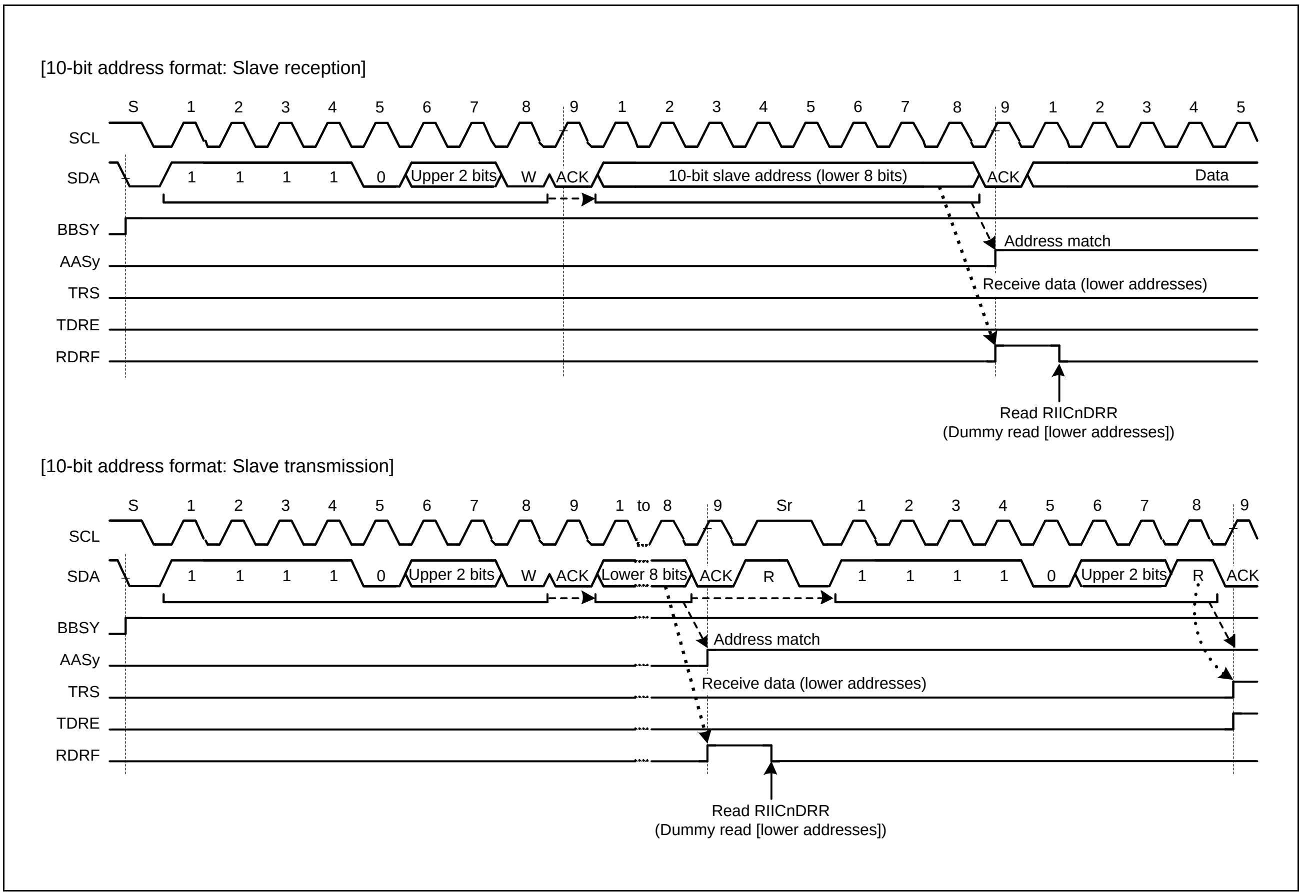


Figure 23.25 AASy Flag Set Timing with 10-Bit Address Format Selected

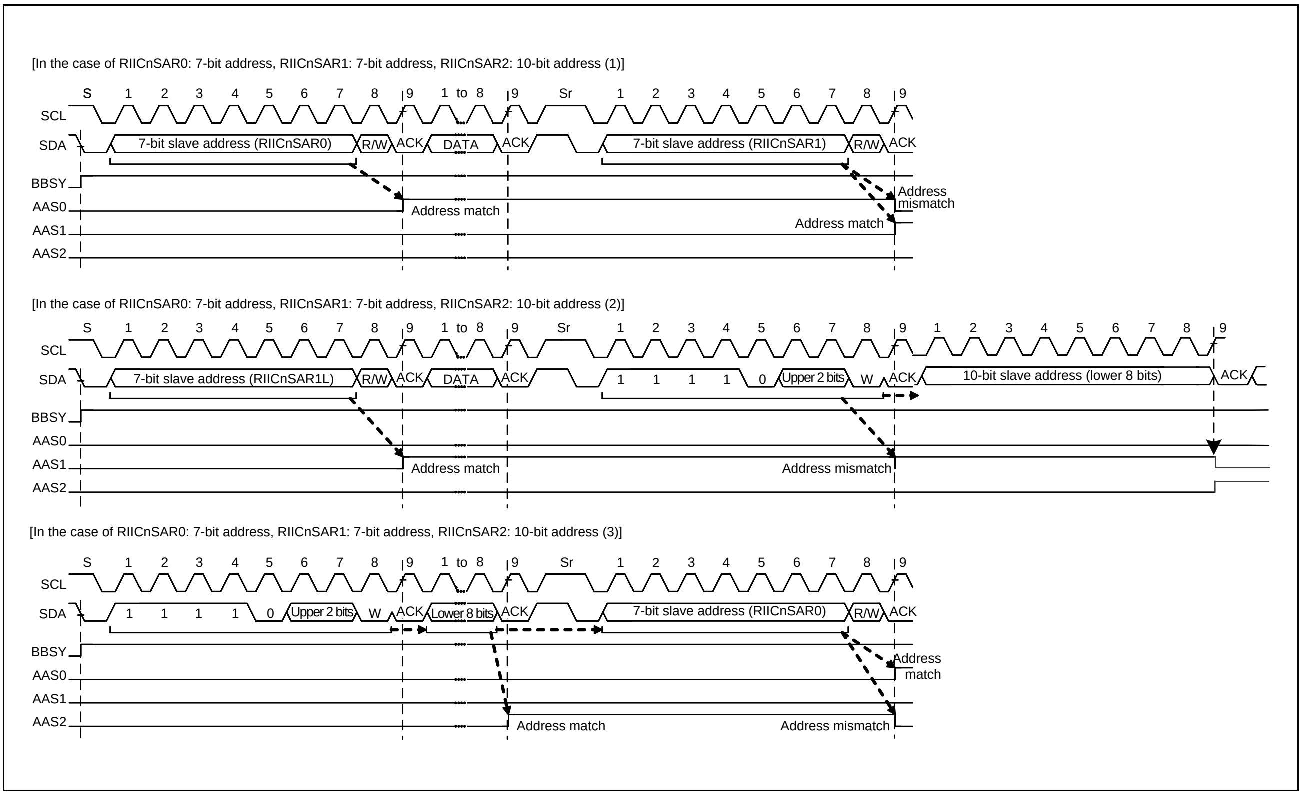


Figure 23.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

23.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address ($0000\ 000_B$ + 0 [W]). This is enabled by setting the RIICnSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is $0000\ 000_B$ + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive complete interrupt (INTRIICnRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

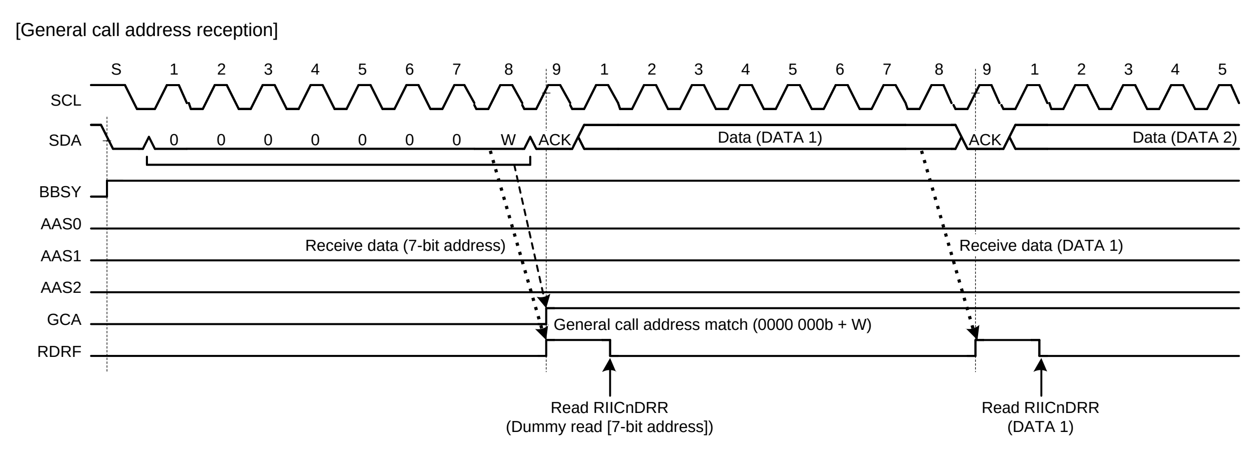


Figure 23.27 Timing of GCA Flag Setting during Reception of General Call Address

23.9.3 Device-ID Address Detection

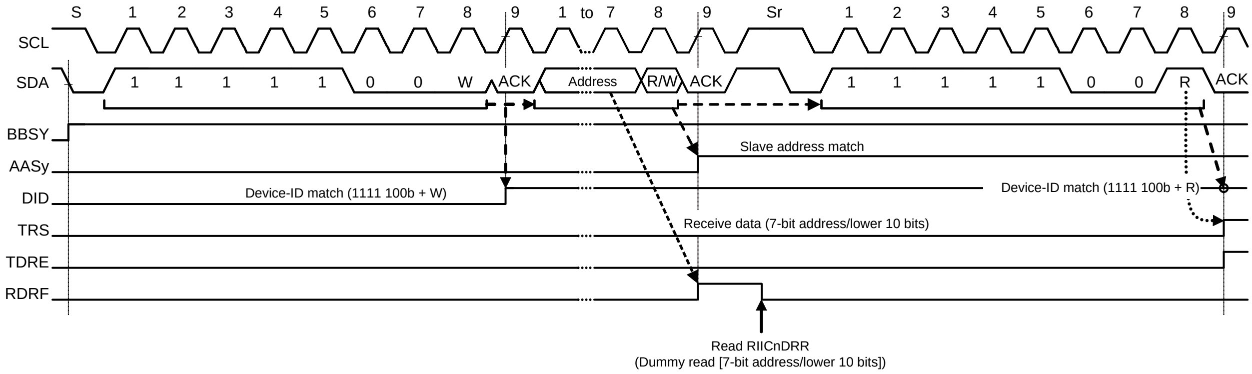
The RIIC module has a facility for detecting device-ID addresses conforming with the I^{2}C bus specification (Rev. 03). When the RIIC receives 1111 100_B as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100_B) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

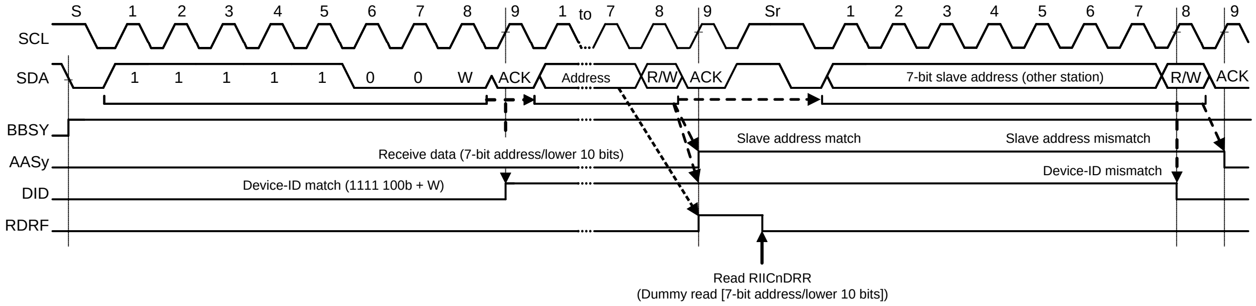
In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC’s own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC’s own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100_B) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC’s slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I^{2}C Bus Standard from NXP Semiconductors.

[Device-ID reception]



[When address received after a restart condition is detected does not match the Device-ID]



[When address before the Device-ID + R does not match the slave address]

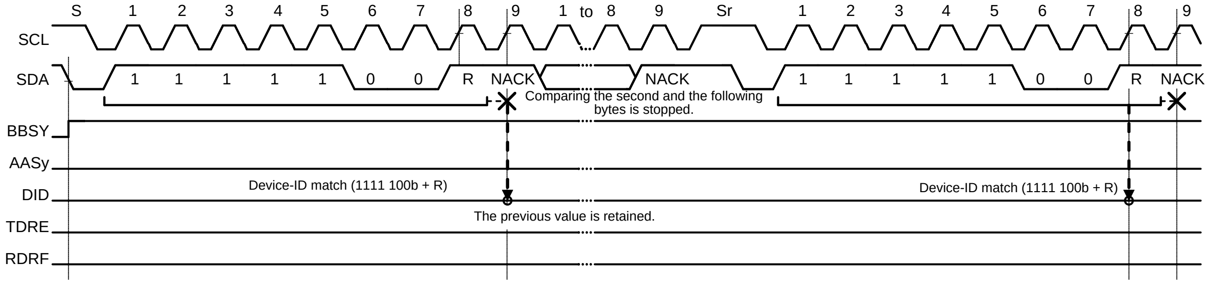


Figure 23.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

23.10 Automatic Low-Hold Function for SCL

23.10.1 Function to Prevent Wrong Transmission of Transmit Data

To prevent the unintended transmission of erroneous data, this low-hold period is extended until data for transmission have been written. In addition, the RIIC holds the SCL line low over the period until a stop condition is issued and also over the period until the RIICnDRR register is dummy read.

<Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle until a stop condition is issued

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle and the RIICnDRR register is dummy read

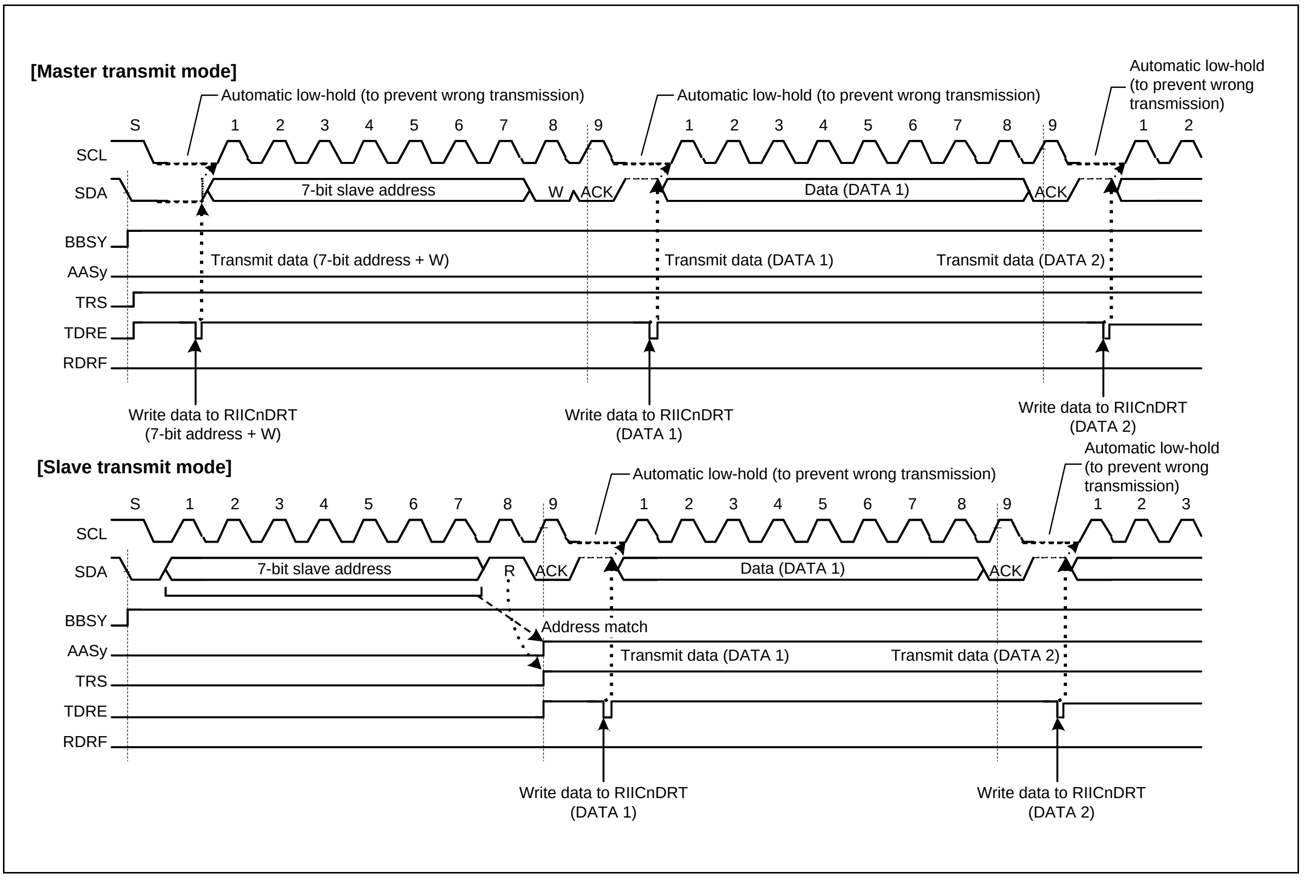


Figure 23.29 Automatic Low-Hold Operation in Transmit Mode

23.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

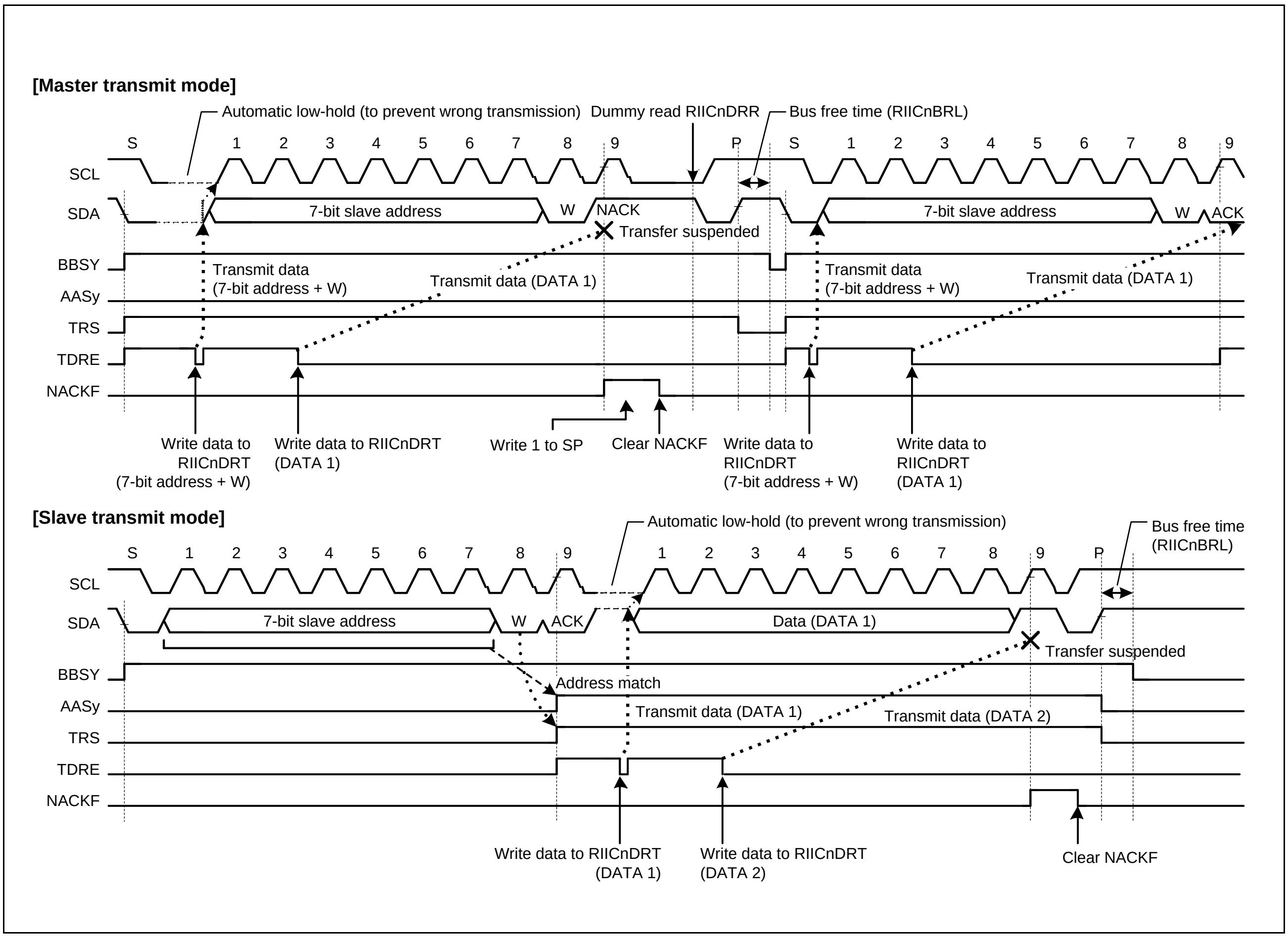


Figure 23.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

23.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive complete (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.
Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.
When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive complete) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

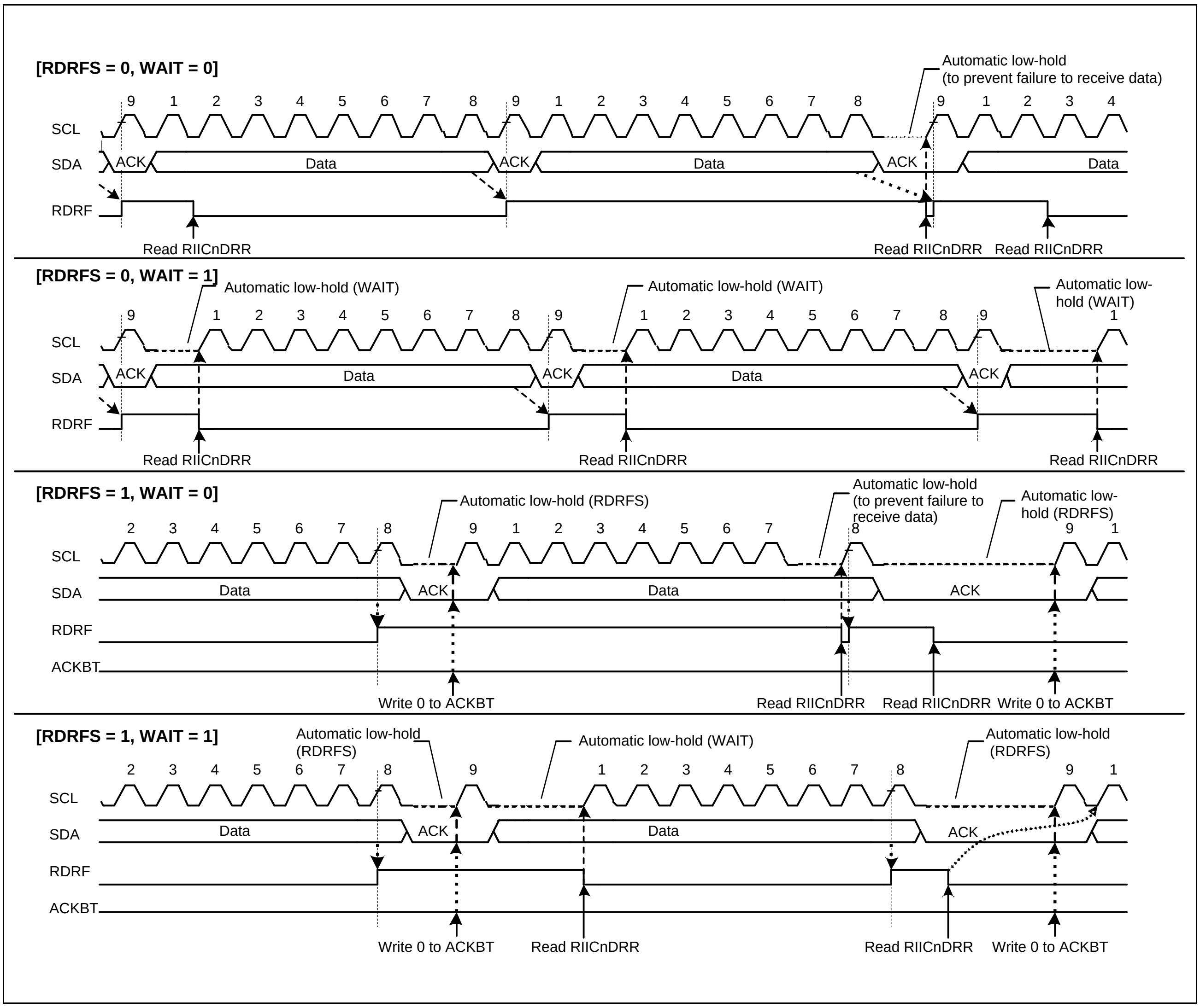


Figure 23.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

23.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^{2}C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

23.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if the RIICnCR2.ST bit is set to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC considers itself to have lost in arbitration, so priority is given to transfer by the other master device and no start condition is generated.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line), the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11_B)

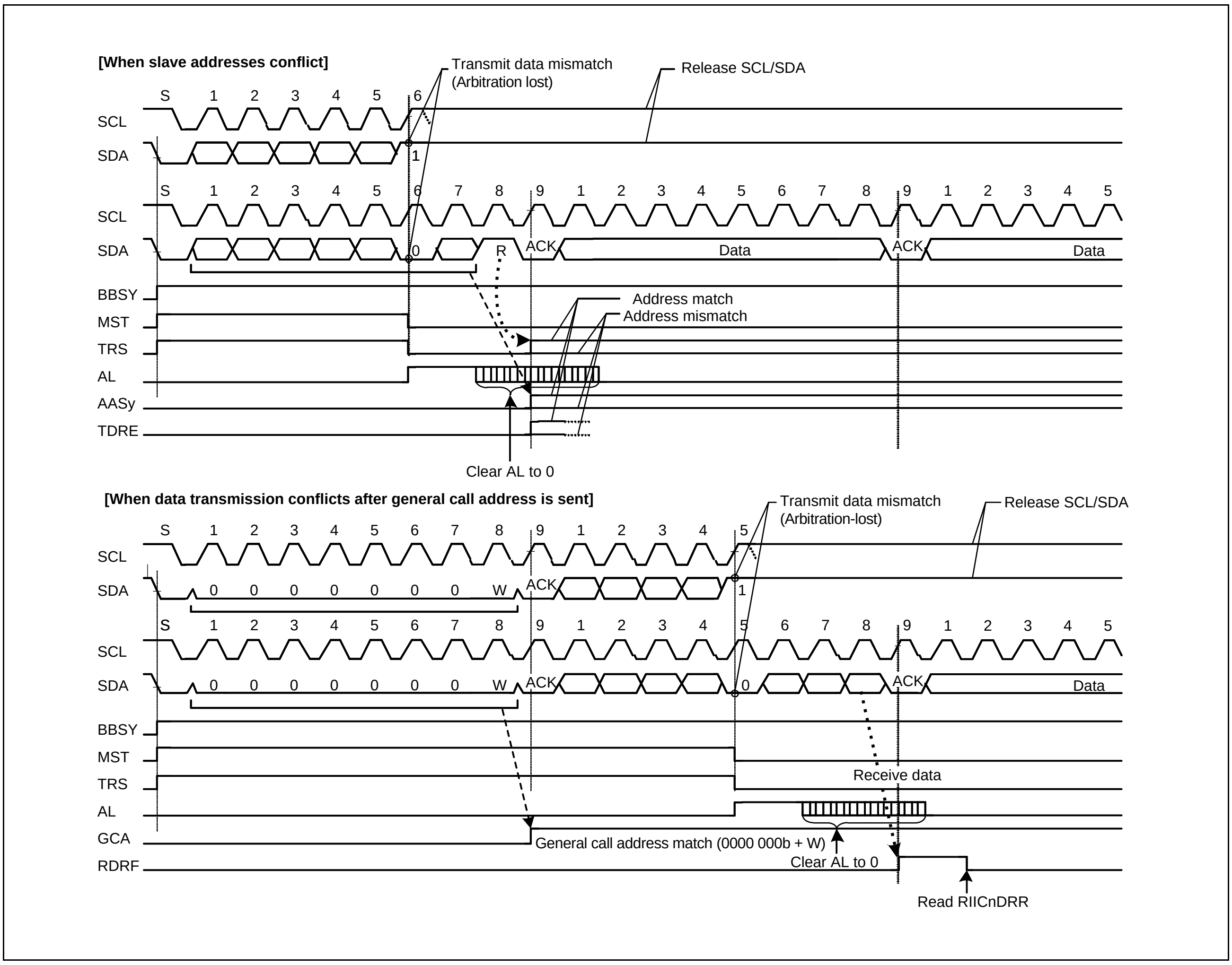


Figure 23.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

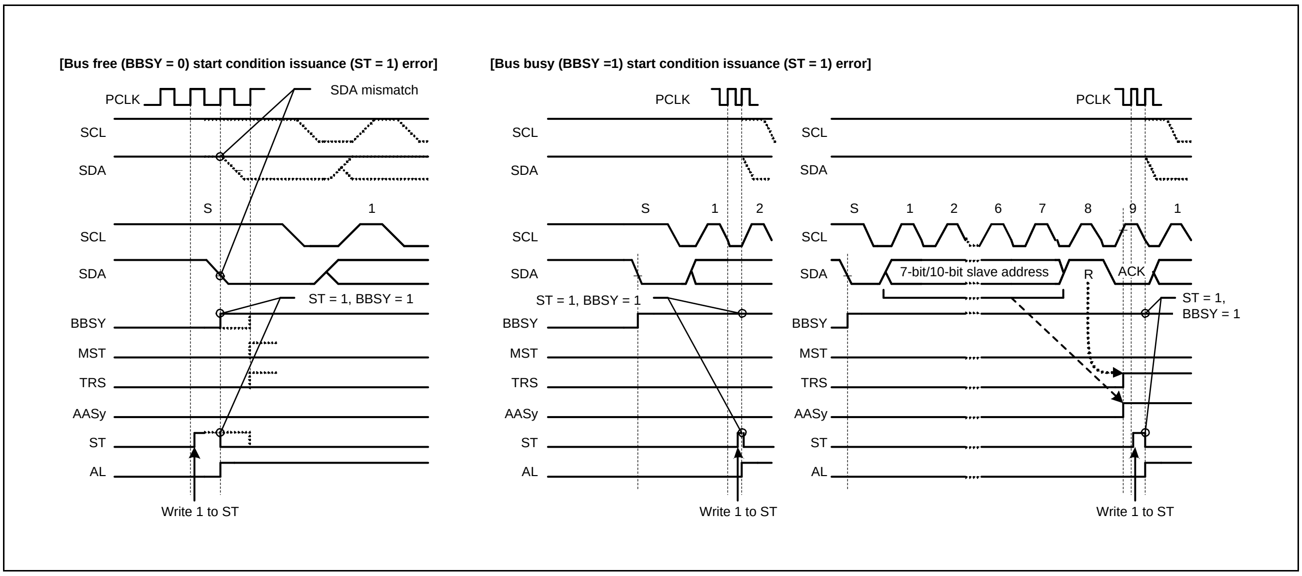


Figure 23.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

The TRS bit is not cleared if 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode.

23.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line) during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 23.34, Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)** shows an example of arbitration-lost detection during transmission of NACK.

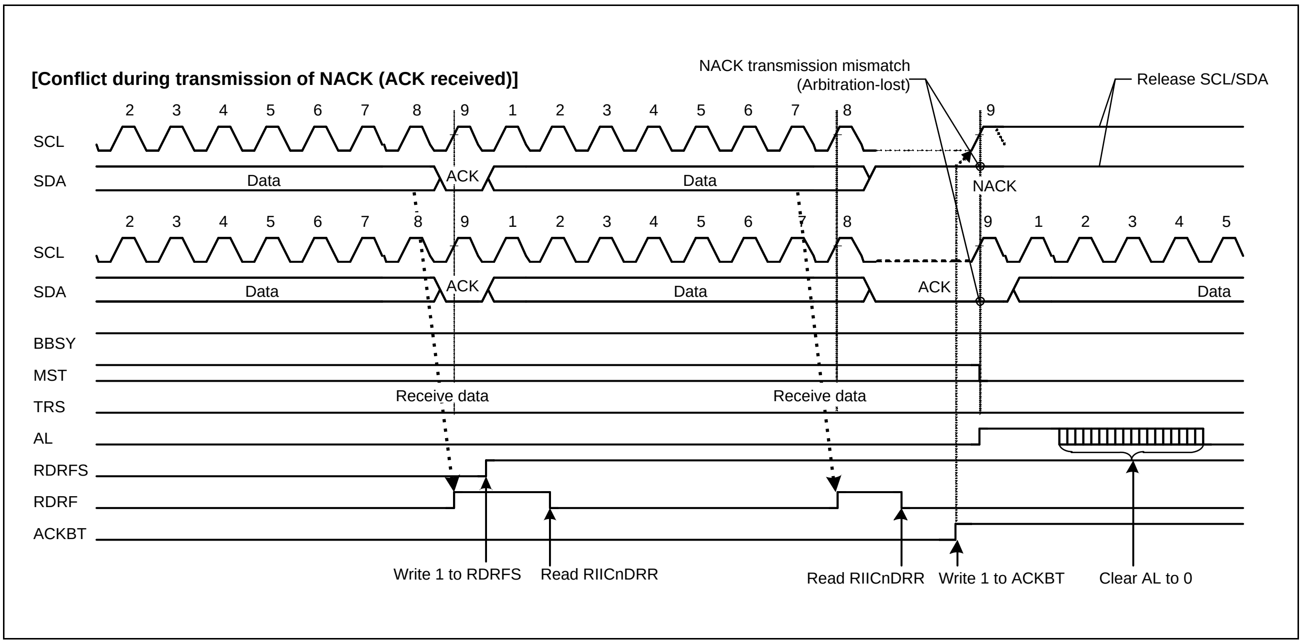


Figure 23.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

23.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the highimpedance state, and the low level is detected on the SDA line) in slave transmit mode.

When it loses slave arbitration, the RIIC enters slave receive mode.

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01_B)

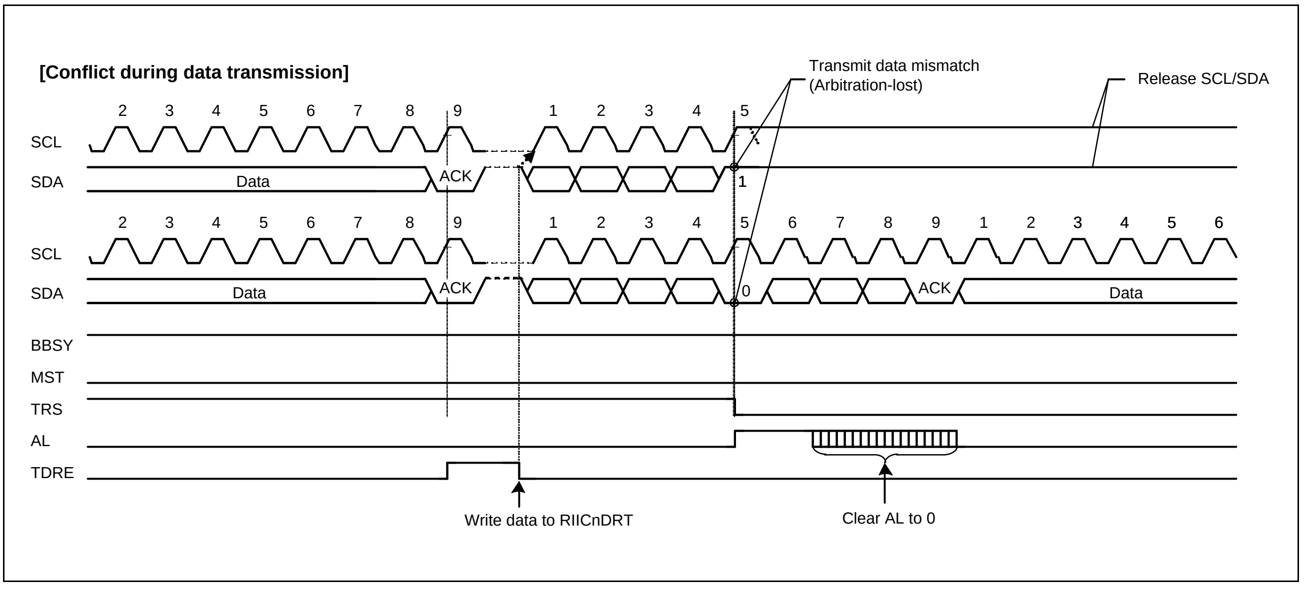


Figure 23.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

23.12 Start Condition/Restart Condition/Stop Condition Issuing Function

23.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

23.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made even during communication and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

(To detect the issuance of a restart condition, clear the RIICnSR2.START flag before a restart condition is issued.)

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

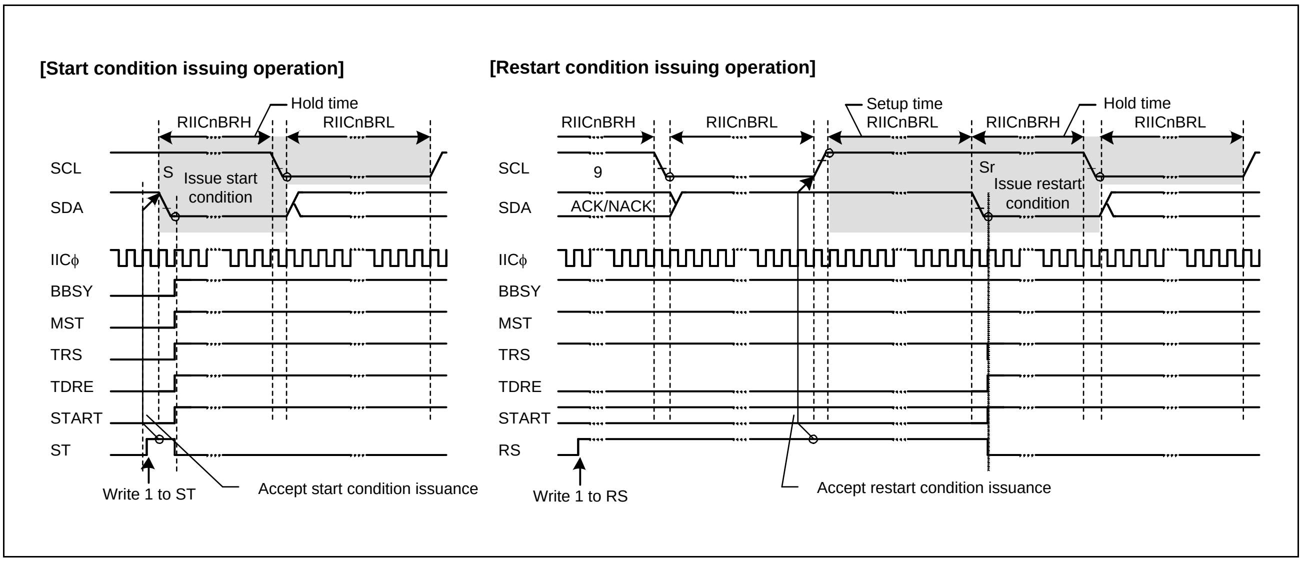


Figure 23.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

23.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

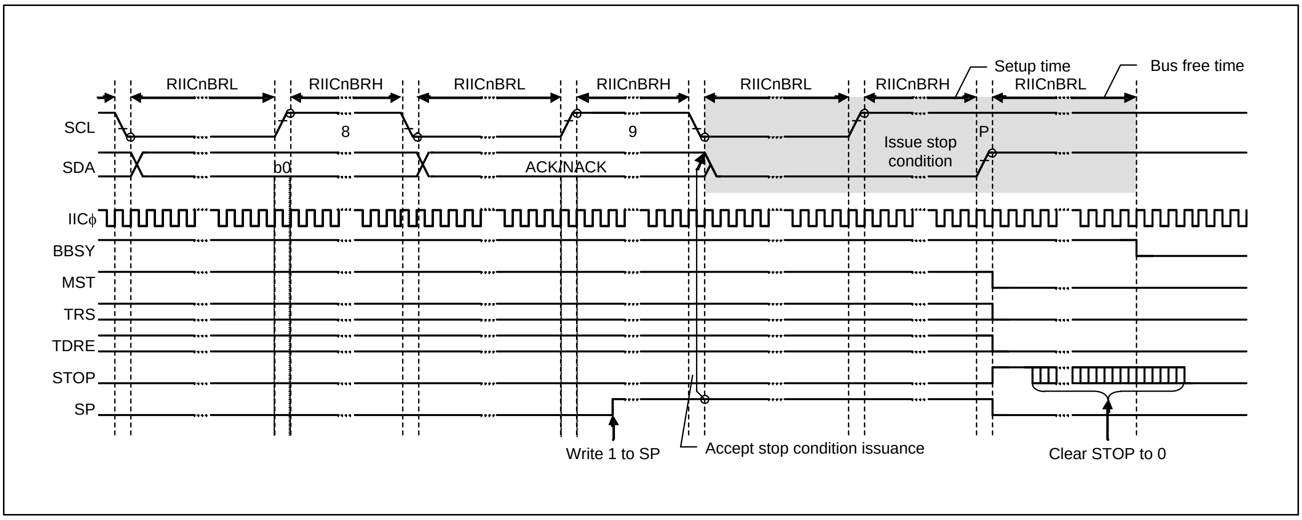


Figure 23.37 Stop Condition Issue Timing (SP Bit)

23.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I^{2}C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

23.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

The internal counter is cleared when one of the conditions is met.

(1) When RIICnMR2.TMOH=0, and RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising

(2) When RIICnMR2.TMOH=1, and RIICnMR2.TMOL=0:
The internal counter is cleared by SCL falling

(3) When RIICnMR2.TMOH=RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising or falling

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY flag is 1) in master mode (RIICnCR2.MST bit is 1).
- The RIIC’s own slave address matches (RIICnSR1 register is not 00_H) and the bus is busy (RIICnCR2.BBSY flag is 1) in slave mode (RIICnCR2.MST bit is 0).
- The bus is free (RIICnCR2.BBSY flag is 0) while generation of a START condition is requested (RIICnCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

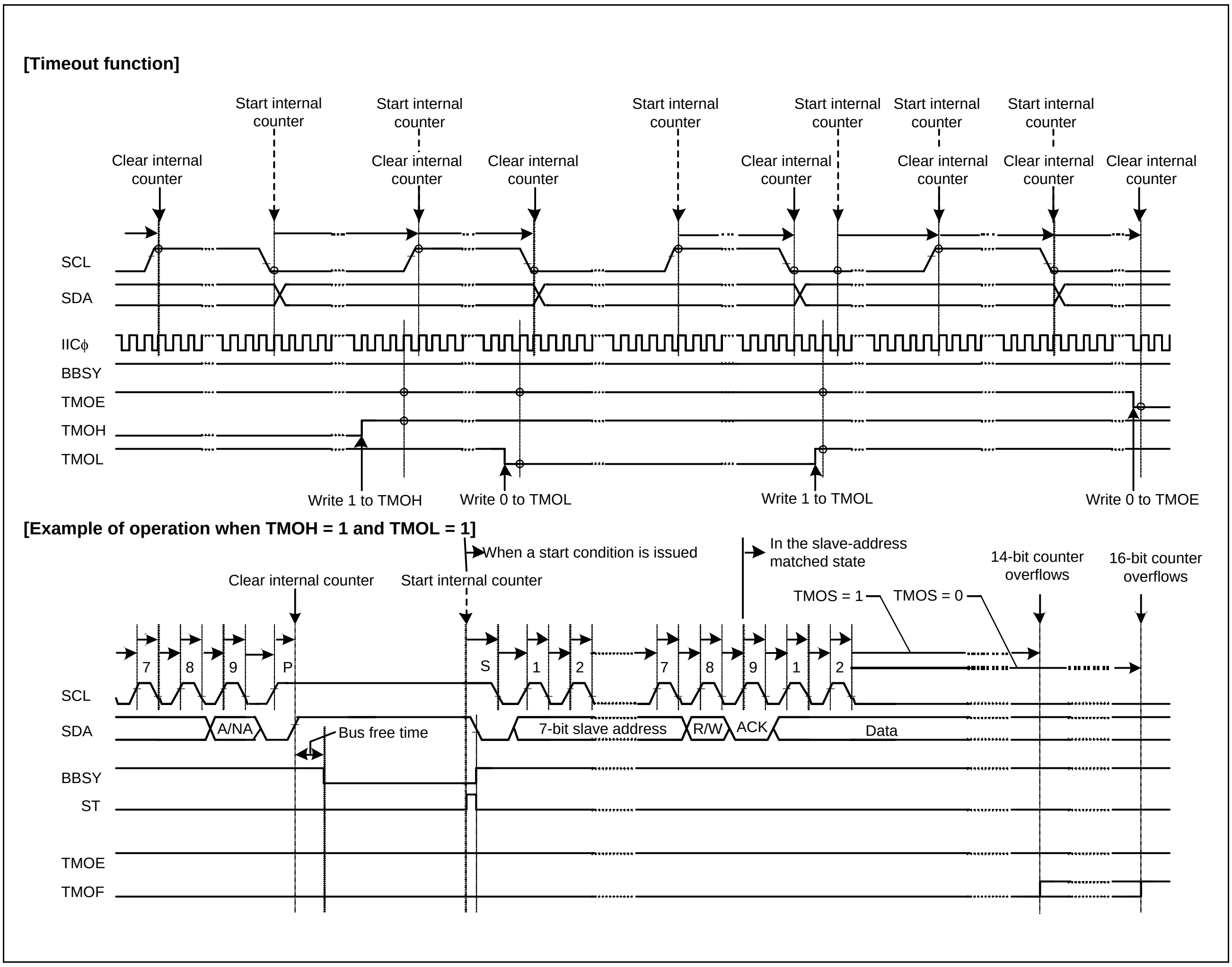


Figure 23.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

23.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the RIICnFER.MALE bit (master arbitration-lost detection disabled) cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the RIICnCR1.SDAO bit does not match the state of the SDA line, so take care on this point.

[Additional output conditions for the SCL clock]

- In master mode and when the bus is free
- In master mode and the SCL line is not held low (the bus is busy)

Figure 23.39, Extra SCL Clock Cycle Output Function (CLO Bit) shows the operation timing of the extra SCL clock cycle output function (CLO bit).

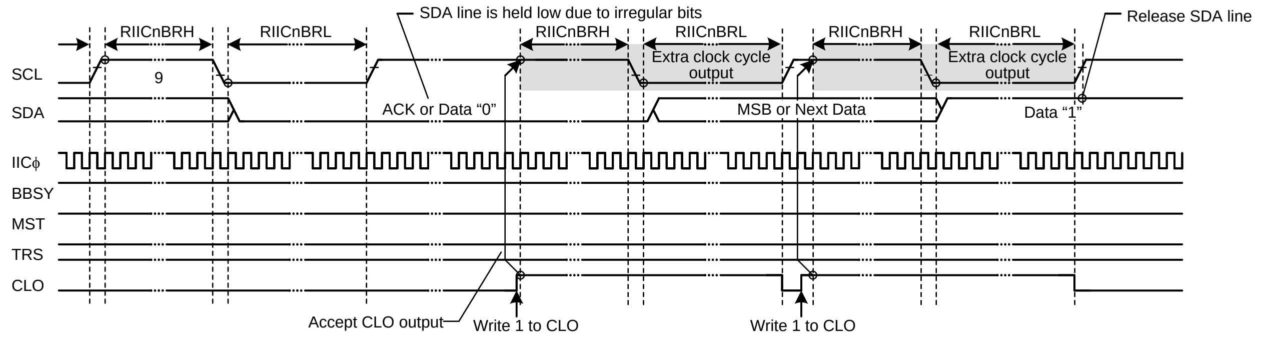


Figure 23.39 Extra SCL Clock Cycle Output Function (CLO Bit)

23.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01_B).

For a detailed description of the RIIC and internal resets, see **Section 23.14, Reset Function of RIIC**.

23.14 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES.
Table 23.28, RIIC Reset Functions lists the scope of each reset and reset conditions.

Table 23.28 RIIC Reset Functions

UM		ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	Initialized	0	1	Retained	Retained
	IICRST	Initialized	1	1	Retained	Retained
	CLO	Initialized	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized	Initialized*[1]	Operation	Retained
	MST	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR1	MTWP	Initialized	Initialized	Retained	Retained	Retained
	CKS[2:0]	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Initialized	Retained	Initialized

Table 23.28 RIIC Reset Functions

UM		ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnSR2	TDRE	Initialized	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Initialized	Retained	Retained
RIICnSAR0, 1, 2		Initialized	Initialized	Retained	Retained	Retained
RIICnBRH, RIICnBRL		Initialized	Initialized	Retained	Retained	Retained
RIICnDRT		Initialized	Initialized	Retained	Retained	Retained
RIICnDRR		Initialized	Initialized	Retained	Retained	Retained
RIICnDRS		Initialized	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared to 0.

Section 24 CANFD Interface (RS-CANFD)

This section contains a generic description of the CAN Interface (RS-CANFD).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RS-CANFD.

24.1 Features of RH850/F1KH, RH850/F1KM RS-CANFD

24.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CANFD units and channels.

Table 24.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	2
Name	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0, 1)

Table 24.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0)

Table 24.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0)	RCFDCn (n = 0)

Table 24.4 Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name RCFDCn	Unit Channel Total Number z	Channel Name CANm	RH850/F1KH-D8 176 Pins (8ch)	RH850/F1KH-D8 233 Pins (8ch)	RH850/F1KH-D8 324 Pins (12ch)
RCFDC0	0	CAN0	✓	✓	✓
	1	CAN1	✓	✓	✓
	2	CAN2	✓	✓	✓
	3	CAN3	✓	✓	✓
	4	CAN4	✓	✓	✓
	5	CAN5	✓	✓	✓
	6	CAN6	✓	✓	✓
	7	CAN7	✓	✓	✓
RCFDC1	8	CAN0	—	—	✓
	9	CAN1	—	—	✓
	10	CAN2	—	—	✓
	11	CAN3	—	—	✓

Table 24.5 Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name RCFDCn	Unit Channel Total Number z	Channel Name CANm	RH850/F1KM-S4 100 Pins (8ch)	RH850/F1KM-S4 144 Pins (8ch)	RH850/F1KM-S4 176 Pins (8ch)	RH850/F1KM-S4 233 Pins (8ch)	RH850/F1KM-S4 272 Pins (8ch)
RCFDC0	0	CAN0	✓	✓	✓	✓	✓
	1	CAN1	✓	✓	✓	✓	✓
	2	CAN2	✓	✓	✓	✓	✓
	3	CAN3	✓	✓	✓	✓	✓
	4	CAN4	✓	✓	✓	✓	✓
	5	CAN5	✓	✓	✓	✓	✓
	6	CAN6	✓	✓	✓	✓	✓
	7	CAN7	✓	✓	✓	✓	✓

Table 24.6 Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name RCFDCn	Unit Channel Total Number z	Channel Name CANm	RH850/F1KM-S1 48 Pins (1ch)	RH850/F1KM-S1 64 Pins (3ch)	RH850/F1KM-S1 80 Pins (3ch)	RH850/F1KM-S1 100 Pins (6ch)
RCFDC0	0	CAN0	✓	✓	✓	✓
	1	CAN1	—	✓	✓	✓
	2	CAN2	—	✓	✓	✓
	3	CAN3	—	—	—	✓
	4	CAN4	—	—	—	✓
	5	CAN5	—	—	—	✓

Table 24.7 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index “n”; for example, RCFDCnCFDGCTR is the global control register of the RCFDCn unit (n = 0, 1).
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “m”; for example, RCFDCnCFDCmSTS is the channel m status register (m = 0 to 7 with n = 0, or m = 0 to 3 with n = 1).
z	Throughout this section, the total number of channels of RS-CANFD units is generically indicated by the index “z” (z = 0 to 11).
j	The individual registers associated with receive rule table are generically indicated by the index “j”; for example, RCFDCnCFDGAFLIDj (j = 0 to 15 with both value of index “n” (n = 0, 1)) is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k”; for example, RCFDCnCFDCFCCk (k = 0 to [channel m × 3 + 2] with both value of index “n” (n = 0, 1)) is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are generically identified by the index “x”; for example, RCFDCnCFDRFSTSx (x = 0 to 7 with both value of index “n” (n = 0, 1)) is the receive FIFO buffer status register.
d	Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by “d”; for example, the transmit/receive FIFO buffer data field register is described as RCFDCnCFDCFDFd_k (d = 0 to 15).
q	The individual receive buffers are generically indicated by the index “q”; for example, RCFDCnCFDRMIDq (q = 0 to [channel m × 16 + 15] with both value of index “n” (n = 0, 1)) is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index “p”; for example, RCFDCnCFDTMCp (p = 0 to [channel m × 32 + 31] with both value of index “n” (n = 0, 1)) is the transmit buffer control register.
b	Data field registers of receive buffers and transmit buffers are identified by “b”; for example, the receive buffer data field register is described as RCFDCnCFDRMDFb_q (b = 0 to 15 with both value of index “n” (n = 0, 1)).
r	The individual RAM tests for CAN are generically indicated by the index “r”; for example, RCFDCnCFDRPGACCr (r = 0 to 63 with both value of index “n” (n = 0, 1)) is the RAM test page access register.
y	The registers not covered above are indicated by the index “y”; for example, RCFDCnCFDRMNDy (y = 0 to 3 with n = 0 or (y= 0, 1)) with n = 1 is a receive buffer new data register.

Note: The functions and descriptions of registers in this section are for the RS-CANFDn that has 8 channels (m = 0 to 7) for unit 0 (n = 0) and 4 channels (m = 0 to 3) for unit 1 (n = 1). When referring to information with indices, regard the index values as the ones corresponding to your target product. Also, note that, if the value of an index exceeds the range described in this section due to your target product, write the value after reset when writing to bits outside the index range.

Table 24.8 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index “n”; for example, RCFDCnCFDGCTR is the global control register of the RCFDCn unit (n = 0).
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “m”; for example, RCFDCnCFDCmSTS is the channel m status register (m = 0 to 7).
z	Throughout this section, the total number of channels of RS-CANFD units is generically indicated by the index “z” (z = 0 to 7).
j	The individual registers associated with receive rule table are generically indicated by the index “j”; for example, RCFDCnCFDGAFLIDj (j = 0 to 15) is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k”; for example, RCFDCnCFDCFCCk (k = 0 to [channel m × 3 + 2]) is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are generically identified by the index “x”; for example, RCFDCnCFDRFSTSx (x = 0 to 7) is the receive FIFO buffer status register.
d	Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by “d”; for example, the transmit/receive FIFO buffer data field register is described as RCFDCnCFDCFDFd_k (d = 0 to 15).
q	The individual receive buffers are generically indicated by the index “q”; for example, RCFDCnCFDRMIDq (q = 0 to [channel m × 16 + 15]) is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index “p”; for example, RCFDCnCFDTMCp (p = 0 to [channel m × 32 + 31]) is the transmit buffer control register.
b	Data field registers of receive buffers and transmit buffers are identified by “b”; for example, the receive buffer data field register is described as RCFDCnCFDRMDFb_q (b = 0 to 15).
r	The individual RAM tests for CAN are generically indicated by the index “r”; for example, RCFDCnCFDRPGACCr (r = 0 to 63) is the RAM test page access register.
y	The registers not covered above are indicated by the index “y”; for example, RCFDCnCFDRMNDy (y = 0 to 3 with n = 0, or y= 0, 1 with n = 1) is a receive buffer new data register.

Note: The functions and descriptions of registers in this section are for the RS-CANFD that has 8 channels (m = 0 to 7). When referring to information with indices, regard the index values as the ones corresponding to your target product. Also, note that, if the value of an index exceeds the range described in this section due to your target product, write the value after reset when writing to bits outside the index range.

Table 24.9 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index “n”; for example, RCFDCnCFDGCTR is the global control register of the RCFDCn unit (n = 0).
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “m”; for example, RCFDCnCFDCmSTS is the channel m status register (m = 0 to 5).
z	Throughout this section, the total number of channels of RS-CANFD units is generically indicated by the index “z” (z = 0 to 5).
j	The individual registers associated with receive rule table are generically indicated by the index “j”; for example, RCFDCnCFDGAFLIDj (j = 0 to 15) is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k”; for example, RCFDCnCFDCFCCk (k = 0 to [channel m × 3 + 2]) is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are generically identified by the index “x”; for example, RCFDCnCFDRFSTSx (x = 0 to 7)*[1] is the receive FIFO buffer status register.
d	Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by “d”; for example, the transmit/receive FIFO buffer data field register is described as RCFDCnCFDCFDFd_k (d = 0 to 15).
q	The individual receive buffers are generically indicated by the index “q”; for example, RCFDCnCFDRMIDq (q = 0 to [channel m × 16 + 15]) is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index “p”; for example, RCFDCnCFDTMCp (p = 0 to [channel m × 32 + 31]) is the transmit buffer control register.
b	Data field registers of receive buffers and transmit buffers are identified by “b”; for example, the receive buffer data field register is described as RCFDCnCFDRMDFb_q (b = 0 to 15).
r	The individual RAM tests for CAN are generically indicated by the index “r”; for example, RCFDCnCFDRPGACCr (r = 0 to 63) is the RAM test page access register.
y	The registers not covered above are indicated by the index “y”; for example, RCFDCnCFDRMNDy (y = 0 to 2 with n = 0, or y = 0, 1 with n = 1) is a receive buffer new data register.

Note: The functions and descriptions of registers in this section are for the RS-CANFD that has 6 channels (m = 0 to 5). When referring to information with indices, regard the index values as the ones corresponding to your target product. Also, note that, if the value of an index exceeds the range described in this section due to your target product, write the value after reset when writing to bits outside the index range.

Note 1. Only the index “x” of RCFDCnCFDCDTCT regisiter and RCFDCnCFDCDTSTS register for RH850/F1KM-S1 are x = 0 to 5.

The following table lists the values of indices for individual products.

Table 24.10 Indices for Individual Products (RH850/F1KH-D8)

	Indices for Individual Products		
Unit Name	176 Pins	233 Pins	324 Pins
RCFDC0	m = 0 to 7	m = 0 to 7	m = 0 to 7
	z = 0 to 7	z = 0 to 7	z = 0 to 7
	j = 0 to 15	j = 0 to 15	j = 0 to 15
	k = 0 to 23	k = 0 to 23	k = 0 to 23
	x = 0 to 7	x = 0 to 7	x = 0 to 7
	d = 0 to 15	d = 0 to 15	d = 0 to 15
	q = 0 to 127	q = 0 to 127	q = 0 to 127
	p = 0 to 255	p = 0 to 255	p = 0 to 255
	b = 0 to 15	b = 0 to 15	b = 0 to 15
	r = 0 to 63	r = 0 to 63	r = 0 to 63
	y = 0 to 3	y = 0 to 3	y = 0 to 3
RCFDC1	—	—	m = 0 to 3
	—	—	z = 8 to 11
	—	—	j = 0 to 15
	—	—	k = 0 to 11
	—	—	x = 0 to 7
	—	—	d = 0 to 15
	—	—	q = 0 to 63
	—	—	p = 0 to 127
	—	—	b = 0 to 15
	—	—	r = 0 to 63
	—	—	y = 0, 1

Table 24.11 Indices for Individual Products (RH850/F1KM-S4)

Unit Name	Indices for Individual Products				
	100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
RCFDC0	m = 0 to 7	m = 0 to 7	m = 0 to 7	m = 0 to 7	m = 0 to 7
	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7	z = 0 to 7
	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15
	k = 0 to 23	k = 0 to 23	k = 0 to 23	k = 0 to 23	k = 0 to 23
	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7
	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15
	q = 0 to 127	q = 0 to 127	q = 0 to 127	q = 0 to 127	q = 0 to 127
	p = 0 to 255	p = 0 to 255	p = 0 to 255	p = 0 to 255	p = 0 to 255
	b = 0 to 15	b = 0 to 15	b = 0 to 15	b = 0 to 15	b = 0 to 15
	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63
	y = 0 to 3	y = 0 to 3	y = 0 to 3	y = 0 to 3	y = 0 to 3

Table 24.12 Indices for Individual Products (RH850/F1KM-S1)

Unit Name	Indices for Individual Products			
	48 Pins	64 Pins	80 Pins	100 Pins
RCFDC0	m = 0	m = 0 to 2	m = 0 to 2	m = 0 to 5
	z = 0	z = 0 to 2	z = 0 to 2	z = 0 to 5
	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15
	k = 0 to 2	k = 0 to 8	k = 0 to 8	k = 0 to 17
	x = 0 to 7*[1]	x = 0 to 7*[1]	x = 0 to 7*[1]	x = 0 to 7*[1]
	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15
	q = 0 to 15	q = 0 to 47	q = 0 to 47	q = 0 to 95
	p = 0 to 31	p = 0 to 95	p = 0 to 95	p = 0 to 191
	b = 0 to 15	b = 0 to 15	b = 0 to 15	b = 0 to 15
	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63
	y = 0	y = 0, 1	y = 0, 1	y = 0 to 2

Note 1. Only the index “x” of RCFDCnCFDCDTCT regisiter and RCFDCnCFDCDTSTS register for RH850/F1KM-S1 are x = 0 to 5.

24.1.2 Register Base Addresses

RCFDCn base addresses are listed in the following table.

RCFDCn register addresses are given as offsets from the base addresses.

Table 24.13 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<RCFDC0_base>	FFD0 0000_H
<RCFDC1_base>	FFD2 0000_H

Table 24.14 Register Base Address (RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<RCFDC0_base>	FFD0 0000_H

24.1.3 Clock Supply

The RCFDCn clock supply is shown in the following table.

Table 24.15 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RCFDCn	clk_xincan	CKSCLK_ICANOSC	Communication clock from OSC clock
	clkc	CKSCLK_IPERI2	Communication clock
	pclk	CKSCLK_ICAN	Module clock
	Register access clock	CPUCLK_M, CKSCLK_ICAN	Bus clock

The operating frequency of the RCFDCn depends on the transfer rate and the number of channels in use. **Table 24.16, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM** shows the range of the frequency.

Table 24.16 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM

Condition			Range of Operating Frequency		
Nominal Bit Rate	Data Bit Rate	No. of Channels in Use (max.)	pclk	clk_xincan*[1]	clkc*[1]
500 kbps	5 Mbps	8 ch	pclk = 80 MHz	Not applicable	clkc = 40 MHz
1 Mbps	2 Mbps	8 ch	pclk = 80 MHz	Not applicable	clkc = 40 MHz
500 kbps	2 Mbps	2 ch	pclk = 24 MHz	clk_xincan = 12 MHz	Not applicable
500 kbps	2 Mbps	8 ch	pclk = 80 MHz	Not applicable	clkc = 40 MHz

Note 1. Setting the DCS bit in the RCFDCnCFDGCFG register enables to select either clk_xincan or clkc. Set clocks less than or equal to pclk/2.

CAUTION

When the RS-CANFD module is used in stop mode, set the MainOSC as the clock source of the RS-CANFD module. For details about how to set the clock source, see **Section 12AB.4.3.10, RS-CANFD Clock Domains C_ISO_CAN and C_ISO_CANOSC** and **Section 12C.4.3.9, RS-CANFD Clock Domains C_ISO_CAN and C_ISO_CANOSC**.

24.1.4 Interrupt Requests

RCFDCn interrupt requests are listed in the following table.

Table 24.17 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal		Description	Interrupt Number	DMA Trigger Number
RCFDC0				
Global	INTRCANGERR0	CAN global error interrupt	22	—
	INTRCANGRECC0	CAN receive FIFO interrupt	23	—
	RSCANFDRF0	Reception FIFO access message buffers0	—	60
	RSCANFDRF1	Reception FIFO access message buffers1	—	61
	RSCANFDRF2	Reception FIFO access message buffers2	—	62
	RSCANFDRF3	Reception FIFO access message buffers3	—	63
	RSCANFDRF4	Reception FIFO access message buffers4	—	68
	RSCANFDRF5	Reception FIFO access message buffers5	—	69
	RSCANFDRF6	Reception FIFO access message buffers6	—	82
	RSCANFDRF7	Reception FIFO access message buffers7	—	83
CAN0	INTRCAN0ERR	CAN0 error interrupt	24	—
	INTRCAN0REC	CAN0 transmit/receive FIFO receive completion interrupt	25	—
	INTRCAN0TRX	CAN0 transmit interrupt	26	—
	RSCANFDCF0	CAN0 common FIFO access message buffers	—	23
CAN1	INTRCAN1ERR	CAN1 error interrupt	113	—
	INTRCAN1REC	CAN1 transmit/receive FIFO receive completion interrupt	114	—
	INTRCAN1TRX	CAN1 transmit interrupt	115	—
	RSCANFDCF1	CAN1 common FIFO access message buffers	—	24
CAN2	INTRCAN2ERR	CAN2 error interrupt	217	—
	INTRCAN2REC	CAN2 transmit/receive FIFO receive completion interrupt	218	—
	INTRCAN2TRX	CAN2 transmit interrupt	219	—
	RSCANFDCF2	CAN2 common FIFO access message buffers	—	26
CAN3	INTRCAN3ERR	CAN3 error interrupt	220	—
	INTRCAN3REC	CAN3 transmit/receive FIFO receive completion interrupt	221	—
	INTRCAN3TRX	CAN3 transmit interrupt	222	—
	RSCANFDCF3	CAN3 common FIFO access message buffers	—	27
CAN4	INTRCAN4ERR	CAN4 error interrupt	272	—
	INTRCAN4REC	CAN4 transmit/receive FIFO receive completion interrupt	273	—
	INTRCAN4TRX	CAN4 transmit interrupt	274	—
	RSCANFDCF4	CAN4 common FIFO access message buffers	—	48
CAN5	INTRCAN5ERR	CAN5 error interrupt	287	—
	INTRCAN5REC	CAN5 transmit/receive FIFO receive completion interrupt	288	—
	INTRCAN5TRX	CAN5 transmit interrupt	289	—
	RSCANFDCF5	CAN5 common FIFO access message buffers	—	49
CAN6	INTRCAN6ERR	CAN6 error interrupt	321	—
	INTRCAN6REC	CAN6 transmit/receive FIFO receive completion interrupt	322	—
	INTRCAN6TRX	CAN6 transmit interrupt	323	—
	RSCANFDCF6	CAN6 common FIFO access message buffers	—	64

Table 24.17 Interrupt Requests (RH850/F1KH-D8)

Unit	Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CAN7	INTRCAN7ERR	CAN7 error interrupt	332	—
	INTRCAN7REC	CAN7 transmit/receive FIFO receive completion interrupt	333	—
	INTRCAN7TRX	CAN7 transmit interrupt	334	—
	RSCANFDCF7	CAN7 common FIFO access message buffers	—	65
RCFDC1				
Global	INTRCANGERR1	CAN global error interrupt	319	—
	INTRCANGRECC1	CAN receive FIFO interrupt	320	—
	RSCANFDRF8	Reception FIFO access message buffers8	—	96
	RSCANFDRF9	Reception FIFO access message buffers9	—	97
	RSCANFDRF10	Reception FIFO access message buffers10	—	98
	RSCANFDRF11	Reception FIFO access message buffers11	—	99
	RSCANFDRF12	Reception FIFO access message buffers12	—	15
	RSCANFDRF13	Reception FIFO access message buffers13	—	25
	RSCANFDRF14	Reception FIFO access message buffers14	—	84
	RSCANFDRF15	Reception FIFO access message buffers15	—	57
CAN0	INTRCAN8ERR	CAN0 error interrupt	244	—
	INTRCAN8REC	CAN0 transmit/receive FIFO receive completion interrupt	245	—
	INTRCAN8TRX	CAN0 transmit interrupt	246	—
	RSCANFDCF8	CAN0 common FIFO access message buffers	—	92
CAN1	INTRCAN9ERR	CAN1 error interrupt	247	—
	INTRCAN9REC	CAN1 transmit/receive FIFO receive completion interrupt	248	—
	INTRCAN9TRX	CAN1 transmit interrupt	249	—
	RSCANFDCF9	CAN1 common FIFO access message buffers	—	93
CAN2	INTRCAN10ERR	CAN2 error interrupt	250	—
	INTRCAN10REC	CAN2 transmit/receive FIFO receive completion interrupt	251	—
	INTRCAN10TRX	CAN2 transmit interrupt	252	—
	RSCANFDCF10	CAN2 common FIFO access message buffers	—	94
CAN3	INTRCAN11ERR	CAN3 error interrupt	253	—
	INTRCAN11REC	CAN3 transmit/receive FIFO receive completion interrupt	254	—
	INTRCAN11TRX	CAN3 transmit interrupt	255	—
	RSCANFDCF11	CAN3 common FIFO access message buffers	—	95

Table 24.18 Interrupt Requests (RH850/F1KM-S4)

Unit	Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RCFDC0				
Global	INTRCANGERR0	CAN global error interrupt	22	—
	INTRCANGRECC0	CAN receive FIFO interrupt	23	—
	RSCANFDRF0	Reception FIFO access message buffers0	—	60
	RSCANFDRF1	Reception FIFO access message buffers1	—	61
	RSCANFDRF2	Reception FIFO access message buffers2	—	62
	RSCANFDRF3	Reception FIFO access message buffers3	—	63
	RSCANFDRF4	Reception FIFO access message buffers4	—	68
	RSCANFDRF5	Reception FIFO access message buffers5	—	69
	RSCANFDRF6	Reception FIFO access message buffers6	—	82
	RSCANFDRF7	Reception FIFO access message buffers7	—	83
CAN0	INTRCAN0ERR	CAN0 error interrupt	24	—
	INTRCAN0REC	CAN0 transmit/receive FIFO receive completion interrupt	25	—
	INTRCAN0TRX	CAN0 transmit interrupt	26	—
	RSCANFDCF0	CAN0 common FIFO access message buffers	—	23
CAN1	INTRCAN1ERR	CAN1 error interrupt	113	—
	INTRCAN1REC	CAN1 transmit/receive FIFO receive completion interrupt	114	—
	INTRCAN1TRX	CAN1 transmit interrupt	115	—
	RSCANFDCF1	CAN1 common FIFO access message buffers	—	24
CAN2	INTRCAN2ERR	CAN2 error interrupt	217	—
	INTRCAN2REC	CAN2 transmit/receive FIFO receive completion interrupt	218	—
	INTRCAN2TRX	CAN2 transmit interrupt	219	—
	RSCANFDCF2	CAN2 common FIFO access message buffers	—	26
CAN3	INTRCAN3ERR	CAN3 error interrupt	220	—
	INTRCAN3REC	CAN3 transmit/receive FIFO receive completion interrupt	221	—
	INTRCAN3TRX	CAN3 transmit interrupt	222	—
	RSCANFDCF3	CAN3 common FIFO access message buffers	—	27
CAN4	INTRCAN4ERR	CAN4 error interrupt	272	—
	INTRCAN4REC	CAN4 transmit/receive FIFO receive completion interrupt	273	—
	INTRCAN4TRX	CAN4 transmit interrupt	274	—
	RSCANFDCF4	CAN4 common FIFO access message buffers	—	48
CAN5	INTRCAN5ERR	CAN5 error interrupt	287	—
	INTRCAN5REC	CAN5 transmit/receive FIFO receive completion interrupt	288	—
	INTRCAN5TRX	CAN5 transmit interrupt	289	—
	RSCANFDCF5	CAN5 common FIFO access message buffers	—	49
CAN6	INTRCAN6ERR	CAN6 error interrupt	321	—
	INTRCAN6REC	CAN6 transmit/receive FIFO receive completion interrupt	322	—
	INTRCAN6TRX	CAN6 transmit interrupt	323	—
	RSCANFDCF6	CAN6 common FIFO access message buffers	—	64
CAN7	INTRCAN7ERR	CAN7 error interrupt	332	—
	INTRCAN7REC	CAN7 transmit/receive FIFO receive completion interrupt	333	—
	INTRCAN7TRX	CAN7 transmit interrupt	334	—
	RSCANFDCF7	CAN7 common FIFO access message buffers	—	65

Table 24.19 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal		Description	Interrupt Number	DMA Trigger Number
RCFDC0				
Global	INTRCANGERR0	CAN global error interrupt	22	—
	INTRCANGRECC0	CAN receive FIFO interrupt	23	—
	RSCANFDRF0	Reception FIFO access message buffers0	—	60
	RSCANFDRF1	Reception FIFO access message buffers1	—	61
	RSCANFDRF2	Reception FIFO access message buffers2	—	62
	RSCANFDRF3	Reception FIFO access message buffers3	—	63
	RSCANFDRF4	Reception FIFO access message buffers4	—	68
	RSCANFDRF5	Reception FIFO access message buffers5	—	69
CAN0	INTRCAN0ERR	CAN0 error interrupt	24	—
	INTRCAN0REC	CAN0 transmit/receive FIFO receive completion interrupt	25	—
	INTRCAN0TRX	CAN0 transmit interrupt	26	—
	RSCANFDCF0	CAN0 common FIFO access message buffers	—	23
CAN1	INTRCAN1ERR	CAN1 error interrupt	113	—
	INTRCAN1REC	CAN1 transmit/receive FIFO receive completion interrupt	114	—
	INTRCAN1TRX	CAN1 transmit interrupt	115	—
	RSCANFDCF1	CAN1 common FIFO access message buffers	—	24
CAN2	INTRCAN2ERR	CAN2 error interrupt	217	—
	INTRCAN2REC	CAN2 transmit/receive FIFO receive completion interrupt	218	—
	INTRCAN2TRX	CAN2 transmit interrupt	219	—
	RSCANFDCF2	CAN2 common FIFO access message buffers	—	26
CAN3	INTRCAN3ERR	CAN3 error interrupt	220	—
	INTRCAN3REC	CAN3 transmit/receive FIFO receive completion interrupt	221	—
	INTRCAN3TRX	CAN3 transmit interrupt	222	—
	RSCANFDCF3	CAN3 common FIFO access message buffers	—	27
CAN4	INTRCAN4ERR	CAN4 error interrupt	272	—
	INTRCAN4REC	CAN4 transmit/receive FIFO receive completion interrupt	273	—
	INTRCAN4TRX	CAN4 transmit interrupt	274	—
	RSCANFDCF4	CAN4 common FIFO access message buffers	—	48
CAN5	INTRCAN5ERR	CAN5 error interrupt	287	—
	INTRCAN5REC	CAN5 transmit/receive FIFO receive completion interrupt	288	—
	INTRCAN5TRX	CAN5 transmit interrupt	289	—
	RSCANFDCF5	CAN5 common FIFO access message buffers	—	49

NOTE

For the wake-up factors from standby mode, see **Section 14.1.2.1, Wake-Up Factors for Stand-By Modes**.

24.1.5 Reset Sources

RCFDCn reset sources are listed in the following table. RCFDCn is initialized by these reset sources.

Table 24.20 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RCFDCn	All reset sources (ISORES)

24.1.6 External Input/Output Signals

External input/output signals of RCFDCn are listed below.

Table 24.21 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
RCFDC0		
CANzRX (z = 0 to 7)	CANm receive data input	CANzRX (z = 0 to 7)
CANzTX (z = 0 to 7)	CANm transmit data output	CANzTX (z = 0 to 7)
RCFDC1		
CANzRX (z = 8 to 11)	CANm receive data input	CANzRX (z = 8 to 11)
CANzTX (z = 8 to 11)	CANm transmit data output	CANzTX (z = 8 to 11)

Table 24.22 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
RCFDC0		
CANzRX (z = 0 to 7)	CANm receive data input	CANzRX (z = 0 to 7)
CANzTX (z = 0 to 7)	CANm transmit data output	CANzTX (z = 0 to 7)

Table 24.23 External Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RCFDC0		
CANzRX (z = 0 to 5)	CANm receive data input	CANzRX (z = 0 to 5)
CANzTX (z = 0 to 5)	CANm transmit data output	CANzTX (z = 0 to 5)

24.2 Overview

24.2.1 Functional Overview

Table 24.24, RS-CANFD Module Specifications shows the RS-CANFD module specifications that has 8 channels (m = 0 to 7) for unit 0 (n = 0) and 4 channels (m = 0 to 3) for unit 1 (n = 1). **Figure 24.1, RS-CANFD Module Block Diagram** shows the RS-CANFD module block diagram.

Table 24.24 RS-CANFD Module Specifications

Item	Specification
Number of channels	8 (n = 0), 4 (n = 1)
Protocol	ISO11898-1 compliant Using CAN FD frames is selectable by switching interface modes.
Communication speed	Classical CAN only mode : • Maximum 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ CANm bit time = CANmTq × Tq count per bit $\text{CANmTq} = \frac{\text{(NBRP[9: 0] bits in the RCFDCnCFDCmNCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RCFDCnCFDGCFG register) CAN FD mode and CAN FD only mode: • Data bit rate: max.5 Mbps (Nominal bit rate ≤ 500 Kbps) • Data bit rate: max.2 Mbps (1Mbps ≥ Nominal bit rate > 500 Kbps) Note : fCAN = 40 MHz $\text{Transmission rate (CANm nominal bit time clock)} = \frac{1}{\text{CANm nominal bit time}}$ $\text{Transmission rate (CANm data bit time clock)} = \frac{1}{\text{CANm data bit time}}$ CANm nominal bit time = CANmTq(N) × Tq count per nominal bit CANm data bit time = CANmTq(D) × Tq count per data bit $\text{CANmTq(N)} = \frac{\text{(NBRP[9: 0] bits in the RCFDCnCFDCmNCFG register + 1)}}{\text{fCAN}}$ $\text{CANmTq(D)} = \frac{\text{(DBRP[7: 0] bits in the RCFDCnCFDCmDCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RCFDCnCFDGCFG register) m = 0 to 7 (n = 0), m = 0 to 3 with (n = 1) Tq: Time quantum

Table 24.24 RS-CANFD Module Specifications

Item	Specification
Buffer	RCFDC0: 1280 buffers in total ● Individual buffers: 256 buffers (32 buffers × 8 channels) Transmit buffer: 32 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 32 buffers allocatable) ● Shared buffers: 1024 buffers for all channels Receive buffer: 0 to 128 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) ● ECC included RCFDC1: 640 buffers in total ● Individual buffers: 128 buffers (32 buffers × 4 channels) Transmit buffer: 32 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 32 buffers allocatable) ● Shared buffers: 512 buffers for all channels Receive buffer: 0 to 64 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) ● ECC included
Reception function	● Receives data frames and remote frames. ● Selects ID format (standard ID, extended ID, or both IDs) to be received. ● Sets interrupt enable/disable for each FIFO. ● Mirror function (reception of messages transmitted from the own CAN node) ● Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	● Selects receive messages according to 1024 receive rules. ● Sets the number of receive rules (0 to 255) for each channel. ● Acceptance filter processing: Sets ID and mask for each receive rule. ● DLC filter processing: Enables DLC filter check for each acceptance rule.
Receive message transfer function	● Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer ● Label addition function Stores label information together with a message in a receive buffer and FIFO buffer.
Transmission function	● Transmits data frames and remote frames. ● Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. ● Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. ● Selects ID priority transmission or transmit buffer number priority transmission. ● Transmit request can be aborted (possible to confirm with a flag) ● One-shot transmission function
Interval transmission function	Transmits messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages
Gateway function	Transmits a received message automatically.
Bus off recovery mode selection	Selects the method for returning from bus off state. ● ISO11898-1 compliant ● Automatic entry to channel halt mode at bus-off entry ● Automatic entry to channel halt mode at bus-off end ● Transition to channel halt mode by program request ● Transition to the error-active state by program request (forcible return from the bus off state)

Table 24.24 RS-CANFD Module Specifications

Item	Specification
Error status monitoring	● Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus lock). ● Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) ● Reads the error counter. ● Monitors DLC errors.
Interrupt source	RCFDC0: 26 sources, RCFDC1: 14 sources ● Global interrupts (2 sources/unit) (n = 0, 1) Receive FIFO interrupt (1 source/unit) (n = 0, 1) Global error interrupt (1 source/unit) (n = 0, 1) ● Channel interrupts (3 sources/channel) CANm transmit interrupt (m = 0 to 7 (n = 0), m = 0 to 3 (n = 1)) 1. CANm transmit interrupt – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt 2. CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) 3. CANm error interrupt
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CANFD module.
CAN clock source	Selects the clkc or the clk_xincan. As for the range of operating frequency, see **Table 24.16, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM**.
Test function	Test function for user evaluation ● Listen-only mode ● Self-test mode 0 (external loopback) ● Self-test mode 1 (internal loopback) ● Restricted operation mode ● RAM test (read/write test) ● Inter-channel communication test [CRC error test enabled]

24.2.2 Interface Modes

The RS-CANFD has three interface modes.

- Classical CAN only mode: Handles only classical CAN frames.
- CAN FD mode: Handles classical CAN frames and CAN FD frames.
- CAN FD only mode: Handles only CAN FD frames.

Interface modes are switched by the CLOE bit and the FDOE in the RCFDCnCFDCmFDCFG register.

Set interface modes for each channel.

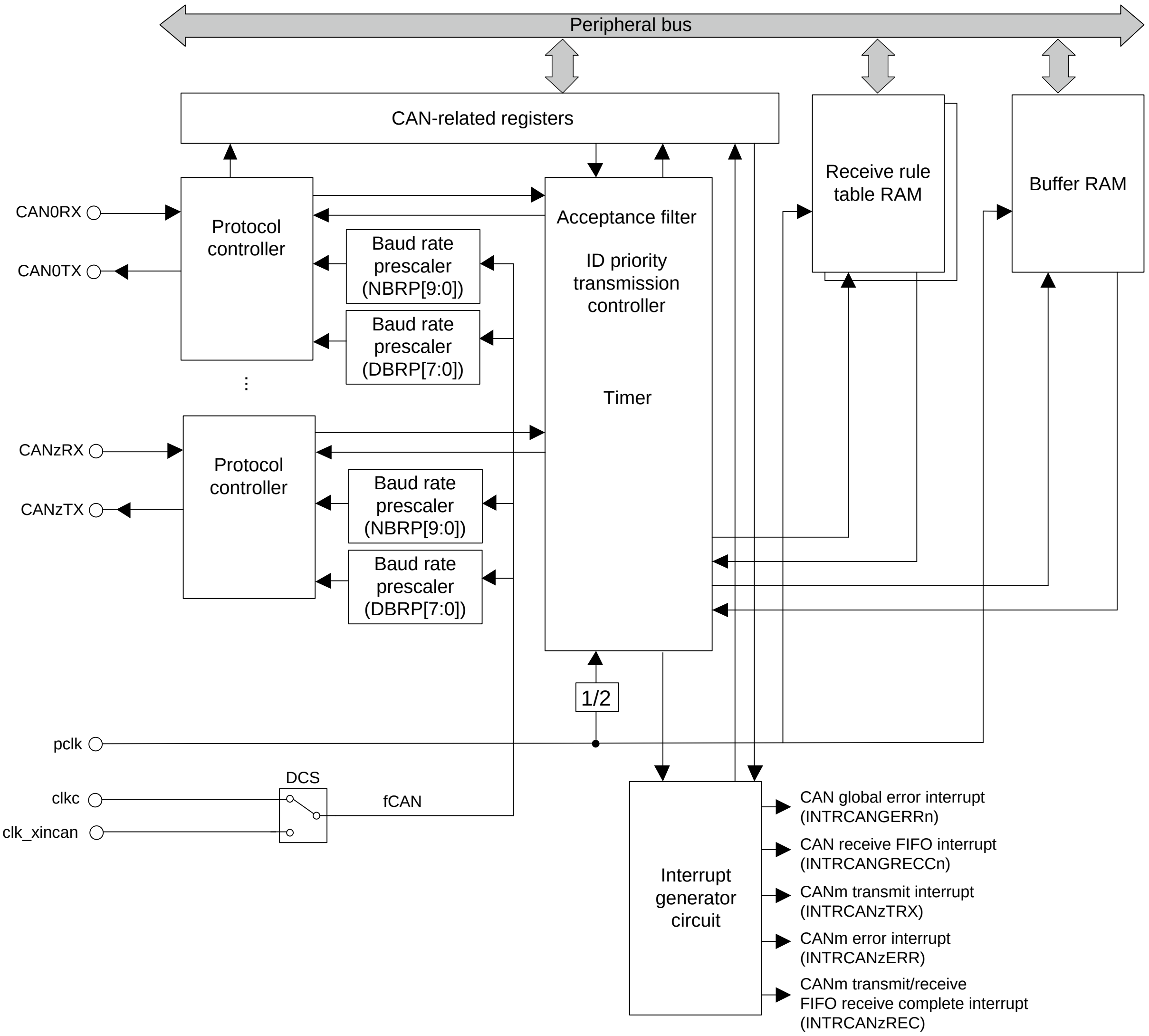
Table 24.25 Description and Configuration of Interface Mode

Interface Mode	Description	Configuration
CAN-FD mode	Classical and CAN-FD frames can be transmitted and received	RCFDCnCFDCmFDCFG.CLOE = 0 RCFDCnCFDCmFDCFG.FDOE = 0
CAN-FD only mode	Only CAN-FD frames can be transmitted and received. Classical frames will cause the detection of an error	RCFDCnCFDCmFDCFG.CLOE = 0 RCFDCnCFDCmFDCFG.FDOE = 1
Classical CAN only mode	Only Classical Frame can be transmitted and received. FD frames will cause the detection of an error	RCFDCnCFDCmFDCFG.CLOE = 1 RCFDCnCFDCmFDCFG.FDOE = 0

24.2.3 CAN FD Protocol

This product supports CAN FD according to the ISO/DIS 11898-1 protocol that specifies the new CRC field including stuff counters.

24.2.4 Block Diagram



Note: n = 0, 1
z = 0 to 11
m = 0 to 7 (n = 0) and m = 0 to 3 (n = 1)
NBRP[9:0]: Bits in the RCFDCnCFDCmNCFG register
DBRP[7:0]: Bits in the RCFDCnCFDCmDCFG register
DCS: Bits in the RCFDCnCFDGCFG register
fCAN: CAN clock

Figure 24.1 RS-CANFD Module Block Diagram

24.3 Registers

This section describes all registers to be used when the RS-CANFD is used.

24.3.1 List of Registers

The following tables list RS-CANFD registers to be used.

For details about <RCFDCn_base>, see **Section 24.1.2, Register Base Addresses**.

For details about registers initialized in Global reset mode or Channel reset mode, see following.

- **Table 24.118, Registers Initialized in Global Reset Mode or Channel Reset Mode**
- **Table 24.119, Registers Initialized Only in Global Reset Mode**

Table 24.26 List of Registers

Module Name	Register Name	Symbol	Address	Guard Group
Channel-related registers				
RCFDCn	Channel nominal bit rate configuration register	RCFDCnCFDCmNCFG	<RCFDCn_base> + 0000_H + (10_H × m)	RCFDCn Chm
RCFDCn	Channel control register	RCFDCnCFDCmCTR	<RCFDCn_base> + 0004_H + (10_H × m)	RCFDCn Chm
RCFDCn	Channel status register	RCFDCnCFDCmSTS	<RCFDCn_base> + 0008_H + (10_H × m)	RCFDCn Chm
RCFDCn	Channel error flag register	RCFDCnCFDCmERFL	<RCFDCn_base> + $000C_H$ + (10_H × m)	RCFDCn Chm
RCFDCn	Channel data bit rate configuration register	RCFDCnCFDCmDCFG	<RCFDCn_base> + 0700_H + (20_H × m)	RCFDCn Chm
RCFDCn	Channel CAN FD configuration register	RCFDCnCFDCmFDCFG	<RCFDCn_base> + 0704_H + (20_H × m)	RCFDCn Chm
RCFDCn	Channel CAN FD control register	RCFDCnCFDCmFDCTR	<RCFDCn_base> + 0708_H + (20_H × m)	RCFDCn Chm
RCFDCn	Channel CAN FD status register	RCFDCnCFDCmFDSTS	<RCFDCn_base> + $070C_H$ + (20_H × m)	RCFDCn Chm
RCFDCn	Channel CAN FD CRC register	RCFDCnCFDCmFDCRC	<RCFDCn_base> + 0710_H + (20_H × m)	RCFDCn Chm
Global-related registers				
RCFDCn	Global configuration register	RCFDCnCFDGCFG	<RCFDCn_base> + 0084_H	RCFDCn global
RCFDCn	Global control register	RCFDCnCFDGCTR	<RCFDCn_base> + 0088_H	RCFDCn global
RCFDCn	Global status register	RCFDCnCFDGSTS	<RCFDCn_base> + $008C_H$	RCFDCn global
RCFDCn	Global error flag register	RCFDCnCFDGERFL	<RCFDCn_base> + 0090_H	RCFDCn global
RCFDCn	Global timestamp counter register	RCFDCnCFDGTSC	<RCFDCn_base> + 0094_H	RCFDCn global
RCFDCn	Global TX interrupt status register 0	RCFDCnCFDGTINTSTS0	<RCFDCn_base> + 0610_H	RCFDCn global
RCFDCn	Global TX interrupt status register 1	RCFDCnCFDGTINTSTS1	<RCFDCn_base> + 0614_H	RCFDCn global
RCFDCn	Global FD configuration register	RCFDCnCFDGFDCFG	<RCFDCn_base> + 0624_H	RCFDCn global
Receive rule-related registers				
RCFDCn	Receive rule entry control register	RCFDCnCFDGAFLECTR	<RCFDCn_base> + 0098_H	RCFDCn global
RCFDCn	Receive rule configuration register 0	RCFDCnCFDGAFLCFG0	<RCFDCn_base> + $009C_H$	RCFDCn global
RCFDCn	Receive rule configuration register 1	RCFDCnCFDGAFLCFG1	<RCFDCn_base> + $00A0_H$	RCFDCn global
RCFDCn	Receive rule ID register	RCFDCnCFDGAFLIDj	<RCFDCn_base> + 1000_H + (10_H × j)	RCFDCn global
RCFDCn	Receive rule mask register	RCFDCnCFDGAFLMj	<RCFDCn_base> + 1004_H + (10_H × j)	RCFDCn global
RCFDCn	Receive rule pointer 0 register	RCFDCnCFDGAFLP0_j	<RCFDCn_base> + 1008_H + (10_H × j)	RCFDCn global
RCFDCn	Receive rule pointer 1 register	RCFDCnCFDGAFLP1_j	<RCFDCn_base> + $100C_H$ + (10_H × j)	RCFDCn global
Receive buffer-related registers				
RCFDCn	Receive buffer number register	RCFDCnCFDRMNB	<RCFDCn_base> + $00A4_H$	RCFDCn global
RCFDCn	Receive buffer new data register	RCFDCnCFDRMNDy	<RCFDCn_base> + $00A8_H$ + (04_H × y)	RCFDCn global
RCFDCn	Receive buffer ID register	RCFDCnCFDRMIDq	<RCFDCn_base> + 2000_H + (80_H × q)	RCFDCn global
RCFDCn	Receive buffer pointer register	RCFDCnCFDRMPTRq	<RCFDCn_base> + 2004_H + (80_H × q)	RCFDCn global
RCFDCn	Receive buffer CAN FD status register	RCFDCnCFDRMFDSTSq	<RCFDCn_base> + 2008_H + (80_H × q)	RCFDCn global
RCFDCn	Receive buffer data field register	RCFDCnCFDRMDFb_q	<RCFDCn_base> + $200C_H$ + (04_H × b) + (80_H × q)	RCFDCn global

Table 24.26 List of Registers

Module Name	Register Name	Symbol	Address	Guard Group
Receive FIFO buffer-related registers				
RCFDCn	Receive FIFO buffer configuration and control register	RCFDCnCFDRFCCx	<RCFDCn_base> + 00B8$_H$ + (04$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO buffer status register	RCFDCnCFDRFSTSx	<RCFDCn_base> + 00D8$_H$ + (04$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO buffer pointer control register	RCFDCnCFDRFPCTRx	<RCFDCn_base> + 00F8$_H$ + (04$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO buffer access id register	RCFDCnCFDRFIDx	<RCFDCn_base> + 6000$_H$ + (80$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO buffer access pointer register	RCFDCnCFDRFPTRx	<RCFDCn_base> + 6004$_H$ + (80$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO CAN FD status register	RCFDCnCFDRFFDSTSx	<RCFDCn_base> + 6008$_H$ + (80$_H$ × x)	RCFDCn global
RCFDCn	Receive FIFO buffer access data field register	RCFDCnCFDRFDFd_x	<RCFDCn_base> + 600C$_H$ + (04$_H$ × d) + (80$_H$ × x)	RCFDCn global
Transmit/receive FIFO buffer-related registers				
RCFDCn	Transmit/receive FIFO buffer configuration and control register	RCFDCnCFDCFCCk	<RCFDCn_base> + 0118$_H$ + (04$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer status register	RCFDCnCFDCFSTSk	<RCFDCn_base> + 0178$_H$ + (04$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer pointer control register	RCFDCnCFDCFPCTRk	<RCFDCn_base> + 01D8$_H$ + (04$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer access id register	RCFDCnCFDCFIDk	<RCFDCn_base> + 6400$_H$ + (80$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer access pointer register	RCFDCnCFDCFPTRk	<RCFDCn_base> + 6404$_H$ + (80$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO CAN FD configuration/status register	RCFDCnCFDCFFDCSTSk	<RCFDCn_base> + 6408$_H$ + (80$_H$ × k)	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer access data field register	RCFDCnCFDCFDFd_k	<RCFDCn_base> + 640C$_H$ + (04$_H$ × d) + (80$_H$ × k)	RCFDCn global
FIFO status-related registers				
RCFDCn	FIFO empty status register	RCFDCnCFDFESTS	<RCFDCn_base> + 0238$_H$	RCFDCn global
RCFDCn	FIFO full status register	RCFDCnCFDFFSTS	<RCFDCn_base> + 023C$_H$	RCFDCn global
RCFDCn	FIFO message lost status register	RCFDCnCFDFMSTS	<RCFDCn_base> + 0240$_H$	RCFDCn global
RCFDCn	Receive FIFO buffer interrupt flag status register	RCFDCnCFDRFISTS	<RCFDCn_base> + 0244$_H$	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer receive interrupt flag status register	RCFDCnCFDCFRISTS	<RCFDCn_base> + 0248$_H$	RCFDCn global
RCFDCn	Transmit/receive FIFO buffer transmit interrupt flag status register	RCFDCnCFDCFTISTS	<RCFDCn_base> + 024C$_H$	RCFDCn global
FIFO DMA-related registers				
RCFDCn	DMA enable register	RCFDCnCFDCDTCT	<RCFDCn_base> + 0640$_H$	RCFDCn global
RCFDCn	DMA status register	RCFDCnCFDCDTSTS	<RCFDCn_base> + 0644$_H$	RCFDCn global
Transmit buffer-related registers				
RCFDCn	Transmit buffer control register	RCFDCnCFDTMCp	<RCFDCn_base> + 0250$_H$ + (01$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer status register	RCFDCnCFDTMSTSp	<RCFDCn_base> + 0350$_H$ + (01$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer ID register	RCFDCnCFDTMIDp	<RCFDCn_base> + 8000$_H$ + (80$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer pointer register	RCFDCnCFDTMPTRp	<RCFDCn_base> + 8004$_H$ + (80$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer CAN FD configuration register	RCFDCnCFDTMFDCTRp	<RCFDCn_base> + 8008$_H$ + (80$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer data field register	RCFDCnCFDTMDFb_p	<RCFDCn_base> + 800C$_H$ + (04$_H$ × b) + (80$_H$ × p)	RCFDCn global
RCFDCn	Transmit buffer interrupt enable configuration register	RCFDCnCFDTMIECm	<RCFDCn_base> + 04D0$_H$ + (04$_H$ × m)	RCFDCn global
Transmit buffer status-related registers				
RCFDCn	Transmit buffer transmit request status register	RCFDCnCFDTMTRSTSm	<RCFDCn_base> + 0450$_H$ + (04$_H$ × m)	RCFDCn global
RCFDCn	Transmit buffer transmit abort request status register	RCFDCnCFDTMTARSTSm	<RCFDCn_base> + 0470$_H$ + (04$_H$ × m)	RCFDCn global
RCFDCn	Transmit buffer transmit complete status register	RCFDCnCFDTMTCSTSm	<RCFDCn_base> + 0490$_H$ + (04$_H$ × m)	RCFDCn global
RCFDCn	Transmit buffer transmit abort status register	RCFDCnCFDTMTASTSm	<RCFDCn_base> + 04B0$_H$ + (04$_H$ × m)	RCFDCn global

Table 24.26 List of Registers

Module Name	Register Name	Symbol	Address	Guard Group
Transmit queue-related registers				
RCFDCn	Transmit queue configuration and control register	RCFDCnCFDTXQCCm	<RCFDCn_base> + 0550_H + (04_H × m)	RCFDCn Chm
RCFDCn	Transmit queue status register	RCFDCnCFDTXQSTSm	<RCFDCn_base> + 0570_H + (04_H × m)	RCFDCn Chm
RCFDCn	Transmit queue pointer control register	RCFDCnCFDTXQPCTRm	<RCFDCn_base> + 0590_H + (04_H × m)	RCFDCn Chm
Transmit history-related registers				
RCFDCn	Transmit history configuration and control register	RCFDCnCFDTHLCCm	<RCFDCn_base> + $05B0_H$ + (04_H × m)	RCFDCn Chm
RCFDCn	Transmit history status register	RCFDCnCFDTHLSTSm	<RCFDCn_base> + $05D0_H$ + (04_H × m)	RCFDCn Chm
RCFDCn	Transmit history pointer control register	RCFDCnCFDTHLPCTRm	<RCFDCn_base> + $05F0_H$ + (04_H × m)	RCFDCn Chm
RCFDCn	Transmit history access register 0	RCFDCnCFDTHLACC0m	<RCFDCn_base> + 10000_H + (08_H × m)	RCFDCn Chm
RCFDCn	Transmit history access register 1	RCFDCnCFDTHLACC1m	<RCFDCn_base> + 10004_H + (08_H × m)	RCFDCn Chm
Test-related registers				
RCFDCn	Global test configuration register	RCFDCnCFDGTSTCFG	<RCFDCn_base> + 0618_H	RCFDCn global
RCFDCn	Global test control register	RCFDCnCFDGTSTCTR	<RCFDCn_base> + $061C_H$	RCFDCn global
RCFDCn	Global lock key register	RCFDCnCFDGLOCKK	<RCFDCn_base> + $062C_H$	RCFDCn global
RCFDCn	RAM test page access register	RCFDCnCFDRPGACCr	<RCFDCn_base> + 10400_H + (04_H × r)	RCFDCn global

Table 24.27 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 32 × m + 0
	Transmit buffer 32 × m + 1
	Transmit buffer 32 × m + 2
	Transmit buffer 32 × m + 3
	Transmit buffer 32 × m + 4
	Transmit buffer 32 × m + 5
	Transmit buffer 32 × m + 6
	Transmit buffer 32 × m + 7
	Transmit buffer 32 × m + 8
	Transmit buffer 32 × m + 9
	Transmit buffer 32 × m + 10
	Transmit buffer 32 × m + 11
	Transmit buffer 32 × m + 12
	Transmit buffer 32 × m + 13
	Transmit buffer 32 × m + 14
	Transmit buffer 32 × m + 15
	Transmit buffer 32 × m + 16
	Transmit buffer 32 × m + 17
	Transmit buffer 32 × m + 18
	Transmit buffer 32 × m + 19
	Transmit buffer 32 × m + 20
	Transmit buffer 32 × m + 21
	Transmit buffer 32 × m + 22
	Transmit buffer 32 × m + 23
	Transmit buffer 32 × m + 24
	Transmit buffer 32 × m + 25
	Transmit buffer 32 × m + 26
	Transmit buffer 32 × m + 27
	Transmit buffer 32 × m + 28
	Transmit buffer 32 × m + 29
	Transmit buffer 32 × m + 30
	Transmit buffer 32 × m + 31

Table 24.28 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 24.29 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[4:0]

Setting of Bits CFTML[4:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
00000_B	Transmit buffer 32 × m + 0
00001_B	Transmit buffer 32 × m + 1
00010_B	Transmit buffer 32 × m + 2
00011_B	Transmit buffer 32 × m + 3
00100_B	Transmit buffer 32 × m + 4
00101_B	Transmit buffer 32 × m + 5
00110_B	Transmit buffer 32 × m + 6
00111_B	Transmit buffer 32 × m + 7
01000_B	Transmit buffer 32 × m + 8
01001_B	Transmit buffer 32 × m + 9
01010_B	Transmit buffer 32 × m + 10
01011_B	Transmit buffer 32 × m + 11
01100_B	Transmit buffer 32 × m + 12
01101_B	Transmit buffer 32 × m + 13
01110_B	Transmit buffer 32 × m + 14
01111_B	Transmit buffer 32 × m + 15
10000_B	Transmit buffer 32 × m + 16
10001_B	Transmit buffer 32 × m + 17
10010_B	Transmit buffer 32 × m + 18
10011_B	Transmit buffer 32 × m + 19
10100_B	Transmit buffer 32 × m + 20
10101_B	Transmit buffer 32 × m + 21
10110_B	Transmit buffer 32 × m + 22
10111_B	Transmit buffer 32 × m + 23
11000_B	Transmit buffer 32 × m + 24
11001_B	Transmit buffer 32 × m + 25
11010_B	Transmit buffer 32 × m + 26
11011_B	Transmit buffer 32 × m + 27
11100_B	Transmit buffer 32 × m + 28
11101_B	Transmit buffer 32 × m + 29
11110_B	Transmit buffer 32 × m + 30
11111_B	Transmit buffer 32 × m + 31

Table 24.30 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[4:0]	Transmit Buffer p Allocated to the Transmit Queue
00000_B	Setting prohibited
00001_B	Setting prohibited
00010_B	Transmit buffer 32 × m + 31 to 32 × m + 29
00011_B	Transmit buffer 32 × m + 31 to 32 × m + 28
00100_B	Transmit buffer 32 × m + 31 to 32 × m + 27
00101_B	Transmit buffer 32 × m + 31 to 32 × m + 26
00110_B	Transmit buffer 32 × m + 31 to 32 × m + 25
00111_B	Transmit buffer 32 × m + 31 to 32 × m + 24
01000_B	Transmit buffer 32 × m + 31 to 32 × m + 23
01001_B	Transmit buffer 32 × m + 31 to 32 × m + 22
01010_B	Transmit buffer 32 × m + 31 to 32 × m + 21
01011_B	Transmit buffer 32 × m + 31 to 32 × m + 20
01100_B	Transmit buffer 32 × m + 31 to 32 × m + 19
01101_B	Transmit buffer 32 × m + 31 to 32 × m + 18
01110_B	Transmit buffer 32 × m + 31 to 32 × m + 17
01111_B	Transmit buffer 32 × m + 31 to 32 × m + 16
10000_B	Transmit buffer 32 × m + 31 to 32 × m + 15
10001_B	Transmit buffer 32 × m + 31 to 32 × m + 14
10010_B	Transmit buffer 32 × m + 31 to 32 × m + 13
10011_B	Transmit buffer 32 × m + 31 to 32 × m + 12
10100_B	Transmit buffer 32 × m + 31 to 32 × m + 11
10101_B	Transmit buffer 32 × m + 31 to 32 × m + 10
10110_B	Transmit buffer 32 × m + 31 to 32 × m + 9
10111_B	Transmit buffer 32 × m + 31 to 32 × m + 8
11000_B	Transmit buffer 32 × m + 31 to 32 × m + 7
11001_B	Transmit buffer 32 × m + 31 to 32 × m + 6
11010_B	Transmit buffer 32 × m + 31 to 32 × m + 5
11011_B	Transmit buffer 32 × m + 31 to 32 × m + 4
11100_B	Transmit buffer 32 × m + 31 to 32 × m + 3
11101_B	Transmit buffer 32 × m + 31 to 32 × m + 2
11110_B	Transmit buffer 32 × m + 31 to 32 × m + 1
11111_B	Transmit buffer 32 × m + 31 to 32 × m + 0

24.3.2 Details of Channel-related Registers

24.3.2.1 RCFDCnCFDCmNCFG — Channel Nominal Bit Rate Configuration Register (m = 0 to 7)

Access: RCFDCnCFDCmNCFG register can be read or written in 32-bit units

RCFDCnCFDCmNCFGL, RCFDCnCFDCmNCFGH registers can be read or written in 16-bit units

RCFDCnCFDCmNCFGLL, RCFDCnCFDCmNCFGLH, RCFDCnCFDCmNCFGHL, RCFDCnCFDCmNCFGHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCmNCFG: <RCFDCn_base> + 0000_H + (10_H × m)

RCFDCnCFDCmNCFGL: <RCFDCn_base> + 0000_H + (10_H × m),

RCFDCnCFDCmNCFGH: <RCFDCn_base> + 0002_H + (10_H × m)

RCFDCnCFDCmNCFGLL: <RCFDCn_base> + 0000_H + (10_H × m),

RCFDCnCFDCmNCFGLH: <RCFDCn_base> + 0001_H + (10_H × m),

RCFDCnCFDCmNCFGHL: <RCFDCn_base> + 0002_H + (10_H × m),

RCFDCnCFDCmNCFGHH: <RCFDCn_base> + 0003_H + (10_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NTSEG2[4:0]					—	NTSEG1[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NSJW[4:0]					—	NBRP[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.31 RCFDCnCFDCmNCFG Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
28 to 24	NTSEG2[4:0]	Nominal Bit Rate Time Segment 2 Control b28 b27 b26 b25 b24 0 0 0 0 0: Setting prohibited 0 0 0 0 1: 2 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
22 to 16	NTSEG1[6:0]	Nominal Bit Rate Time Segment 1 Control b22 b21 b20 b19 b18 b17 b16 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 1: Setting prohibited 0 0 0 0 0 1 0: Setting prohibited 0 0 0 0 0 1 1: 4 Tq : : 1 1 1 1 1 1 0: 127 Tq 1 1 1 1 1 1 1: 128 Tq

Table 24.31 RCFDCnCFDCmNCFG Register Contents

Bit Position	Bit Name	Function
15 to 11	NSJW[4:0]	Nominal Bit Rate Resynchronization Jump Width Control b15 b14 b13 b12 b11 0 0 0 0 0: 1 Tq 0 0 0 0 1: 2 Tq 0 0 0 1 0: 3 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
10	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
9 to 0	NBRP[9:0]	Nominal Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 1023), the nominal bit rate prescaler divides fCAN by (P + 1).

Modify the RCFDCnCFDCmNCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. For the description and settings for bit timing parameters, see **Section 24.10.1, Initial Settings**.

NTSEG2[4:0] Bits

These bits specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate.

Possible values are 2 Tq to 32 Tq, inclusive.

Set a value smaller than the value of the NTSEG1[6:0] bits.

NTSEG1[6:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of nominal bit rate as a Tq value.

Possible values are 4 to 128 Tq.

NSJW[4:0] Bits

These bits specify the resynchronization jump width of nominal bit rate as a Tq value. Possible values are 1 to 32 Tq. Specify a value equal to or smaller than the NTSEG2[4:0] value.

NBRP[9:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the nominal bit rate prescaler ((NBRP[9:0]) + 1) becomes CANmTq(N) clock (fCANTQ(N)m). One clock of the CANmTq(N) clock becomes one Time Quantum (Tq).

The NBRP[9:0] value and the DBRP[7:0] value should be equal and the two corresponding bit rate values are different according to the respective segment values.

24.3.2.2 RCFDCnCFDCmCTR — Channel Control Register (m = 0 to 7)

Access: RCFDCnCFDCmCTR register can be read or written in 32-bit units

RCFDCnCFDCmCTRL, RCFDCnCFDCmCTRH registers can be read or written in 16-bit units

RCFDCnCFDCmCTRLL, RCFDCnCFDCmCTRLH, RCFDCnCFDCmCTRHL, RCFDCnCFDCmCTRHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCmCTR: <RCFDCn_base> + 0004_H + (10_H × m)

RCFDCnCFDCmCTRL: <RCFDCn_base> + 0004_H + (10_H × m),

RCFDCnCFDCmCTRH: <RCFDCn_base> + 0006_H + (10_H × m)

RCFDCnCFDCmCTRLL: <RCFDCn_base> + 0004_H + (10_H × m),

RCFDCnCFDCmCTRLH: <RCFDCn_base> + 0005_H + (10_H × m),

RCFDCnCFDCmCTRHL: <RCFDCn_base> + 0006_H + (10_H × m),

RCFDCnCFDCmCTRHH: <RCFDCn_base> + 0007_H + (10_H × m)

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ROM	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.32 RCFDCnCFDCmCTR Register Contents

Bit Position	Bit Name	Function
31	ROM	Restricted Operation Mode Enable 0: Restricted operation mode is disabled. 1: Restricted operation mode is enabled.
30	CRCT	CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted.
29 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RCFDCnCFDCmERFL register are all cleared. 1: Error flags for all error information are displayed.

Table 24.32 RCFDCnCFDCmCTR Register Contents

Bit Position	Bit Name	Function
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
19	TDCVFIE	Transmitter Delay Compensation Violation Interrupt Enable 0: A transmitter delay compensation violation interrupt is disabled. 1: A transmitter delay compensation violation interrupt is enabled.
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: A successful occurrence counter overflow interrupt is disabled. 1: A successful occurrence counter overflow interrupt is enabled.
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: An error occurrence counter overflow interrupt is disabled. 1: An error occurrence counter overflow interrupt is enabled.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.

Table 24.32 RCFDCnCFDCmCTR Register Contents

Bit Position	Bit Name	Function
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

ROM Bit

When the ROM bit and the CTME bit in the RCFDCnCFDCmCTR register are set to 1, restricted operation mode is enabled. Use the restricted operation mode only when the CTMS[1:0] value in the RCFDCnCFDCmCTR register is 00_B (standard test mode). Modify this bit only in channel halt mode.

This bit is always 0 in channel reset mode.

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RCFDCnCFDCmERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RCFDCnCFDCmCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RCFDCnCFDGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is set to 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RCFDCnCFDCmERFL register. When this bit is clear to 0, if any error is detected while the flags of bits 14 to 8 in the RCFDCnCFDCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RCFDCnCFDCmCTR register (m = 0 to 7) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RCFDCnCFDCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CANFD module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the program’s writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TDCVFIE Bit

When the TDCVF flag in the RCFDCnCFDCmFDSTS register is set to 1 after the TDCVFIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

SOCOIE Bit

When the SOCO flag in the RCFDCnCFDCmFDSTS register is set to 1 after the SOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

EOCOIE Bit

When the EOCO flag in the RCFDCnCFDCmFDSTS register is set to 1 after the EOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RCFDCnCFDCmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RCFDCnCFDCmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RCFDCnCFDCmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RCFDCnCFDCmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RCFDCnCFDCmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RCFDCnCFDCmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RCFDCnCFDCmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RCFDCnCFDCmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly returns from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RCFDCnCFDCmSTS register to 00_H and also clears the BOSTS flag in the RCFDCnCFDCmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RCFDCnCFDCmCTR register are 00_B (ISO11898-1 compliant). A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RS-CANFD module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode. Clearing this bit to 0 makes the channel exit channel stop mode.

This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 24.5.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B.

When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits are automatically set to 10_B.

24.3.2.3 RCFDCnCFDCmSTS — Channel Status Register (m = 0 to 7)

Access: RCFDCnCFDCmSTS register can be read or written in 32-bit units

RCFDCnCFDCmSTSL register can be read or written in 16-bit units

RCFDCnCFDCmSTSH register is a read-only register that can be read in 16-bit units

RCFDCnCFDCmSTSLL register is a read-only register that can be read in 8-bit units

RCFDCnCFDCmSTSLH register can be read or written in 8-bit units

RCFDCnCFDCmSTSHL, RCFDCnCFDCmSTSHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCmSTS: <RCFDCn_base> + 0008_{H} + (10_{H} × m)

RCFDCnCFDCmSTSL: <RCFDCn_base> + 0008_{H} + (10_{H} × m),

RCFDCnCFDCmSTSH: <RCFDCn_base> + $000A_{H}$ + (10_{H} × m)

RCFDCnCFDCmSTSLL: <RCFDCn_base> + 0008_{H} + (10_{H} × m),

RCFDCnCFDCmSTSLH: <RCFDCn_base> + 0009_{H} + (10_{H} × m),

RCFDCnCFDCmSTSHL: <RCFDCn_base> + $000A_{H}$ + (10_{H} × m),

RCFDCnCFDCmSTSHH: <RCFDCn_base> + $000B_{H}$ + (10_{H} × m)

Value after reset: $0000\ 0005_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R/W*[1]	R	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.33 RCFDCnCFDCmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	ESIF	Error State Indication Flag 0: No CAN FD message whose ESI bit is recessive has been received. 1: At least one CAN FD message whose ESI bit is recessive has been received.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state

Table 24.33 RCFDCnCFDCmSTS Register Contents

Bit Position	Bit Name	Function
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

ESIF Flag

When the recessive ESI bit is detected in a successfully received message, this flag is set to 1. In loopback mode or mirror mode, the own transmission message is regarded as a received message. To clear this flag to 0, write 0 to this bit by the program. This bit cannot be set to 1 by the program. If the flag setting (to 1) timing matches the writing 0 (by the program) timing, this flag is set to 1.

This flag is set to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag is set to 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

24.3.2.4 RCFDCnCFDCmERFL — Channel Error Flag Register (m = 0 to 7)

Access:	RCFDCnCFDCmERFL register can be read or written in 32-bit units
	RCFDCnCFDCmERFLL register can be read or written in 16-bit units
	RCFDCnCFDCmERFLH register is a read-only register that can be read in 16-bit units
	RCFDCnCFDCmERFLLL, RCFDCnCFDCmERFLLH registers can be read or written in 8-bit units
	RCFDCnCFDCmERFLHL, RCFDCnCFDCmERFLHH registers are read-only registers that can be read in 8-bit units
Address:	RCFDCnCFDCmERFL: <RCFDCn_base> + 000C$_H$ + (10$_H$ × m)
	RCFDCnCFDCmERFLL: <RCFDCn_base> + 000C$_H$ + (10$_H$ × m),
	RCFDCnCFDCmERFLH: <RCFDCn_base> + 000E$_H$ + (10$_H$ × m)
	RCFDCnCFDCmERFLLL: <RCFDCn_base> + 000C$_H$ + (10$_H$ × m),
	RCFDCnCFDCmERFLLH: <RCFDCn_base> + 000D$_H$ + (10$_H$ × m),
	RCFDCnCFDCmERFLHL: <RCFDCn_base> + 000E$_H$ + (10$_H$ × m),
	RCFDCnCFDCmERFLHH: <RCFDCn_base> + 000F$_H$ + (10$_H$ × m)
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]	R/W[*1]

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.34 RCFDCnCFDCmERFL Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data (CRC length:15 bits) A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.

Table 24.34 RCFDCnCFDCmERFL Register Contents

Bit Position	Bit Name	Function
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is set to 1. Transition to channel reset mode resets these flags to 0.

If the ERRD bit in the RCFDCnCFDCmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RCFDCnCFDCmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RCFDCnCFDCmCTR register is set to 1 (communication test mode is enabled), if transmit or receive message is a classical CAN frame (CRC length = 15 bits), this flag is updated and the CRC value calculated based on the message can be read. When a CAN FD frame is sent or received, the value of CRCREG[20:0] bits in the RCFDCnCFDCmFDCRC register is updated. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.

This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected even though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected even though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of bus lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RCFDCnCFDCmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RCFDCnCFDCmCTR register is set to 1 (forcible return from the bus off state).
- The BOM[1:0] bits in the RCFDCnCFDCmCTR register are set to 01_B (transition to channel halt mode at bus off entry).

- The CHMDC[1:0] bits in the RCFDCnCFDCmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RCFDCnCFDCmCTR register (m = 0 to 7) are set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag is set to 1 when the error passive state is entered ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])). This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RCFDCnCFDCmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

24.3.2.5 RCFDCnCFDCmDCFG — Channel Data Bit Rate Configuration Register (m = 0 to 7)

Access: RCFDCnCFDCmDCFG register can be read or written in 32-bit units

RCFDCnCFDCmDCFGL, RCFDCnCFDCmDCFGH registers can be read or written in 16-bit units

RCFDCnCFDCmDCFGLL, RCFDCnCFDCmDCFGHL, RCFDCnCFDCmDCFGHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCmDCFG: <RCFDCn_base> + 0700_H + (20_H × m)
RCFDCnCFDCmDCFGL: <RCFDCn_base> + 0700_H + (20_H × m),
RCFDCnCFDCmDCFGH: <RCFDCn_base> + 0702_H + (20_H × m)
RCFDCnCFDCmDCFGLL: <RCFDCn_base> + 0700_H + (20_H × m),
RCFDCnCFDCmDCFGHL: <RCFDCn_base> + 0702_H + (20_H × m),
RCFDCnCFDCmDCFGHH: <RCFDCn_base> + 0703_H + (20_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSJW[2:0]			—	DTSEG2[2:0]			DTSEG1[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DBRP[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.35 RCFDCnCFDCmDCFG Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26 to 24	DSJW[2:0]	Data Bit Rate Resynchronization Jump Width Control b26 b25 b24 0 0 0: 1 Tq 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
22 to 20	DTSEG2[2:0]	Data Bit Rate Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 24.35 RCFDCnCFDCmDCFG Register Contents

Bit Position	Bit Name	Function
19 to 16	DTSEG1[3:0]	Data Bit Rate Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: 2 Tq 0 0 1 0: 3 Tq 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7 to 0	DBRP[7:0]	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).

Modify the RCFDCnCFDCmDCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. When only classical CAN frames are used in CAN FD mode, set the RCFDCnCFDCmDCFG register to the value equal to the set RCFDCnCFDCmNCFG register value. For the description and settings of bit timing parameters, see **Section 24.10.1, Initial Settings**.

The channel of Classical only mode does not have to perform the configuration of this register.

DSJW[2:0] Bits

These bits specify the resynchronization jump width of data bit rate as a Tq value. Possible values are 1 to 8 Tq. Specify a value equal to or smaller than the DTSEG2[2:0] bits value.

DTSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate.

Possible values are 2 to 8 Tq.

Set a value less than or equal to the value of the DTSEG1[3:0] bits.

DTSEG1[3:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of data bit rate as a Tq value.

Possible values are 2 to 16 Tq.

DBRP[7:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the data bit rate prescaler ((DBRP[7:0]) + 1) becomes CANmTq(D) clock (fCANTQ(D)m). One clock of the CANmTq(D) clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].

To specify different values for the nominal bit rate and the data bit rate, change the values of the RCFDCnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RCFDCnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RCFDCnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

24.3.2.6 RCFDCnCFDCmFDCFG — Channel CAN FD Configuration Register (m = 0 to 7)

Access: RCFDCnCFDCmFDCFG register can be read or written in 32-bit units

RCFDCnCFDCmFDCFGL, RCFDCnCFDCmFDCFGH registers can be read or written in 16-bit units

RCFDCnCFDCmFDCFGLL, RCFDCnCFDCmFDCFGLH, RCFDCnCFDCmFDCFGHL, RCFDCnCFDCmFDCFGHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCmFDCFG: <RCFDCn_base> + 0704_H + (20_H × m)

RCFDCnCFDCmFDCFGL: <RCFDCn_base> + 0704_H + (20_H × m),

RCFDCnCFDCmFDCFGH: <RCFDCn_base> + 0706_H + (20_H × m)

RCFDCnCFDCmFDCFGLL: <RCFDCn_base> + 0704_H + (20_H × m),

RCFDCnCFDCmFDCFGLH: <RCFDCn_base> + 0705_H + (20_H × m),

RCFDCnCFDCmFDCFGHL: <RCFDCn_base> + 0706_H + (20_H × m),

RCFDCnCFDCmFDCFGHH: <RCFDCn_base> + 0707_H + (20_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CLOE	REFE	FDOE	—	GWBRS	GWFDF	GWEN	—	TDCO[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 24.36 RCFDCnCFDCmFDCFG Register Contents

Bit Position	Bit Name	Function
31	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
30	CLOE	Classical CAN only mode enable bit 0: Classical CAN only mode is disabled 1: Classical CAN only mode is enabled
29	REFE	Reception data edge filter enable bit 0: Reception data edge filter is disabled 1: Reception data edge filter is enabled
28	FDOE	FD-only mode enable bit 0: FD-only mode is disabled 1: FD-only mode is enabled
27	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
26	GWBRS	Gateway BRS 0: A frame is transmitted with the BRS bit in the received frame set to 0. 1: A frame is transmitted with the BRS bit in the received frame set to 1.
25	GWFDF	Gateway FDF 0: A frame is transmitted regarding the received frame as a classical CAN frame. 1: A frame is transmitted regarding the received frame as a CAN FD frame.
24	GWEN	CAN-CAN FD Gateway Enable 0: The CAN-CAN FD gateway is disabled. 1: The CAN-CAN FD gateway is enabled.
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.

Table 24.36 RCFDCnCFDCmFDCFG Register Contents

Bit Position	Bit Name	Function
22 to 16	TDCO[6:0]	Transmitter Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.
15 to 11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10	ESIC	Error State Display Mode Select 0: Always displays the node error state. 1: When the node is not in the error passive state: Displays the message buffer error state. When the node is in the error passive state: Displays the node error state.
9	TDCE	Transmitter Delay Compensation Enable 0: Transmitter delay compensation is disabled. 1: Transmitter delay compensation is enabled.
8	TDCOC	Transmitter Delay Compensation Measurement Select 0: Measurement and offset 1: Only offset
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2 to 0	EOCCFG[2:0]	Error Occurrence Counting Method Select b2 b1 b0 0 0 0: All transmit messages and receive messages 0 0 1: All transmit messages 0 1 0: All receive messages 0 1 1: Setting prohibited 1 0 0: Only data phase of transmitted or received CAN FD message 1 0 1: Only data phase of transmitted CAN FD message 1 1 0: Only data phase of received CAN FD message 1 1 1: Setting prohibited

CLOE Bit

Setting this bit to 1 enables Classical CAN only mode. When data is transmitted, a classical CAN frame will be sent. When a CAN FD frame is received, a form error or a CRC error is detected.

Modify this bit only in channel reset mode.

Do not set RCFDCnCFDCmFDCFG.CLOE and RCFDCnCFDCmFDCFG.FDOE simultaneously.

CLOE Bit	FDOE Bit	Interface Mode
0	0	CAN-FD mode
0	1	FD only mode
1	0	Classical CAN only mode
1	1	Reserved

REFE Bit

Setting this bit to 1 enables reception data edge filtering when the idle condition is detected, and a dominant level with less than 2 time quanta is ignored. A dominant level with more than or equal to 2 time quanta is detected as an edge. Modify this bit only in channel reset mode.

Setting this bit to 0 when using in Classical CAN only mode.

FDOE Bit

Setting this bit to 1 enables FD-only mode. When data is transmitted, a CAN FD frame will be sent regardless of the settings to the CFFDF bit in the RCFDCnCFDCFFDCSTSk register or the TMFDF bit in the RCFDCnCFDTMFDCTRp register. When a classical CAN frame is received, a form error is detected. Modify this bit only in channel reset mode.

In case a Classical frame is confiured for transmitting, the FDF bit is sent as recessive, so a FD frame is sent.

If the DLC is configured bigger than 8, the remaining data bytes are padded with CCh.

Do not set RCFDCnCFDCmFDCFG.CLOE and RCFDCnCFDCmFDCFG.FDOE simultaneously.

GWBRS Bit

When the GWEN bit is 1, the BRS bit in a CAN FD frame to be transmitted by the gateway function is set.

When the GWFDF bit is set to 0, write 0 to this bit. Modify this bit only in channel reset mode.

GWFDF Bit

When the GWEN bit is 1, the FDF bit in a CAN FD frame to be transmitted by the gateway function is set. Modify this bit only in channel reset mode.

GWEN Bit

This bit is used to control the operation of the transmit/receive FIFO buffer with the CFM[1:0] bits in the RCFDCnCFDCFCCk register set to 10_B (gateway mode).

Setting this bit to 1 enables the CAN-CAN FD gateway, enabling transmission in a format different from that of frames received by the gateway function. Received frames are replaced in accordance with the settings of the GWFDF bit and the GWBRS bit. When the DLC value in the received classical CAN frame is 1001_B or more and the GWFDF bit is set to 1 (CAN FD frame), the DLC value is replaced with 1000_B.

While this bit is set to 1, do not perform routing the following frames by using the gateway function.

- Remote frames

Modify this bit only in channel reset mode.

Table 24.37, Operation when the CAN-CAN FD Gateway is Enabled shows the settings and formats of transmit frame and receive frame while the CAN- CAN FD gateway is enabled.

Table 24.37 Operation when the CAN-CAN FD Gateway is Enabled

Receive Frame			GWFDF Bit	Transmit Frame		
Format	BRS Bit	Received DLC Value		Format	BRS Bit	DLC Value to be Transmitted
Classical CAN	None	DLC ≤ 1000_B	0	Classical CAN	None	Not replaced
		DLC > 1000_B				
CAN FD	Arbitrary	DLC ≤ 1000_B				
		DLC > 1000_B				Replaced with 1000_B
Classical CAN	None	DLC ≤ 1000_B	1	CAN FD	According to GWBRS bit setting	Not replaced
		DLC >1000_B				Replaced with 1000_B
CAN FD	Arbitrary	DLC ≤ 1000_B				Not replaced
		DLC > 1000_B				

TDCO[6:0] Bits

These bits set the SSP offset value. How to use this value depends on the TDCOC bit in the RCFDCnCFDCmFDCFG register.

These bits are based on CAN clock frequency(fCAN).

When the TDCOC bit is set to 0, the transmitter delay compensation result equals to the total value of the measured delay value and the TDCO[6:0] value (rounded down to the nearest integer Tq).

When the TDCOC bit is set to 1, the transmitter delay compensation result equals to the TDCO[6:0] value. The SSP offset value = (set value of TDCO[6:0] bits + 1).

Modify these bits only in channel reset mode or channel halt mode.

ESIC Bit

When the ESIC bit is set to 1, if the channel is in the error active state, the ESI bit value (CFESI bit in the RCFDCnCFDCFFDCSTSk register or TMESI bit in the RCFDCnCFDTMFDCTRp register) set in the transmit/receive FIFO buffer or transmit buffer is transmitted as an ESI bit value of the transmit message. When the channel is in the error passive state or the ESIC bit is set to 0, the channel status is transmitted as an ESI bit value. Modify this bit only in channel reset mode or channel halt mode

Table 24.38 ESI Value to Be Transmitted

ESIC Bit	Channel Status	ESI Value to be Transmitted
0	Error active	0 (error active node)
	Error passive	1 (error passive node)
1	Error active	ESI value set in the transmit/receive FIFO buffer or transmit buffer (CFESI bit in the RCFDCnCFDCFFDCSTSk register or TMESI bit in the RCFDCnCFDTMFDCTRp register)
	Error passive	1 (error passive node)

TDCE Bit

Setting this bit to 1 enables transmitter delay compensation. Modify this bit only in channel reset mode or channel halt mode.

TDCOC Bit

When this bit is set to 0, the SSP position is defined by the total of the measured delay value and the SSP offset value (fixed value).

When this bit is set to 1, the SSP position is defined only by the SSP offset value. Modify this bit only in channel reset mode or channel halt mode.

EOCCFG[2:0] Bits

These bits are used to select a frame format and a transmission/reception direction when the error occurrence counter counts CAN bus errors.

Modify these bits only in channel reset mode or channel halt mode.

24.3.2.7 RCFDCnCFDCmFDCTR — Channel CAN FD Control Register (m = 0 to 7)

Access: RCFDCnCFDCmFDCTR register can be read or written in 32-bit units
RCFDCnCFDCmFDCTRL register can be read or written in 16-bit units
RCFDCnCFDCmFDCTRLL register can be read or written in 8-bit units

Address: RCFDCnCFDCmFDCTR: <RCFDCn_base> + 0708_H + (20_H × m)
RCFDCnCFDCmFDCTRL: <RCFDCn_base> + 0708_H + (20_H × m)
RCFDCnCFDCmFDCTRLL: <RCFDCn_base> + 0708_H + (20_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.39 RCFDCnCFDCmFDCTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
1	SOCCLR	Successful Occurrence Counter Clear Setting the SOCCLR bit to 1 clears the successful occurrence counter. This bit is always read as 0.
0	EOCCLR	Error Occurrence Counter Clear Setting the EOCCLR bit to 1 clears the error occurrence counter. This bit is always read as 0.

SOCCLR Bit

Setting this bit to 1 clears the successful occurrence counter (SOC[7:0] bits in the RCFDCnCFDCmFDSTS register). This bit is automatically cleared to 0.

EOCCLR Bit

Setting this bit to 1 clears the error occurrence counter (EOC[7:0] bits in the RCFDCnCFDCmFDSTS register). This bit is automatically cleared to 0.

24.3.2.8 RCFDCnCFDCmFDSTS — Channel CAN FD Status Register (m = 0 to 7)

Access: RCFDCnCFDCmFDSTS register can be read or written in 32-bit units
RCFDCnCFDCmFDSTSL register can be read or written in 16-bit units
RCFDCnCFDCmFDSTSH register is a read-only register that can be read in 16-bit units
RCFDCnCFDCmFDSTSLL, RCFDCnCFDCmFDSTSLH registers can be read or written in 8-bit units
RCFDCnCFDCmFDSTSHL, RCFDCnCFDCmFDSTSHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCmFDSTS: <RCFDCn_base> + 070C$_H$ + (20$_H$ × m)
RCFDCnCFDCmFDSTSL: <RCFDCn_base> + 070C$_H$ + (20$_H$ × m),
RCFDCnCFDCmFDSTSH: <RCFDCn_base> + 070E$_H$ + (20$_H$ × m)
RCFDCnCFDCmFDSTSLL: <RCFDCn_base> + 070C$_H$ + (20$_H$ × m),
RCFDCnCFDCmFDSTSLH: <RCFDCn_base> + 070D$_H$ + (20$_H$ × m),
RCFDCnCFDCmFDSTSHL: <RCFDCn_base> + 070E$_H$ + (20$_H$ × m),
RCFDCnCFDCmFDSTSHH: <RCFDCn_base> + 070F$_H$ + (20$_H$ × m)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOC[7:0]								EOC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SOCO	EOCO	TDCVF	TDCR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W^{*1}	R/W^{*1}	R/W^{*1}	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.40 RCFDCnCFDCmFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	SOC[7:0]	Successful Occurrence Counter The successful occurrence counter value can be read.
23 to 16	EOC[7:0]	Error Occurrence Counter The error occurrence counter value can be read.
15 to 10	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
9	SOCO	Successful Occurrence Counter Overflow Flag 0: The successful occurrence counter does not overflow. 1: The successful occurrence counter has overflowed.
8	EOCO	Error Occurrence Counter Overflow Flag 0: The error occurrence counter does not overflow. 1: The error occurrence counter has overflowed.
7	TDCVF	Transmitter Delay Compensation Violation Flag 0: No transmitter delay compensation violation is present. 1: A transmitter delay compensation violation is present.
6 to 0	TDCR[6:0]	Transmitter Delay Compensation Result Status The transmitter delay compensation result can be read.

SOC[7:0] Bits

These bits show the successful occurrence counter value. The successful occurrence counter is incremented upon completion of message reception or transmission without an error. This counter stops counting when it reaches FF_H. In loopback mode, this counter is incremented twice.

These bits are cleared to 0 by writing 1 to the SOCCLR bit in the RCFDCnCFDCmFDCTR register. These bits are set to 0 in channel reset mode.

EOC[7:0] Bits

These bits show the error occurrence counter value. The error occurrence counter is incremented each time an error occurs according to the condition specified by the EOCCFG[2:0] bits in the RCFDCnCFDCmFDCFG register. This counter stops counting when it reaches FF_H.

These bits are cleared to 0 by writing 1 to the EOCCLR bit in the RCFDCnCFDCmFDCTR register. These bits are set to 0 in channel reset mode.

SOCO Flag

This bit indicates that successful occurrence counter overflow has occurred.

This flag is set to 1 when message reception or transmission is completed while the SOC[7:0] value has reached FF_H. This flag is set to 0 in channel reset mode.

EOCO Flag

This bit indicates that error occurrence counter overflow has occurred.

This flag is set to 1 when a CAN bus error is detected under the condition specified by the EOCCFG[2:0] bits in the RCFDCnCFDCmFDCFG register when the EOC[7:0] value has reached FF_H. This flag is set to 0 in channel reset mode.

TDCVF Flag

This bit indicates violation of transmitter delay compensation.

The transmit data is compared with the reception CAN bus level delayed due to the transceiver’s loop delay. This delay changes due to physical factors such as temperature. Because the TDCR[6:0] flags are updated for each message, temporary maximum delay cannot be confirmed.

This bit is set to 1 when the transmitter delay compensation exceeds the maximum compensation 3 CANm bit times - 2 fCAN (CANm bit time is the value of data bit rate). This flag is set to 0 in channel reset mode.

TDCR[6:0] Flags

These bits indicate the transmitter delay compensation result as a multiple of CAN clock frequency (fCAN).

This result depends on the settings of the TDCOC bit and TDCO[6:0] bits in the RCFDCnCFDCmFDCFG register.

These flags are updated at a falling edge between the FDF bit and res bit when the TDCE bit in the RCFDCnCFDCmFDCFG register is set to 1(transmitter delay compensation enable) and also the TDCOC bit in the RCFDCnCFDCmFDCFG register is set to 0(measurement and offset).

This flag is set to 0 in channel reset mode.

24.3.2.9 RCFDCnCFDCmFDCRC — Channel CAN FD CRC Register (m = 0 to 7)

Access: RCFDCnCFDCmFDCRC register is a read-only register that can be read in 32-bit units

RCFDCnCFDCmFDCRCL, RCFDCnCFDCmFDCRCH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDCmFDCRCLL, RCFDCnCFDCmFDCRCLH, RCFDCnCFDCmFDCRCHL, RCFDCnCFDCmFDCRCHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCmFDCRC: <RCFDCn_base> + 0710_H + (20_H × m)

RCFDCnCFDCmFDCRCL: <RCFDCn_base> + 0710_H + (20_H × m),

RCFDCnCFDCmFDCRCH: <RCFDCn_base> + 0712_H + (20_H × m)

RCFDCnCFDCmFDCRCLL: <RCFDCn_base> + 0710_H + (20_H × m),

RCFDCnCFDCmFDCRCLH: <RCFDCn_base> + 0711_H + (20_H × m),

RCFDCnCFDCmFDCRCHL: <RCFDCn_base> + 0712_H + (20_H × m),

RCFDCnCFDCmFDCRCHH: <RCFDCn_base> + 0713_H + (20_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRCREG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.41 RCFDCnCFDCmFDCRC Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	These bits are read as the value after reset.
27 to 24	SCNT[3:0]	Stuff count bit Indicate a value of the stuff count in a CAN FD frame. Bits 25 to 27 indicates the Gray-coded value of the stuff bit count modulo 8 in the transmitted/received frames. Bit 24 indicates an even parity value of bits 25 to 27.
23 to 21	Reserved	These bits are read as the value after reset.
20 to 0	CRCREG[20:0]	CRC Calculation Data (CRC Length:17 Bit or 21 Bit) These bits show the CRC value calculated based on the transmit message or receive message. When the CRC length is 17 bits, bits b20 to b17 are read as 0.

SCNT[3:0] Flags

When the CTME bit in the RCFDCnCFDCmCTR register is set to 1 (communication test mode enabled), a stuff count bit value of the CAN FD frame can be read if a message transmitted/received is a CAN FD frame. When the CTME bit is 0 (communication test mode disabled), this flag is always read as 0. These flags are updated at the first bit in the CRC field of the CAN FD frame. These bits are cleared to 0 in channel reset mode.

CRCREG[20:0] Flags

When the CTME bit in the RCFDCnCFDCmCTR register is 1 (communication test mode enabled), if transmit or receive message is a CAN FD frame (CRC length = 17 or 21 bits), these flags are updated and the CRC value calculated based on the message can be read. When the CRC length of the message is 17 bits, bits b20 to b17 are always read as 0. When a classical CAN frame is transmitted or received the CRCREG[14:0] value in the RCFDCnCFDCmERFL register is updated. When the CTME bit is 0 (communication test mode disabled), these bits are always read as 0.

These bits are cleared to 0 in channel reset mode.

24.3.3 Details of Global-related Registers

24.3.3.1 RCFDCnCFDGCFG — Global Configuration Register

Access: RCFDCnCFDGCFG register can be read or written in 32-bit units

RCFDCnCFDGCFGL, RCFDCnCFDGCFGH registers can be read or written in 16-bit units

RCFDCnCFDGCFGLL, RCFDCnCFDGCFGLH, RCFDCnCFDGCFGHL, RCFDCnCFDGCFGHH registers can be read or written in 8-bit units

Address: RCFDCnCFDGCFG: <RCFDCn_base> + 0084_H

RCFDCnCFDGCFGL: <RCFDCn_base> + 0084_H,

RCFDCnCFDGCFGH: <RCFDCn_base> + 0086_H

RCFDCnCFDGCFGLL: <RCFDCn_base> + 0084_H,

RCFDCnCFDGCFGLH: <RCFDCn_base> + 0085_H,

RCFDCnCFDGCFGHL: <RCFDCn_base> + 0086_H,

RCFDCnCFDGCFGHH: <RCFDCn_base> + 0087_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.42 RCFDCnCFDGCFG Register Contents

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 nominal bit time clock 0 0 1: Channel 1 nominal bit time clock 0 1 0: Channel 2 nominal bit time clock 0 1 1: Channel 3 nominal bit time clock 1 0 0: Channel 4 nominal bit time clock 1 0 1: Channel 5 nominal bit time clock 1 1 0: Channel 6 nominal bit time clock 1 1 1: Channel 7 nominal bit time clock
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Nominal bit time clock

Table 24.42 RCFDCnCFDGCFG Register Contents

Bit Position	Bit Name	Function
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	CMPOC	Payload Overflow Mode Select 0: No message is stored. 1: Messages are stored and payloads exceeding the buffer size are discarded.
4	DCS	CAN Clock Source Select[*2] 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see **Table 24.16, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM**.

Modify the RCFDCnCFDGCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 24.7.3.1, Interval Transmission Function**.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the nominal bit time clock that will be the clock source of the timestamp counter. However, do not select the channel that handles the CAN FD frames.

TSSS Bit

This bit is used to select a clock source of the timestamp counter. Select pclk if there is no channel that handles only classical CAN frames.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

CMPOC Bit

This bit is used to select operation in case the payload length of received message exceeds the payload storage size of the storage buffer.

When this bit is 0, the received message in which the payload overflows is not stored in the buffer.

When this bit is 1, the received message in which the payload overflows is stored in the buffer, and depending on the DRE bit the received DLC value or the DLC value of the receive rule is stored in the buffer. At this time, payloads exceeding the buffer's payload storage size are discarded.

The buffer's payload storage size is set by the following bits.

- Receive buffer: RMPLS[2:0] bits in the RCFDCnCFDRMNB register
- Receive FIFO buffer: RFPLS[2:0] bits in the RCFDCnCFDRFCCx register
- Transmit/receive FIFO buffer: CFPLS[2:0] bits in the RCFDCnCFDCFCCk register

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN). For the CAN clock frequency settings, see **Table 24.16, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM**.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte that exceeds the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RCFDCnCFDGAFLP0_j register to 0000_B before clearing the DCE bit in the RCFDCnCFDGCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number has the highest priority.

When using the transmit queue, this bit should be set to 0.

24.3.3.2 RCFDCnCFDGCTR — Global Control Register

Access: RCFDCnCFDGCTR register can be read or written in 32-bit units

RCFDCnCFDGCTRL, RCFDCnCFDGCTRH registers can be read or written in 16-bit units

RCFDCnCFDGCTRLL, RCFDCnCFDGCTRLH, RCFDCnCFDGCTRHL registers can be read or written in 8-bit units

Address: RCFDCnCFDGCTR: <RCFDCn_base> + 0088$_H$

RCFDCnCFDGCTRL: <RCFDCn_base> + 0088$_H$,

RCFDCnCFDGCTRH: <RCFDCn_base> + 008A$_H$

RCFDCnCFDGCTRLL: <RCFDCn_base> + 0088$_H$,

RCFDCnCFDGCTRLH: <RCFDCn_base> + 0089$_H$,

RCFDCnCFDGCTRHL: <RCFDCn_base> + 008A$_H$

Value after reset: 0000 0005$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 24.43 RCFDCnCFDGCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11	CMPOFIE	Payload Overflow Interrupt Enable 0: A payload overflow interrupt is disabled. 1: A payload overflow interrupt is enabled.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode

Table 24.43 RCFDCnCFDGCTR Register Contents

Bit Position	Bit Name	Function
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RCFDCnCFDGTSC register is cleared to 0000_H.

CMPOFIE Bit

When the CMPOF flag in the RCFDCnCFDGERFL register is set to 1 after the CMPOFIE bit is set to 1, an interrupt request occurs. Modify this bit only in global reset mode.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RCFDCnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RCFDCnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RCFDCnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RS-CANFD module into global stop mode.

Clearing this bit to 0 makes the RS-CANFD module leave from global stop mode. This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see **Section 24.5.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module in global stop mode.

24.3.3.3 RCFDCnCFDGSTS — Global Status Register

Access: RCFDCnCFDGSTS register is a read-only register that can be read in 32-bit units
RCFDCnCFDGSTSL register is a read-only register that can be read in 16-bit units
RCFDCnCFDGSTSLL register is a read-only register that can be read in 8-bit units

Address: RCFDCnCFDGSTS: <RCFDCn_base> + 008C$_H$
RCFDCnCFDGSTSL: <RCFDCn_base> + 008C$_H$
RCFDCnCFDGSTSLL: <RCFDCn_base> + 008C$_H$

Value after reset: 0000 000D$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.44 RCFDCnCFDGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

24.3.3.4 RCFDCnCFDGERFL — Global Error Flag Register

Access: RCFDCnCFDGERFL register can be read or written in 32-bit units
RCFDCnCFDGERFLL, RCFDCnCFDGERFLH registers can be read or written in 16-bit units
RCFDCnCFDGERFLLL, RCFDCnCFDGERFLHL registers can be read or written in 8-bit units

Address: RCFDCnCFDGERFL: <RCFDCn_base> + 0090$_H$
RCFDCnCFDGERFLL: <RCFDCn_base> + 0090$_H$,
RCFDCnCFDGERFLH: <RCFDCn_base> + 0092$_H$
RCFDCnCFDGERFLLL: <RCFDCn_base> + 0090$_H$,
RCFDCnCFDGERFLHL: <RCFDCn_base> + 0092$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EEF7	EEF6	EEF5	EEF4	EEF3	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPOF	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.45 RCFDCnCFDGERFL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
23	EEF7	ECC Error Flag for Channel 7 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
22	EEF6	ECC Error Flag for Channel 6 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
21	EEF5	ECC Error Flag for Channel 5 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
20	EEF4	ECC Error Flag for Channel 4 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
19	EEF3	ECC Error Flag for Channel 3 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
18	EEF2	ECC Error Flag for Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
17	EEF1	ECC Error Flag for Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority

Table 24.45 RCFDCnCFDGERFL Register Contents

Bit Position	Bit Name	Function
16	EEF0	ECC Error Flag for Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
15 to 4	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
3	CMPOF	Payload Overflow Flag 0: No payload overflow has occurred. 1: A payload overflow has occurred.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RCFDCnCFDGERFL register are cleared to 0 in global reset mode.

EEFm Flag

When a 2-bit ECC error is detected during the transmission priority determination of channel m (m = 0 to 7), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

CMPOF Flag

When a payload overflow occurs in any of channel m (m = 0 to 7), the CMPOF flag is set to 1. This flag can be cleared to 0 by writing 0 to this bit by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RCFDCnCFDTHLSTSm register (m = 0 to 7) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RCFDCnCFDRFSTSx register (x = 0 to 7) or the CFMLT flags in the RCFDCnCFDCFSTSk register (k = 0 to 17) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

24.3.3.5 RCFDCnCFDGTSC — Global Timestamp Counter Register

Access:	RCFDCnCFDGTSC register is a read-only register that can be read in 32-bit units. RCFDCnCFDGTSCL register is a read-only register that can be read in 16-bit units.
Address:	RCFDCnCFDGTSC: <RCFDCn_base> + 0094_H RCFDCnCFDGTSCL: <RCFDCn_base> + 0094_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.46 RCFDCnCFDGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. Furthermore, the TS[15:0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RCFDCnCFDGCFG register is 0 (pclk):

  The timestamp counter starts counting when the RS-CANFD module has transitioned to global operating mode.

  This counter stops counting when the RS-CANFD module has transitioned to global stop mode or global test mode.

- When the TSSS bit is 1 (CANm nominal bit time clock):

  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.

  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

24.3.3.6 RCFDCnCFDGTINTSTS0 — Global TX Interrupt Status Register 0

Access: RCFDCnCFDGTINTSTS0 register is a read-only register that can be read in 32-bit units

RCFDCnCFDGTINTSTS0L, RCFDCnCFDGTINTSTS0H registers are read-only registers that can be read in 16-bit units

RCFDCnCFDGTINTSTS0LL, RCFDCnCFDGTINTSTS0LH, RCFDCnCFDGTINTSTS0HL, RCFDCnCFDGTINTSTS0HH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDGTINTSTS0: <RCFDCn_base> + 0610_H
RCFDCnCFDGTINTSTS0L: <RCFDCn_base> + 0610_H,
RCFDCnCFDGTINTSTS0H: <RCFDCn_base> + 0612_H
RCFDCnCFDGTINTSTS0LL: <RCFDCn_base> + 0610_H,
RCFDCnCFDGTINTSTS0LH: <RCFDCn_base> + 0611_H,
RCFDCnCFDGTINTSTS0HL: <RCFDCn_base> + 0612_H,
RCFDCnCFDGTINTSTS0HH: <RCFDCn_base> + 0613_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 24.47 RCFDCnCFDGTINTSTS0 Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	These bits are read as the value after reset.
28	THIF3	Channel 3 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF3	Channel 3 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF3	Channel 3 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF3	Channel 3 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested
24	TSIF3	Channel 3 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
23 to 21	Reserved	These bits are read as the value after reset.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.

Table 24.47 RCFDCnCFDGTINTSTS0 Register Contents

Bit Position	Bit Name	Function
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	These bits are read as the value after reset.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	These bits are read as the value after reset.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RCFDCnCFDTMIECm register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the corresponding RCFDCnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RCFDCnCFDCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RCFDCnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQTXIE bit in the RCFDCnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQTXIF bit in the RCFDCnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQTXIF bit (transmit queue interrupt request) in the RCFDCnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQTXIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RCFDCnCFDCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RCFDCnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RCFDCnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RCFDCnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RCFDCnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

24.3.3.7 RCFDCnCFDGTINTSTS1 — Global TX Interrupt Status Register 1

Access: RCFDCnCFDGTINTSTS1 register is a read-only register that can be read in 32-bit units

RCFDCnCFDGTINTSTS1L, RCFDCnCFDGTINTSTS1H registers are read-only registers that can be read in 16-bit units

RCFDCnCFDGTINTSTS1LL, RCFDCnCFDGTINTSTS1LH, RCFDCnCFDGTINTSTS1HL, RCFDCnCFDGTINTSTS1HH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDGTINTSTS1: <RCFDCn_base> + 0614$_H$
RCFDCnCFDGTINTSTS1L: <RCFDCn_base> + 0614$_H$,
RCFDCnCFDGTINTSTS1H: <RCFDCn_base> + 0616$_H$
RCFDCnCFDGTINTSTS1LL: <RCFDCn_base> + 0614$_H$,
RCFDCnCFDGTINTSTS1LH: <RCFDCn_base> + 0615$_H$,
RCFDCnCFDGTINTSTS1HL: <RCFDCn_base> + 0616$_H$,
RCFDCnCFDGTINTSTS1HH: <RCFDCn_base> + 0617$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF7	CFTIF7	TQIF7	TAIF7	TSIF7	—	—	—	THIF6	CFTIF6	TQIF6	TAIF6	TSIF6
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF5	CFTIF5	TQIF5	TAIF5	TSIF5	—	—	—	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 24.48 RCFDCnCFDGTINTSTS1 Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	These bits are read as the value after reset.
28	THIF7	Channel 7 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF7	Channel 7 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF7	Channel 7 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF7	Channel 7 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
24	TSIF7	Channel 7 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested
23 to 21	Reserved	These bits are read as the value after reset.
20	THIF6	Channel 6 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.

Table 24.48 RCFDCnCFDGTINTSTS1 Register Contents

Bit Position	Bit Name	Function
19	CFTIF6	Channel 6 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF6	Channel 6 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF6	Channel 6 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
16	TSIF6	Channel 6 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested
15 to 13	Reserved	These bits are read as the value after reset.
12	THIF5	Channel 5 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF5	Channel 5 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF5	Channel 5 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF5	Channel 5 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF5	Channel 5 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested
7 to 5	Reserved	These bits are read as the value after reset.
4	THIF4	Channel 4 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF4	Channel 4 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF4	Channel 4 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF4	Channel 4 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF4	Channel 4 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RCFDCnCFDTMIECm register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the corresponding RCFDCnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RCFDCnCFDCmCTR register is set to 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RCFDCnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQTXIE bit in the RCFDCnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQTXIF bit in the RCFDCnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQTXIF bit (transmit queue interrupt request) in the RCFDCnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. Clearing the TXQTXIE bit to 0 also clears this flag to 0.

CFTIFm Bits

When the CFTXIE bit in the RCFDCnCFDCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RCFDCnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RCFDCnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RCFDCnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RCFDCnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

24.3.3.8 RCFDCnCFDGFDCFG — Global FD Configuration Register

Access: RCFDCnCFDGFDCFG register can be read or written in 32-bit units
RCFDCnCFDGFDCFGL register can be read or written in 16-bit units
RCFDCnCFDGFDCFGLL, RCFDCnCFDGFDCFGLH registers can be read or written in 8-bit units

Address: RCFDCnCFDGFDCFG: <RCFDCn_base> + 0624_H
RCFDCnCFDGFDCFGL: <RCFDCn_base> + 0624_H
RCFDCnCFDGFDCFGLL: <RCFDCn_base> + 0624_H,
RCFDCnCFDGFDCFGLH: <RCFDCn_base> + 0625_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 24.49 RCFDCnCFDGFDCFG register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9, 8	TSCCFG[1:0]	Time-stamp capture setting bit b9 b8 0 0: Captured at a sample point in the SOF bit. 0 1: Captured when a valid frame has been transmitted/received. 1 0: Captured at a sample point of the res bit.*1 1 1: Setting prohibited.
7 to 1	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
0	RPED	Protocol exception event detection disabled bit 0: Protocol exception event detection is enabled 1: Protocol exception event detection is disabled

Note 1. When a classical CAN frame is transmitted/received, a time-stamp value will be captured at the sample point in the SOF bit.

TSCCFG bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

RPED bit

Setting this bit to 1 disables the protocol exception event detection. When a protocol exception event is detected while this bit is set to 1, the event is regarded as a form error and an error frame will be output. Modify this bit only in global reset mode.

24.3.4 Details of Receive Rule-related Registers

24.3.4.1 RCFDCnCFDGAFLECTR — Receive Rule Entry Control Register

Access:	RCFDCnCFDGAFLECTR register can be read or written in 32-bit units RCFDCnCFDGAFLECTRL register can be read or written in 16-bit units RCFDCnCFDGAFLECTRLL, RCFDCnCFDGAFLECTRLH registers can be read or written in 8-bit units
Address:	RCFDCnCFDGAFLECTR: <RCFDCn_base> + 0098_H RCFDCnCFDGAFLECTRL: <RCFDCn_base> + 0098_H RCFDCnCFDGAFLECTRLL: <RCFDCn_base> + 0098_H, RCFDCnCFDGAFLECTRLH: <RCFDCn_base> + 0099_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	AFLPN[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.50 RCFDCnCFDGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7, 6	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
5 to 0	AFLPN[5:0]	Receive Rule Table Page Number Configuration A page number can be selected in the range of page 0 (000000_B) to page 63 (111111_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[5:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 000000_B to 111111_B.

For details about the receive rule table, see **Section 24.6.1, Data Processing Using the Receive Rule Table**.

24.3.4.2 RCFDCnCFDGAFLCFG0 — Receive Rule Configuration Register 0

Access: RCFDCnCFDGAFLCFG0 register can be read or written in 32-bit units

RCFDCnCFDGAFLCFG0L, RCFDCnCFDGAFLCFG0H registers can be read or written in 16-bit units

RCFDCnCFDGAFLCFG0LL, RCFDCnCFDGAFLCFG0LH, RCFDCnCFDGAFLCFG0HL, RCFDCnCFDGAFLCFG0HH registers can be read or written in 8-bit units

Address: RCFDCnCFDGAFLCFG0: <RCFDCn_base> + 009C$_H$
RCFDCnCFDGAFLCFG0L: <RCFDCn_base> + 009C$_H$,
RCFDCnCFDGAFLCFG0H: <RCFDCn_base> + 009E$_H$
RCFDCnCFDGAFLCFG0LL: <RCFDCn_base> + 009C$_H$,
RCFDCnCFDGAFLCFG0LH: <RCFDCn_base> + 009D$_H$,
RCFDCnCFDGAFLCFG0HL: <RCFDCn_base> + 009E$_H$,
RCFDCnCFDGAFLCFG0HH: <RCFDCn_base> + 009F$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								RNC3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.51 RCFDCnCFDGAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	RNC3[7:0]	Number of Rules for Channel 3 Set the number of receive rules exclusively used for channel 3.

Modify the RCFDCnCFDGAFLCFG0 register only in global reset mode.

Up to 128 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 255.
- The total number of rules allocated to each channel does not exceed the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table. Set these bits to a value within the range of 00$_H$ to FF$_H$.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

RNC3[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 3 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

24.3.4.3 RCFDCnCFDGAFLCFG1 — Receive Rule Configuration Register 1

Access: RCFDCnCFDGAFLCFG1 register can be read or written in 32-bit units

RCFDCnCFDGAFLCFG1L, RCFDCnCFDGAFLCFG1H registers can be read or written in 16-bit units

RCFDCnCFDGAFLCFG1LL, RCFDCnCFDGAFLCFG1LH, RCFDCnCFDGAFLCFG1HL, RCFDCnCFDGAFLCFG1HH registers can be read or written in 8-bit units

Address: RCFDCnCFDGAFLCFG1: <RCFDCn_base> + 00A0$_H$

RCFDCnCFDGAFLCFG1L: <RCFDCn_base> + 00A0$_H$,

RCFDCnCFDGAFLCFG1H: <RCFDCn_base> + 00A2$_H$

RCFDCnCFDGAFLCFG1LL: <RCFDCn_base> + 00A0$_H$,

RCFDCnCFDGAFLCFG1LH: <RCFDCn_base> + 00A1$_H$,

RCFDCnCFDGAFLCFG1HL: <RCFDCn_base> + 00A2$_H$,

RCFDCnCFDGAFLCFG1HH: <RCFDCn_base> + 00A3$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC4[7:0]								RNC5[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC6[7:0]								RNC7[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.52 RCFDCnCFDGAFLCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC4[7:0]	Number of Rules for Channel 4 Set the number of receive rules exclusively used for channel 4.
23 to 16	RNC5[7:0]	Number of Rules for Channel 5 Set the number of receive rules exclusively used for channel 5.
15 to 8	RNC6[7:0]	Number of Rules for Channel 6 Set the number of receive rules exclusively used for channel 6.
7 to 0	RNC7[7:0]	Number of Rules for Channel 7 Set the number of receive rules exclusively used for channel 7.

Modify the RCFDCnCFDGAFLCFG1 register only in global reset mode.

Up to 128 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 255.
- The total number of rules allocated to each channel does not exceed the number of rules that can be registered for the entire unit.

RNC4[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 4 receive rule table. Set these bits to a value within the range of 00$_H$ to FF$_H$.

RNC5[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 5 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

RNC6[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 6 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

RNC7[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 7 receive rule table. Set these bits to a value within the range of 00_H to FF_H.

24.3.4.4 RCFDCnCFDGAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RCFDCnCFDGAFLIDj register can be read or written in 32-bit units

RCFDCnCFDGAFLIDjL, RCFDCnCFDGAFLIDjH registers can be read or written in 16-bit units

RCFDCnCFDGAFLIDjLL, RCFDCnCFDGAFLIDjLH, RCFDCnCFDGAFLIDjHL, RCFDCnCFDGAFLIDjHH registers can be read or written in 8-bit units

Address: RCFDCnCFDGAFLIDj: <RCFDCn_base> + 1000_H + (10_H × j)

RCFDCnCFDGAFLIDjL: <RCFDCn_base> + 1000_H + (10_H × j),

RCFDCnCFDGAFLIDjH: <RCFDCn_base> + 1002_H + (10_H × j)

RCFDCnCFDGAFLIDjLL: <RCFDCn_base> + 1000_H + (10_H × j),

RCFDCnCFDGAFLIDjLH: <RCFDCn_base> + 1001_H + (10_H × j),

RCFDCnCFDGAFLIDjHL: <RCFDCn_base> + 1002_H + (10_H × j),

RCFDCnCFDGAFLIDjHH: <RCFDCn_base> + 1003_H + (10_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.53 RCFDCnCFDGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RCFDCnCFDGAFLIDj register when the AFLDAE bit in the RCFDCnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

24.3.4.5 RCFDCnCFDGAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RCFDCnCFDGAFLMj register can be read or written in 32-bit units

RCFDCnCFDGAFLMjL, RCFDCnCFDGAFLMjH registers can be read or written in 16-bit units

RCFDCnCFDGAFLMjLL, RCFDCnCFDGAFLMjLH, RCFDCnCFDGAFLMjHL, RCFDCnCFDGAFLMjHH registers can be read or written in 8-bit units

Address: RCFDCnCFDGAFLMj: <RCFDCn_base> + 1004_H + (10_H × j)

RCFDCnCFDGAFLMjL: <RCFDCn_base> + 1004_H + (10_H × j),

RCFDCnCFDGAFLMjH: <RCFDCn_base> + 1006_H + (10_H × j)

RCFDCnCFDGAFLMjLL: <RCFDCn_base> + 1004_H + (10_H × j),

RCFDCnCFDGAFLMjLH: <RCFDCn_base> + 1005_H + (10_H × j),

RCFDCnCFDGAFLMjHL: <RCFDCn_base> + 1006_H + (10_H × j),

RCFDCnCFDGAFLMjHH: <RCFDCn_base> + 1007_H + (10_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.54 RCFDCnCFDGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared.
29	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RCFDCnCFDGAFLMj register when the AFLDAE bit in the RCFDCnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RCFDCnCFDGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set all the GAFLIDM[28:0] bits to 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

24.3.4.6 RCFDCnCFDGAFLP0_j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RCFDCnCFDGAFLP0_j register can be read or written in 32-bit units

RCFDCnCFDGAFLP0_jL, RCFDCnCFDGAFLP0_jH registers can be read or written in 16-bit units

RCFDCnCFDGAFLP0_jLL, RCFDCnCFDGAFLP0_jLH, RCFDCnCFDGAFLP0_jHL, RCFDCnCFDGAFLP0_jHH registers can be read or written in 8-bit units

Address: RCFDCnCFDGAFLP0_j: <RCFDCn_base> + 1008_H + $(10_H \times j)$

RCFDCnCFDGAFLP0_jL: <RCFDCn_base> + 1008_H + $(10_H \times j)$,

RCFDCnCFDGAFLP0_jH: <RCFDCn_base> + $100A_H$ + $(10_H \times j)$

RCFDCnCFDGAFLP0_jLL: <RCFDCn_base> + 1008_H + $(10_H \times j)$,

RCFDCnCFDGAFLP0_jLH: <RCFDCn_base> + 1009_H + $(10_H \times j)$,

RCFDCnCFDGAFLP0_jHL: <RCFDCn_base> + $100A_H$ + $(10_H \times j)$,

RCFDCnCFDGAFLP0_jHH: <RCFDCn_base> + $100B_H$ + $(10_H \times j)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	GAFLDLC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.55 RCFDCnCFDGAFLP0_j Register Contents

Bit Position	Bit Name	Function
31 to 16	GAFLPTR[15:0]	Receive Rule Label Set the 16-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 4	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 24.55 RCFDCnCFDGAFLP0_j Register Contents

Bit Position	Bit Name	Function
3 to 0	GAFLDLC[3:0]	Receive Rule DLC

b3	b2	b1	b0	Classical CAN Frame	CAN FD Frame
0	0	0	0	DLC check is disabled	
0	0	0	1	1 data byte	
0	0	1	0	2 data bytes	
0	0	1	1	3 data bytes	
0	1	0	0	4 data bytes	
0	1	0	1	5 data bytes	
0	1	1	0	6 data bytes	
0	1	1	1	7 data bytes	
1	0	0	0	8 data bytes	
1	0	0	1	8 data bytes	12 data bytes
1	0	1	0		16 data bytes
1	0	1	1		20 data bytes
1	1	0	0		24 data bytes
1	1	0	1		32 data bytes
1	1	1	0		48 data bytes
1	1	1	1		64 data bytes

Modify the RCFDCnCFDGAFLP0_j register when the AFLDAE bit in the RCFDCnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLPTR[15:0] Bits

These bits are used to set a 16-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the setting value by the NRXMB[7:0] bits in the RCFDCnCFDRMNB register.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

24.3.4.7 RCFDCnCFDGAFLP1_j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RCFDCnCFDGAFLP1_j register can be read or written in 32-bit units

RCFDCnCFDGAFLP1_jL, RCFDCnCFDGAFLP1_jH registers can be read or written in 16-bit units

RCFDCnCFDGAFLP1_jLL, RCFDCnCFDGAFLP1_jLH, RCFDCnCFDGAFLP1_jHL, RCFDCnCFDGAFLP1_jHH registers can be read or written in 8-bit units

Address:
RCFDCnCFDGAFLP1_j: <RCFDCn_base> + $100C_H$ + (10_H × j)
RCFDCnCFDGAFLP1_jL: <RCFDCn_base> +$100C_H$ + (10_H × j),
RCFDCnCFDGAFLP1_jH: <RCFDCn_base> + $100E_H$ + (10_H × j)
RCFDCnCFDGAFLP1_jLL: <RCFDCn_base> + $100C_H$ + (10_H × j),
RCFDCnCFDGAFLP1_jLH: <RCFDCn_base> + $100D_H$ + (10_H × j),
RCFDCnCFDGAFLP1_jHL: <RCFDCn_base> + $100E_H$ + (10_H × j),
RCFDCnCFDGAFLP1_jHH: <RCFDCn_base> + $100F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLFDP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.56 RCFDCnCFDGAFLP1_j Register Contents

Bit Position	Bit Name	Function
31 to 8	GAFLFDP[31:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RCFDCnCFDGAFLP1_j register when the AFLDAE bit in the RCFDCnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP[31:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers can be selected. However, when the GAFLRMV bit in the RCFDCnCFDGAFLP0_j register is set to 1 (messages are stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RCFDCnCFDCFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) can be selected.

24.3.5 Details of Receive Buffer-related Registers

24.3.5.1 RCFDCnCFDRMNB — Receive Buffer Number Register

Access: RCFDCnCFDRMNB register can be read or written in 32-bit units

RCFDCnCFDRMNBL register can be read or written in 16-bit units

RCFDCnCFDRMNBLL, RCFDCnCFDRMNBLH registers can be read or written in 8-bit units

Address: RCFDCnCFDRMNB: <RCFDCn_base> + 00A4$_\text{H}$

RCFDCnCFDRMNBL: <RCFDCn_base> + 00A4$_\text{H}$

RCFDCnCFDRMNBLL: <RCFDCn_base> + 00A4$_\text{H}$,

RCFDCnCFDRMNBLH: <RCFDCn_base> + 00A5$_\text{H}$

Value after reset: 0000 0000$_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RMPLS[2:0]			NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.57 RCFDCnCFDRMNB Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	RMPLS[2:0]	Receive Buffer Payload Storage Size Select b10 b9 b8 0 0 0: 8byte 0 0 1: 12byte 0 1 0: 16byte 0 1 1: 20byte 1 0 0: 24byte 1 0 1: 32byte 1 1 0: 48byte 1 1 1: 64byte
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 128.

Modify the RCFDCnCFDRMNB register only in global reset mode.

RMPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive buffer.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is 16 × (number of channels).

Setting all of these bits to 0 makes receive buffers unavailable.

24.3.5.2 RCFDCnCFDRMNDy — Receive Buffer New Data Register (y = 0 to 3)

Access:	RCFDCnCFDRMNDy register can be read or written in 32-bit units RCFDCnCFDRMNDyL, RCFDCnCFDRMNDyH registers can be read or written in 16-bit units RCFDCnCFDRMNDyLL, RCFDCnCFDRMNDyLH, RCFDCnCFDRMNDyHL, RCFDCnCFDRMNDyHH registers can be read or written in 8-bit units
Address:	RCFDCnCFDRMNDy: <RCFDCn_base> + $00A8_H$ + (04_H × y) RCFDCnCFDRMNDyL: <RCFDCn_base> + $00A8_H$ + (04_H × y), RCFDCnCFDRMNDyH: <RCFDCn_base> + $00AA_H$ + (04_H × y) RCFDCnCFDRMNDyLL: <RCFDCn_base> + $00A8_H$ + (04_H × y), RCFDCnCFDRMNDyLH: <RCFDCn_base> + $00A9_H$ + (04_H × y), RCFDCnCFDRMNDyHL: <RCFDCn_base> + $00AA_H$ + (04_H × y), RCFDCnCFDRMNDyHH: <RCFDCn_base> + $00AB_H$ + (04_H × y)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2, 3))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2, 3))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.58 RCFDCnCFDRMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RCFDCnCFDRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 127)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. The message storing time depends on the storage payload size of the receive buffer. When the RMPLS[2:0] value in the RCFDCnCFDRMNB register is 000_B (8 bytes), the message storing time is 12 pclk clock cycles.

When the RMPLS[2:0] value is 111_B (64 bytes), the message storing time is 40 pclk clock cycles. (2 pclk clock cycles per 4 bytes of storage payload size). These flags are cleared to 0 in global reset mode.

24.3.5.3 RCFDCnCFDRMIDq — Receive Buffer ID Register (q = 0 to 127)

Access: RCFDCnCFDRMIDq register is a read-only register that can be read in 32-bit units

RCFDCnCFDRMIDqL, RCFDCnCFDRMIDqH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRMIDqLL, RCFDCnCFDRMIDqLH, RCFDCnCFDRMIDqHL, RCFDCnCFDRMIDqHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRMIDq: <RCFDCn_base> + 2000_H + (80_H × q)

RCFDCnCFDRMIDqL: <RCFDCn_base> + 2000_H + (80_H × q),

RCFDCnCFDRMIDqH: <RCFDCn_base> + 2002_H + (80_H × q)

RCFDCnCFDRMIDqLL: <RCFDCn_base> + 2000_H + (80_H × q),

RCFDCnCFDRMIDqLH: <RCFDCn_base> + 2001_H + (80_H × q),

RCFDCnCFDRMIDqHL: <RCFDCn_base> + 2002_H + (80_H × q),

RCFDCnCFDRMIDqHH: <RCFDCn_base> + 2003_H + (80_H × q)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.59 RCFDCnCFDRMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR/RRS ● When the received message is a classical CAN frame 0: Data frame 1: Remote frame ● When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	Reserved	This bit is read as the value after reset.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

24.3.5.4 RCFDCnCFDRMPTRq — Receive Buffer Pointer Register (q = 0 to 127)

Access: RCFDCnCFDRMPTRq register is a read-only register that can be read in 32-bit units

RCFDCnCFDRMPTRqL, RCFDCnCFDRMPTRqH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRMPTRqLL, RCFDCnCFDRMPTRqLH, RCFDCnCFDRMPTRqHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRMPTRq: <RCFDCn_base> + 2004$_H$ + (80$_H$ × q)
RCFDCnCFDRMPTRqL: <RCFDCn_base> + 2004$_H$ + (80$_H$ × q),
RCFDCnCFDRMPTRqH: <RCFDCn_base> + 2006$_H$ + (80$_H$ × q)
RCFDCnCFDRMPTRqLL: <RCFDCn_base> + 2004$_H$ + (80$_H$ × q),
RCFDCnCFDRMPTRqLH: <RCFDCn_base> + 2005$_H$ + (80$_H$ × q),
RCFDCnCFDRMPTRqHH: <RCFDCn_base> + 2007$_H$ + (80$_H$ × q)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.60 RCFDCnCFDRMPTRq Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>RMDLC[3:0]</td><td>Receive Buffer DLC Data
<table>
<tr><th>b31</th><th>b30</th><th>b29</th><th>b28</th><th>Classical CAN Frame</th><th>CAN FD Frame</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td></tr>
</table></td></tr>
<tr><td>27 to 16</td><td>Reserved</td><td>These bits are read as the value after reset.</td></tr>
<tr><td>15 to 0</td><td>RMTS[15:0]</td><td>Receive Buffer Timestamp Data
Timestamp value of the received message.</td></tr>
</table>

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer. The number of bytes of the payload to be stored in the receive buffer is determined by the RMPLS[2:0] bits in the RCFDCnCFDRMNB register.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

24.3.5.5 RCFDCnCFDRMFDSTSq — Receive Buffer CAN FD Status Register (q = 0 to 127)

Access: RCFDCnCFDRMFDSTSq register is a read-only register that can be read in 32-bit units

RCFDCnCFDRMFDSTSqL, RCFDCnCFDRMFDSTSqH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRMFDSTSqLL, RCFDCnCFDRMFDSTSqHL, RCFDCnCFDRMFDSTSqHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRMFDSTSq: <RCFDCn_base> + 2008_H + (80_H × q)
RCFDCnCFDRMFDSTSqL: <RCFDCn_base> + 2008_H + (80_H × q),
RCFDCnCFDRMFDSTSqH: <RCFDCn_base> + $200A_H$ + (80_H × q)
RCFDCnCFDRMFDSTSqLL: <RCFDCn_base> + 2008_H + (80_H × q),
RCFDCnCFDRMFDSTSqHL: <RCFDCn_base> + $200A_H$ + (80_H × q),
RCFDCnCFDRMFDSTSqHH: <RCFDCn_base> + $200B_H$ + (80_H × q)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.61 RCFDCnCFDRMFDSTSq Register Contents

Bit Position	Bit Name	Function
31 to 16	RMPTR[15:0]	Receive Buffer Label Data Label information of the received message.
15 to 3	Reserved	These bits are read as the value after reset.
2	RMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RMESI	ESI 0: Error active node 1: Error passive node

RMPTR[15:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive buffer.

RMBRS Bit

When the RMFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

RMESI Bit

When the RMFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

24.3.5.6 RCFDCnCFDRMDFb_q — Receive Buffer Data Field Register (b = 0 to 15, q = 0 to 127)

Access: RCFDCnCFDRMDFb_q register is a read-only register that can be read in 32-bit units

RCFDCnCFDRMDFb_qL, RCFDCnCFDRMDFb_qH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRMDFb_qLL, RCFDCnCFDRMDFb_qLH, RCFDCnCFDRMDFb_qHL, RCFDCnCFDRMDFb_qHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRMDFb_q: <RCFDCn_base> + 200C$_H$ + (04$_H$ × b) + (80$_H$ × q)
RCFDCnCFDRMDFb_qL: <RCFDCn_base> + 200C$_H$ + (04$_H$ × b) + (80$_H$ × q),
RCFDCnCFDRMDFb_qH: <RCFDCn_base> + 200E$_H$ + (04$_H$ × b) + (80$_H$ × q)
RCFDCnCFDRMDFb_qLL: <RCFDCn_base> + 200C$_H$ + (04$_H$ × b) + (80$_H$ × q),
RCFDCnCFDRMDFb_qLH: <RCFDCn_base> + 200D$_H$ + (04$_H$ × b) + (80$_H$ × q),
RCFDCnCFDRMDFb_qHL: <RCFDCn_base> + 200E$_H$ + (04$_H$ × b) + (80$_H$ × q),
RCFDCnCFDRMDFb_qHH: <RCFDCn_base> + 200F$_H$ + (04$_H$ × b) + (80$_H$ × q)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB4 × b + 3 [7:0]								RMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB4 × b + 1 [7:0]								RMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.62 RCFDCnCFDRMDFb_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB4 × b + 3 [7:0]	Receive Buffer Data Byte 4 × b + 3 Receive Buffer Data Byte 4 × b + 2 Receive Buffer Data Byte 4 × b + 1 Receive Buffer Data Byte 4 × b + 0 Data for a message stored in the receive buffer can be read.
23 to 16	RMDB4 × b + 2 [7:0]	
15 to 8	RMDB4 × b + 1 [7:0]	
7 to 0	RMDB4 × b + 0 [7:0]	

When the RMDLC[3:0] value in the RCFDCnCFDRMPTRq register is smaller than the payload storage size of the receive buffer, data bytes for which no data is set are read as 00$_H$.

Specify the payload storage size of the receive buffer by the RMPLS[2:0] bits in the RCFDCnCFDRMNB register. Do not read or write the RCFDCnCFDRMDFb_q register corresponding to an area that exceeds the specified size.

24.3.6 Details of Receive FIFO Buffer-related Registers

24.3.6.1 RCFDCnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RCFDCnCFDRFCCx register can be read or written in 32-bit units

RCFDCnCFDRFCCxL register can be read or written in 16-bit units

RCFDCnCFDRFCCxLL, RCFDCnCFDRFCCxLH registers can be read or written in 8-bit units

Address: RCFDCnCFDRFCCx: <RCFDCn_base> + $00B8_H$ + (04_H × x)

RCFDCnCFDRFCCxL: <RCFDCn_base> + $00B8_H$ + (04_H × x)

RCFDCnCFDRFCCxLL: <RCFDCn_base> + $00B8_H$ + (04_H × x),

RCFDCnCFDRFCCxLH: <RCFDCn_base> + $00B9_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Table 24.63 RCFDCnCFDRFCCx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	This bit is read as the value after reset. The write value should be the value after reset.

Table 24.63 RCFDCnCFDRFCCx Register Contents

Bit Position	Bit Name	Function
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
6 to 4	RFPLS[2:0]	Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3, 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive FIFO buffer. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 enables the use of FIFO buffers. Clearing this bit to 0 sets the RFEMP flag in the RCFDCnCFDRFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode. Set this bit to 1 with another instruction after the settings to all bits in the RCFDCnCFDRFCCx register have been done. This bit is cleared to 0 in global reset mode.

24.3.6.2 RCFDCnCFDRFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RCFDCnCFDRFSTSx register can be read or written in 32-bit units
RCFDCnCFDRFSTSxL register can be read or written in 16-bit units
RCFDCnCFDRFSTSxLL register can be read or written in 8-bit units
RCFDCnCFDRFSTSxLH register is a read-only register that can be read in 8-bit units

Address: RCFDCnCFDRFSTSx: <RCFDCn_base> + $00D8_H$ + (04_H × x)
RCFDCnCFDRFSTSxL: <RCFDCn_base> + $00D8_H$ + (04_H × x)
RCFDCnCFDRFSTSxLL: <RCFDCn_base> + $00D8_H$ + (04_H × x),
RCFDCnCFDRFSTSxLH: <RCFDCn_base> + $00D9_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.64 RCFDCnCFDRFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RCFDCnCFDRFCCx register is set to 0. This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RCFDCnCFDRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RCFDCnCFDRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RCFDCnCFDRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RCFDCnCFDRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when a received message is in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

24.3.6.3 RCFDCnCFDRFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RCFDCnCFDRFPCTRx register is a write-only register that can be written in 32-bit units

RCFDCnCFDRFPCTRxL register is a write-only register that can be written in 16-bit units

RCFDCnCFDRFPCTRxLL register is a write-only register that can be written in 8-bit units

Address: RCFDCnCFDRFPCTRx: <RCFDCn_base> + $00F8_H$ + (04_H × x)

RCFDCnCFDRFPCTRxL: <RCFDCn_base> + $00F8_H$ + (04_H × x)

RCFDCnCFDRFPCTRxLL: <RCFDCn_base> + $00F8_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 24.65 RCFDCnCFDRFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

When the RFDMAEx value in the RCFDCnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RCFDCnCFDRFSTSx register is decremented by 1. Read the RCFDCnCFDRFIDx, RCFDCnCFDRFPTRx, RCFDCnCFDRFFDSTSx, and RCFDCnCFDRFDFd_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RCFDCnCFDRFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RCFDCnCFDRFSTSx register is 0 (the receive FIFO buffer contains unread messages).

24.3.6.4 RCFDCnCFDRFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RCFDCnCFDRFIDx register is a read-only register that can be read in 32-bit units

RCFDCnCFDRFIDxL, RCFDCnCFDRFIDxH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRFIDxLL, RCFDCnCFDRFIDxLH, RCFDCnCFDRFIDxHL, RCFDCnCFDRFIDxHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRFIDx: <RCFDCn_base> + 6000$_H$ + (80$_H$ × x)
RCFDCnCFDRFIDxL: <RCFDCn_base> + 6000$_H$ + (80$_H$ × x),
RCFDCnCFDRFIDxH: <RCFDCn_base> + 6002$_H$ + (80$_H$ × x)
RCFDCnCFDRFIDxLL: <RCFDCn_base> + 6000$_H$ + (80$_H$ × x),
RCFDCnCFDRFIDxLH: <RCFDCn_base> + 6001$_H$ + (80$_H$ × x),
RCFDCnCFDRFIDxHL: <RCFDCn_base> + 6002$_H$ + (80$_H$ × x),
RCFDCnCFDRFIDxHH: <RCFDCn_base> + 6003$_H$ + (80$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.66 RCFDCnCFDRFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR/RRS ● When the received message is a classical CAN frame 0: Data frame 1: Remote frame ● When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	Reserved	This bit is read as the value after reset.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

24.3.6.5 RCFDCnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RCFDCnCFDRFPTRx register is a read-only register that can be read in 32-bit units

RCFDCnCFDRFPTRxL, RCFDCnCFDRFPTRxH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRFPTRxLL, RCFDCnCFDRFPTRxLH, RCFDCnCFDRFPTRxHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRFPTRx: <RCFDCn_base> + 6004_H + (80_H × x)
RCFDCnCFDRFPTRxL: <RCFDCn_base> + 6004_H + (80_H × x),
RCFDCnCFDRFPTRxH: <RCFDCn_base> + 6006_H + (80_H × x)
RCFDCnCFDRFPTRxLL: <RCFDCn_base> + 6004_H + (80_H × x),
RCFDCnCFDRFPTRxLH: <RCFDCn_base> + 6005_H + (80_H × x),
RCFDCnCFDRFPTRxHH: <RCFDCn_base> + 6007_H + (80_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.67 RCFDCnCFDRFPTRx Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>RFDLC[3:0]</td><td>Receive FIFO Buffer DLC Data
<table>
<tr><th>b31</th><th>b30</th><th>b29</th><th>b28</th><th>Classical CAN Frame</th><th>CAN FD Frame</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td></tr>
</table>
</td></tr>
<tr><td>27 to 16</td><td>Reserved</td><td>These bits are read as the value after reset.</td></tr>
<tr><td>15 to 0</td><td>RFTS[15:0]</td><td>Receive FIFO Buffer Timestamp Data
Timestamp value of the received message can be read.</td></tr>
</table>

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

24.3.6.6 RCFDCnCFDRFFDSTSx — Receive FIFO CAN FD Status Register (x = 0 to 7)

Access: RCFDCnCFDRFFDSTSx register is a read-only register that can be read in 32-bit units

RCFDCnCFDRFFDSTSxL, RCFDCnCFDRFFDSTSxH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRFFDSTSxLL, RCFDCnCFDRFFDSTSxHL, RCFDCnCFDRFFDSTSxHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRFFDSTSx: <RCFDCn_base> + 6008_H + (80_H × x)
RCFDCnCFDRFFDSTSxL: <RCFDCn_base> + 6008_H + (80_H × x),
RCFDCnCFDRFFDSTSxH: <RCFDCn_base> + $600A_H$ + (80_H × x)
RCFDCnCFDRFFDSTSxLL: <RCFDCn_base> + 6008_H + (80_H × x),
RCFDCnCFDRFFDSTSxHL: <RCFDCn_base> + $600A_H$ + (80_H × x),
RCFDCnCFDRFFDSTSxHH: <RCFDCn_base> + $600B_H$ + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.68 RCFDCnCFDRFFDSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	RFPTR[15:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 3	Reserved	These bits are read as the value after reset.
2	RFFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RFESI	ESI 0: Error active node 1: Error passive node

RFPTR[15:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive FIFO buffer.

RFBRS Bit

When the RFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

RFESI Bit

When the RFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

24.3.6.7 RCFDCnCFDRFDFd_x — Receive FIFO Buffer Access Data Field Register (d = 0 to 15, x = 0 to 7)

Access: RCFDCnCFDRFDFd_x register is a read-only register that can be read in 32-bit units

RCFDCnCFDRFDFd_xL, RCFDCnCFDRFDFd_xH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDRFDFd_xLL, RCFDCnCFDRFDFd_xLH, RCFDCnCFDRFDFd_xHL, RCFDCnCFDRFDFd_xHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDRFDFd_x: <RCFDCn_base> + 600C$_H$ + (04$_H$ × d) + (80$_H$ × x)
RCFDCnCFDRFDFd_xL: <RCFDCn_base> + 600C$_H$ + (04$_H$ × d) + (80$_H$ × x),
RCFDCnCFDRFDFd_xH: <RCFDCn_base> + 600E$_H$ + (04$_H$ × d) + (80$_H$ × x)
RCFDCnCFDRFDFd_xLL: <RCFDCn_base> + 600C$_H$ + (04$_H$ × d) + (80$_H$ × x),
RCFDCnCFDRFDFd_xLH: <RCFDCn_base> + 600D$_H$ + (04$_H$ × d) + (80$_H$ × x),
RCFDCnCFDRFDFd_xHL: <RCFDCn_base> + 600E$_H$ + (04$_H$ × d) + (80$_H$ × x),
RCFDCnCFDRFDFd_xHH: <RCFDCn_base> + 600F$_H$ + (04$_H$ × d) + (80$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB4 × d + 3 [7:0]								RFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB4 × d + 1 [7:0]								RFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.69 RCFDCnCFDRFDFd_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB4 × d + 3 [7:0]	Receive Buffer Data Byte 4 × d + 3 Receive Buffer Data Byte 4 × d + 2 Receive Buffer Data Byte 4 × d + 1 Receive Buffer Data Byte 4 × d + 0 Data for a message stored in the receive buffer can be read.
23 to 16	RFDB4 × d + 2 [7:0]	
15 to 8	RFDB4 × d + 1 [7:0]	
7 to 0	RFDB4 × d + 0 [7:0]	

When the RFDLC[3:0] value in the RCFDCnCFDRFPTRx register is smaller than the payload storage size of the receive FIFO buffer, data bytes for which no data is set are read as 00$_H$.

Specify the payload storage size of the receive FIFO buffer by the RFPLS[2:0] bits in the RCFDCnCFDRFCCx register. Do not read or write the RCFDCnCFDRFDFd_x register corresponding to an area that exceeds the specified size.

24.3.7 Transmit/Receive FIFO Buffer-related Registers

24.3.7.1 RCFDCnCFDCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register (k = 0 to 23)

Access: RCFDCnCFDCFCCk register can be read or written in 32-bit units

RCFDCnCFDCFCCkL, RCFDCnCFDCFCCkH registers can be read or written in 16-bit units

RCFDCnCFDCFCCkLL, RCFDCnCFDCFCCkLH, RCFDCnCFDCFCCkHL, RCFDCnCFDCFCCkHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCFCCk: <RCFDCn_base> + 0118_H + (04_H × k)

RCFDCnCFDCFCCkL: <RCFDCn_base> + 0118_H + (04_H × k),

RCFDCnCFDCFCCkH: <RCFDCn_base> + $011A_H$ + (04_H × k)

RCFDCnCFDCFCCkLL: <RCFDCn_base> + 0118_H + (04_H × k),

RCFDCnCFDCFCCkLH: <RCFDCn_base> + 0119_H + (04_H × k),

RCFDCnCFDCFCCkHL: <RCFDCn_base> + $011A_H$ + (04_H × k),

RCFDCnCFDCFCCkHH: <RCFDCn_base> + $011B_H$ + (04_H × k)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFDC[2:0]			CFTML[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	CFITR	CFITSS	CFM[1:0]		—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 24.70 RCFDCnCFDCFCCk Register Contents

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 21	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration B23 b22 b21 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
20 to 16	CFTML[4:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.

Table 24.70 RCFDCnCFDCFCCk Register Contents

Bit Position	Bit Name	Function
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: ● Receive mode/gateway mode When the number of received messages meets the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. ● Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: ● Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. ● Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock obtained by dividing pclk/2 by the value of the ITRCP [15:0] bits 1: Clock obtained by dividing pclk/2 by “value of ITRCP [15:0] bits x 10”
10	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the nominal bit time clock for the channel to which the FIFO is linked.
9, 8	CFM[1:0]	Transmit/Receive FIFO Mode Select b9 b8 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
7	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
6 to 4	CFPLS[2:0]	Transmit/Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.

Table 24.70 RCFDCnCFDCFCCk Register Contents

Bit Position	Bit Name	Function
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTML[4:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/ receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[4:0]) (See **Table 24.29, Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[4:0]**).

See **Table 24.27, Transmit Buffer p Allocated to Each Channel** and **Table 24.28, Transmit/Receive FIFO Buffer k Allocated to Each Channel**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[4:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RCFDCnCFDGCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RCFDCnCFDGCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer. When this bit is 1, the nominal bit time clock of the channel to which the FIFO is linked is the count source of the interval timer. Use this count source only for the channel which does not handle the CAN FD frames.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RCFDCnCFDCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RCFDCnCFDCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 enables the transmit/receive FIFO buffers.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration- lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RCFDCnCFDCFCCk register have been set, set this bit to 1 by using another instruction.

24.3.7.2 RCFDCnCFDCFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 23)

Access:	RCFDCnCFDCFSTSk register can be read or written in 32-bit units RCFDCnCFDCFSTSkL register can be read or written in 16-bit units RCFDCnCFDCFSTSkLL register can be read or written in 8-bit units RCFDCnCFDCFSTSkLH register is a read-only register that can be read in 8-bit units
Address:	RCFDCnCFDCFSTSk: <RCFDCn_base> + 0178_H + (04_H × k) RCFDCnCFDCFSTSkL: <RCFDCn_base> + 0178_H + (04_H × k) RCFDCnCFDCFSTSkLL: <RCFDCn_base> + 0178_H + (04_H × k), RCFDCnCFDCFSTSkLH: <RCFDCn_base> + 0179_H + (04_H × k)
Value after reset:	0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.71 RCFDCnCFDCFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values according to the setting of the CFM[1:0] bits in the RCFDCnCFDCFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RCFDCnCFDCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the interrupt source selected by the CFIM bit in the RCFDCnCFDCFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RCFDCnCFDCFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RCFDCnCFDCFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RCFDCnCFDCFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RCFDCnCFDCFPCTRk register after data was written to the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k register

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

24.3.7.3 RCFDCnCFDCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 23)

Access:	RCFDCnCFDCFPCTRk register is a write-only register that can be written in 32-bit units RCFDCnCFDCFPCTRkL register is a write-only register that can be written in 16-bit units RCFDCnCFDCFPCTRkLL register is a write-only register that can be written in 8-bit units
Address:	RCFDCnCFDCFPCTRk: <RCFDCn_base> + $01D8_H$ + (04_H × k) RCFDCnCFDCFPCTRkL: <RCFDCn_base> + $01D8_H$ + (04_H × k) RCFDCnCFDCFPCTRkLL: <RCFDCn_base> + $01D8_H$ + (04_H × k)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 24.72 RCFDCnCFDCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control ● Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. ● Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. ● Gateway mode: Setting prohibited

When the corresponding transmit/receive FIFO buffer is the first transmit/receive FIFO buffer (k = 3 × m) allocated to channel m and when the CFDMAEm bit in the RCFDCnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RCFDCnCFDCFCCk register is 00_B):

  Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RCFDCnCFDCFSTSk register is decremented by 1. Read the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.

  When writing FF_H to these bits, make sure that the CFE bit in the RCFDCnCFDCFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RCFDCnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] value in the RCFDCnCFDCFCCk register is 01_B):

  Writing FF_H to the CFPC[7:0] bits stores the data written to the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k registers before writing FF_H to the CFPC[7:0] bits.

  When writing FF_H to these bits, make sure that the CFE bit in the RCFDCnCFDCFCCk register is set to 1 and the CFFLL flag in the RCFDCnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).

- Gateway mode (CFM[1:0] value in the RCFDCnCFDCFCCk register is 10_B): Setting prohibited

24.3.7.4 RCFDCnCFDCFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 23)

Access: RCFDCnCFDCFIDk register can be read or written in 32-bit units

RCFDCnCFDCFIDkL, RCFDCnCFDCFIDkH registers can be read or written in 16-bit units

RCFDCnCFDCFIDkLL, RCFDCnCFDCFIDkLH, RCFDCnCFDCFIDkHL, RCFDCnCFDCFIDkHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCFIDk: <RCFDCn_base> + 6400_H + (80_H × k)

RCFDCnCFDCFIDkL: <RCFDCn_base> + 6400_H + (80_H × k),

RCFDCnCFDCFIDkH: <RCFDCn_base> + 6402_H + (80_H × k)

RCFDCnCFDCFIDkLL: <RCFDCn_base> + 6400_H + (80_H × k),

RCFDCnCFDCFIDkLH: <RCFDCn_base> + 6401_H + (80_H × k),

RCFDCnCFDCFIDkHL: <RCFDCn_base> + 6402_H + (80_H × k),

RCFDCnCFDCFIDkHH: <RCFDCn_base> + 6403_H + (80_H × k)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.73 RCFDCnCFDCFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR/RRS ● When the CFM[1:0] value is 01_B (transmit mode) – When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame – When the transmit message is a CAN FD frame Write 0 to this bit. ● When the CFM[1:0] value is 00_B (receive mode) – When the received message is a classical CAN frame 0: Data frame 1: Remote frame – When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.

Table 24.73 RCFDCnCFDCFIDk Register Contents

Bit Position	Bit Name	Function
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data ● When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. ● When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register can be written only when the CFM[1:0] value in the RCFDCnCFDCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This RCFDCnCFDCFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

If the received message is a classical CAN frame, this bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. If the received message is a CAN FD frame, this bit indicates the RRS bit value of the received message.

When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFFDF bit in the RCFDCnCFDCFFDCSTSk register is 1 (CAN FD frame), set this bit to 0.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

24.3.7.5 RCFDCnCFDCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 23)

Access: RCFDCnCFDCFPTRk register can be read or written in 32-bit units

RCFDCnCFDCFPTRkL, RCFDCnCFDCFPTRkH registers can be read or written in 16-bit units

RCFDCnCFDCFPTRkLL, RCFDCnCFDCFPTRkLH, RCFDCnCFDCFPTRkHH registers can be read or written in 8-bit units

Address: RCFDCnCFDCFPTRk: <RCFDCn_base> + 6404_H + (80_H × k)

RCFDCnCFDCFPTRkL: <RCFDCn_base> + 6404_H + (80_H × k),

RCFDCnCFDCFPTRkH: <RCFDCn_base> + 6406_H + (80_H × k)

RCFDCnCFDCFPTRkLL: <RCFDCn_base> + 6404_H + (80_H × k),

RCFDCnCFDCFPTRkLH: <RCFDCn_base> + 6405_H + (80_H × k),

RCFDCnCFDCFPTRkHH: <RCFDCn_base> + 6407_H + (80_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.74 RCFDCnCFDCFPTRk Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>CFDLC[3:0]</td><td>Transmit/Receive FIFO Buffer DLC Data
<table>
<tr><th>b31</th><th>b30</th><th>b29</th><th>b28</th><th>Classical CAN Frame</th><th>CAN FD Frame</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td rowspan="9"></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td rowspan="7">8 data bytes</td><td>12 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>16 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>20 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>24 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>32 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>48 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>64 data bytes</td></tr>
</table>
</td></tr>
<tr><td>27 to 16</td><td>Reserved</td><td>These bits are read as the value after reset.
The write value should be the value after reset.</td></tr>
<tr><td>15 to 0</td><td>CFTS[15:0]</td><td>Transmit/Receive FIFO Buffer Timestamp Data
These bits are valid only when the CFM[1:0] value is 00_B (receive mode). The timestamp value of the received message can be read.</td></tr>
</table>

This register can be written only when the CFM[1:0] value in the RCFDCnCFDCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFDLC[3:0] bits are set to 1001_B or more while the CFFDF bit in the RCFDCnCFDCFFDCSTSk register is 0 (classical CAN frame), 8-byte data is transmitted actually.

A value of 0000_B to 1111_B is settable. If the specified data length exceeds the payload storage size specified by the CFPLS[2:0] bits, excessive payloads are padded by CC_H.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer. These bits are valid when the CFM[1:0] value is 00_B.

24.3.7.6 RCFDCnCFDCFFDCSTSk — Transmit/receive FIFO CAN FD Configuration/status Register (k = 0 to 23)

Access: RCFDCnCFDCFFDCSTSk register can be read or written in 32-bit units

RCFDCnCFDCFFDCSTSkL register can be read or written in 16-bit units

RCFDCnCFDCFFDCSTSkH register is a read-only register that can be read in 16-bit units

RCFDCnCFDCFFDCSTSkLL register can be read or written in 8-bit units

RCFDCnCFDCFFDCSTSkHL, RCFDCnCFDCFFDCSTSkHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCFFDCSTSk: <RCFDCn_base> + 6408_H + (80_H × k)

RCFDCnCFDCFFDCSTSkL: <RCFDCn_base> + 6408_H + (80_H × k),

RCFDCnCFDCFFDCSTSkH: <RCFDCn_base> + $640A_H$ + (80_H × k)

RCFDCnCFDCFFDCSTSkLL: <RCFDCn_base> + 6408_H + (80_H × k),

RCFDCnCFDCFFDCSTSkHL: <RCFDCn_base> + $640A_H$ + (80_H × k),

RCFDCnCFDCFFDCSTSkHH: <RCFDCn_base> + $640B_H$ + (80_H × k)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24.75 RCFDCnCFDCFFDCSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	CFPTR[15:0]	Transmit/Receive FIFO Buffer Label Data ● When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[15:0] are valid. ● When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
15 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	CFFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	CFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	CFESI	ESI 0: Error active node 1: Error passive node

This register can be written only when the CFM[1:0] value in the RCFDCnCFDCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). Do not read or write this register when the CFM[1:0] value is 10_B (gateway mode).

CFPTR[15:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[15:0] value is stored in the transmit history buffer when message transmission has been completed.

CFFDF Bit

When the CFM[1:0] value is 00_B, this bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the transmit/receive FIFO buffer. When the CFM[1:0] value is 01_B, this bit is used to set the FD format of the message to be transmitted from the transmit/receive FIFO buffer.

CFBRS Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the BRS bit value of the message to be transmitted from the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, write 0 to this bit.

CFESI Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the ESI bit value of the message to be transmitted from the transmit/receive FIFO buffer. The set value is transmitted when the ESIC bit in the RCFDCnCFDCmFDCFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the CFFDF bit is set to 0, write 0 to this bit.

24.3.7.7 RCFDCnCFDCFDFd_k — Transmit/receive FIFO Buffer Access Data Field Register (d = 0 to 15, k = 0 to 23)

Access:	RCFDCnCFDCFDFd_k register can be read or written in 32-bit units RCFDCnCFDCFDFd_kL, RCFDCnCFDCFDFd_kH registers can be read or written in 16-bit units RCFDCnCFDCFDFd_kLL, RCFDCnCFDCFDFd_kLH, RCFDCnCFDCFDFd_kHL, RCFDCnCFDCFDFd_kHH registers can be read or written in 8-bit units
Address:	RCFDCnCFDCFDFd_k: <RCFDCn_base> + $640C_H + (04_H \times d) + (80_H \times k)$ RCFDCnCFDCFDFd_kL: <RCFDCn_base> + $640C_H + (04_H \times d) + (80_H \times k)$, RCFDCnCFDCFDFd_kH: <RCFDCn_base> + $640E_H + (04_H \times d) + (80_H \times k)$ RCFDCnCFDCFDFd_kLL: <RCFDCn_base> + $640C_H + (04_H \times d) + (80_H \times k)$, RCFDCnCFDCFDFd_kLH: <RCFDCn_base> + $640D_H + (04_H \times d) + (80_H \times k)$, RCFDCnCFDCFDFd_kHL: <RCFDCn_base> + $640E_H + (04_H \times d) + (80_H \times k)$, RCFDCnCFDCFDFd_kHH: <RCFDCn_base> + $640F_H + (04_H \times d) + (80_H \times k)$
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB4 × d + 3 [7:0]								CFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB4 × d + 1 [7:0]								CFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.76 RCFDCnCFDCFDFd_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB4 × d + 3 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 3 Transmit/Receive FIFO Buffer Data Byte 4 × d + 2 Transmit/Receive FIFO Buffer Data Byte 4 × d + 1 Transmit/Receive FIFO Buffer Data Byte 4 × d + 0 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.
23 to 16	CFDB4 × d + 2 [7:0]	
15 to 8	CFDB4 × d + 1 [7:0]	
7 to 0	CFDB4 × d + 0 [7:0]	

This register can be written only when the CFM[1:0] value in the RCFDCnCFDCFCCk register is 01_B (transmit mode).

This register can be read only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RCFDCnCFDCFPTRk register is smaller than the payload storage size of the transmit/receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the transmit/receive FIFO buffer by the CFPLS[2:0] bits in the RCFDCnCFDCFCCk register. Do not read or write the RCFDCnCFDCFDFd_k register corresponding to an area that exceeds the specified size.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

24.3.8 Details of FIFO Status-related Registers

24.3.8.1 RCFDCnCFDFESTS — FIFO Empty Status Register

Access: RCFDCnCFDFESTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDFESTSL, RCFDCnCFDFESTSH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDFESTSLL, RCFDCnCFDFESTSLH, RCFDCnCFDFESTSHL, RCFDCnCFDFESTSHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDFESTS: <RCFDCn_base> + 0238_H
RCFDCnCFDFESTSL: <RCFDCn_base> + 0238_H,
RCFDCnCFDFESTSH: <RCFDCn_base> + $023A_H$
RCFDCnCFDFESTSLL: <RCFDCn_base> + 0238_H,
RCFDCnCFDFESTSLH: <RCFDCn_base> + 0239_H,
RCFDCnCFDFESTSHL: <RCFDCn_base> + $023A_H$,
RCFDCnCFDFESTSHH: <RCFDCn_base> + $023B_H$

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CF23EMP	CF22EMP	CF21EMP	CF20EMP	CF19EMP	CF18EMP	CF17EMP	CF16EMP	CF15EMP	CF14EMP	CF13EMP	CF12EMP	CF11EMP	CF10EMP	CF9EMP	CF8EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.77 RCFDCnCFDFESTS Register Contents

Bit Position	Bit Name	Function
31	CF23EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 23)
30	CF22EMP	
29	CF21EMP	
28	CF20EMP	
27	CF19EMP	
26	CF18EMP	
25	CF17EMP	
24	CF16EMP	
23	CF15EMP	
22	CF14EMP	
21	CF13EMP	
20	CF12EMP	
19	CF11EMP	
18	CF10EMP	
17	CF9EMP	
16	CF8EMP	
15	CF7EMP	
14	CF6EMP	

Table 24.77 RCFDCnCFDFESTS Register Contents

Bit Position	Bit Name	Function
13	CF5EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 23)
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message . (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RCFDCnCFDFESTS register is set to FFFF FFFF_H in global reset mode.

CFkEMP Flag (k = 0 to 23)

The CFkEMP flag is set to 1 when the CFEMP flag in the RCFDCnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RCFDCnCFDRFSTSx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

24.3.8.2 RCFDCnCFDFFSTS — FIFO Full Status Register

Access: RCFDCnCFDFFSTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDFFSTSL, RCFDCnCFDFFSTSH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDFFSTSLL, RCFDCnCFDFFSTSLH, RCFDCnCFDFFSTSHL, RCFDCnCFDFFSTSHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDFFSTS: <RCFDCn_base> + $023C_H$
RCFDCnCFDFFSTSL: <RCFDCn_base> + $023C_H$,
RCFDCnCFDFFSTSH: <RCFDCn_base> + $023E_H$
RCFDCnCFDFFSTSLL: <RCFDCn_base> + $023C_H$,
RCFDCnCFDFFSTSLH: <RCFDCn_base> + $023D_H$,
RCFDCnCFDFFSTSHL: <RCFDCn_base> + $023E_H$,
RCFDCnCFDFFSTSHH: <RCFDCn_base> + $023F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CF23FLL	CF22FLL	CF21FLL	CF20FLL	CF19FLL	CF18FLL	CF17FLL	CF16FLL	CF15FLL	CF14FLL	CF13FLL	CF12FLL	CF11FLL	CF10FLL	CF9FLL	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.78 RCFDCnCFDFFSTS Register Contents

Bit Position	Bit Name	Function
31	CF23FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive FIFO buffer k is not full. 1: Transmit/receive FIFO buffer k is full. (k = 0 to 23)
30	CF22FLL	
29	CF21FLL	
28	CF20FLL	
27	CF19FLL	
26	CF18FLL	
25	CF17FLL	
24	CF16FLL	
23	CF15FLL	
22	CF14FLL	
21	CF13FLL	
20	CF12FLL	
19	CF11FLL	
18	CF10FLL	
17	CF9FLL	
16	CF8FLL	
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	

Table 24.78 RCFDCnCFDFFSTS Register Contents

Bit Position	Bit Name	Function
11	CF3EMP	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive FIFO buffer k is not full. 1: Transmit/receive FIFO buffer k is full. (k = 0 to 23)
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RCFDCnCFDFFSTS register is cleared to 0000 0000$_H$ in global reset mode.

CFkFLL Flag (k = 0 to 23)

The CFkFLL flag is set to 1 when the CFFLL flag in the RCFDCnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RCFDCnCFDRFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

24.3.8.3 RCFDCnCFDFMSTS — FIFO Message Lost Status Register

Access: RCFDCnCFDFMSTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDFMSTSL, RCFDCnCFDFMSTSH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDFMSTSLL, RCFDCnCFDFMSTSLH, RCFDCnCFDFMSTSHL, RCFDCnCFDFMSTSHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDFMSTS: <RCFDCn_base> + 0240_H

RCFDCnCFDFMSTSL: <RCFDCn_base> + 0240_H,

RCFDCnCFDFMSTSH: <RCFDCn_base> + 0242_H

RCFDCnCFDFMSTSLL: <RCFDCn_base> + 0240_H,

RCFDCnCFDFMSTSLH: <RCFDCn_base> + 0241_H,

RCFDCnCFDFMSTSHL: <RCFDCn_base> + 0242_H,

RCFDCnCFDFMSTSHH: <RCFDCn_base> + 0243_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CF23MLT	CF22MLT	CF21MLT	CF20MLT	CF19MLT	CF18MLT	CF17MLT	CF16MLT	CF15MLT	CF14MLT	CF13MLT	CF12MLT	CF11MLT	CF10MLT	CF9MLT	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.79 RCFDCnCFDFMSTS Register Contents

Bit Position	Bit Name	Function
31	CF23MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 23)
30	CF22MLT	
29	CF21MLT	
28	CF20MLT	
27	CF19MLT	
26	CF18MLT	
25	CF17MLT	
24	CF16MLT	
23	CF15MLT	
22	CF14MLT	
21	CF13MLT	
20	CF12MLT	
19	CF11MLT	
18	CF10MLT	
17	CF9MLT	
16	CF8MLT	
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	

Table 24.79 RCFDCnCFDFMSTS Register Contents

Bit Position	Bit Name	Function
11	CF3MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 23)
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RCFDCnCFDFMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = 0 to 23)

The CFkMLT flag is set to 1 when the CFMLT flag in the RCFDCnCFDCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RCFDCnCFDRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

24.3.8.4 RCFDCnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access:	RCFDCnCFDRFISTS register is a read-only register that can be read in 32-bit units RCFDCnCFDRFISTSL register is a read-only register that can be read in 16-bit units RCFDCnCFDRFISTSLL register is a read-only register that can be read in 8-bit units
Address:	RCFDCnCFDRFISTS: <RCFDCn_base> + 0244$_H$ RCFDCnCFDRFISTSL: <RCFDCn_base> + 0244$_H$ RCFDCnCFDRFISTSLL: <RCFDCn_base> + 0244$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.80 RCFDCnCFDRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RCFDCnCFDRFISTS register is cleared to 0000 0000$_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RCFDCnCFDRFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

24.3.8.5 RCFDCnCFDCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RCFDCnCFDCFRISTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDCFRISTSL, RCFDCnCFDCFRISTSH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDCFRISTSLL, RCFDCnCFDCFRISTSLH, RCFDCnCFDCFRISTSHL registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCFRISTS: <RCFDCn_base> + 0248$_{H}$

RCFDCnCFDCFRISTSL: <RCFDCn_base> + 0248$_{H}$,

RCFDCnCFDCFRISTSH: <RCFDCn_base> + 024A$_{H}$

RCFDCnCFDCFRISTSLL: <RCFDCn_base> + 0248$_{H}$,

RCFDCnCFDCFRISTSLH: <RCFDCn_base> + 0249$_{H}$,

RCFDCnCFDCFRISTSHL: <RCFDCn_base> + 024A$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CF23RXIF	CF22RXIF	CF21RXIF	CF20RXIF	CF19RXIF	CF18RXIF	CF17RXIF	CF16RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15RXIF	CF14RXIF	CF13RXIF	CF12RXIF	CF11RXIF	CF10RXIF	CF9RXIF	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.81 RCFDCnCFDCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	These bits are read as the value after reset.
23	CF23RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 23)
22	CF22RXIF	
21	CF21RXIF	
20	CF20RXIF	
19	CF19RXIF	
18	CF18RXIF	
17	CF17RXIF	
16	CF16RXIF	
15	CF15RXIF	
14	CF14RXIF	
13	CF13RXIF	
12	CF12RXIF	
11	CF11RXIF	
10	CF10RXIF	
9	CF9RXIF	
8	CF8RXIF	
7	CF7RXIF	
6	CF6RXIF	

Table 24.81 RCFDCnCFDCFRISTS Register Contents

Bit Position	Bit Name	Function
5	CF5RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 23)
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RCFDCnCFDCFRISTS register is cleared to 0000 0000_H in global reset mode.

CFkRXIF Flag (k = 0 to 23)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RCFDCnCFDCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

24.3.8.6 RCFDCnCFDCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RCFDCnCFDCFTISTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDCFTISTSL, RCFDCnCFDCFTISTSH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDCFTISTSLL, RCFDCnCFDCFTISTSLH, RCFDCnCFDCFTISTSHL registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCFTISTS: <RCFDCn_base> + $024C_H$

RCFDCnCFDCFTISTSL: <RCFDCn_base> + $024C_H$,

RCFDCnCFDCFTISTSH: <RCFDCn_base> + $024E_H$

RCFDCnCFDCFTISTSLL: <RCFDCn_base> + $024C_H$,

RCFDCnCFDCFTISTSLH: <RCFDCn_base> + $024D_H$,

RCFDCnCFDCFTISTSHL: <RCFDCn_base> + $024E_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CF23TXIF	CF22TXIF	CF21TXIF	CF20TXIF	CF19TXIF	CF18TXIF	CF17TXIF	CF16TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15TXIF	CF14TXIF	CF13TXIF	CF12TXIF	CF11TXIF	CF10TXIF	CF9TXIF	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.82 RCFDCnCFDCFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	These bits are read as the value after reset.
23	CF23TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 23)
22	CF22TXIF	
21	CF21TXIF	
20	CF20TXIF	
19	CF19TXIF	
18	CF18TXIF	
17	CF17TXIF	
16	CF16TXIF	
15	CF15TXIF	
14	CF14TXIF	
13	CF13TXIF	
12	CF12TXIF	
11	CF11TXIF	
10	CF10TXIF	
9	CF9TXIF	
8	CF8TXIF	
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	

Table 24.82 RCFDCnCFDCFTISTS Register Contents

Bit Position	Bit Name	Function
3	CF3TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 23)
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RCFDCnCFDCFTISTS register is cleared to 0000 0000_H in global reset mode.

CFkTXIF Flag (k = 0 to 23)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RCFDCnCFDCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

24.3.9 Details of FIFO DMA-related Registers

24.3.9.1 RCFDCnCFDCDTCT — DMA Enable Register

Access: RCFDCnCFDCDTCT register can be read or written in 32-bit units

RCFDCnCFDCDTCTL register can be read or written in 16-bit units

RCFDCnCFDCDTCTLL, RCFDCnCFDCDTCTLH registers can be read or written in 8-bit units

Address: RCFDCnCFDCDTCT: <RCFDCn_base> + 0640_H

RCFDCnCFDCDTCTL: <RCFDCn_base> + 0640_H

RCFDCnCFDCDTCTLL: <RCFDCn_base> + 0640_H,

RCFDCnCFDCDTCTLH: <RCFDCn_base> + 0641_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDMAE7	CFDMAE6	CFDMAE5	CFDMAE4	CFDMAE3	CFDMAE2	CFDMAE1	CFDMAE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.83 RCFDCnCFDCDTCT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15	CFDMAE7	Transmit/Receive FIFO Buffer 21 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 21 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 21 is enabled.
14	CFDMAE6	Transmit/Receive FIFO Buffer 18 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 18 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 18 is enabled.
13	CFDMAE5	Transmit/Receive FIFO Buffer 15 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 15 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 15 is enabled.
12	CFDMAE4	Transmit/Receive FIFO Buffer 12 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 12 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 12 is enabled.
11	CFDMAE3	Transmit/Receive FIFO Buffer 9 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 9 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 9 is enabled.
10	CFDMAE2	Transmit/Receive FIFO Buffer 6 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 6 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 6 is enabled.

Table 24.83 RCFDCnCFDCDTCT Register Contents

Bit Position	Bit Name	Function
9	CFDMAE1	Transmit/Receive FIFO Buffer 3 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 3 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 3 is enabled.
8	CFDMAE0	Transmit/Receive FIFO Buffer 0 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 0 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 0 is enabled.
7	RFDMAE7	RH850/F1KH-D8, RH850/F1KM-S4: Receive FIFO Buffer x DMA Enable 0: A DMA transfer request of receive FIFO buffer x is disabled. 1: A DMA transfer request of receive FIFO buffer x is enabled. (x = 0 to 7) RH850/F1KM-S1: Receive FIFO Buffer x DMA Enable 0: A DMA transfer request of receive FIFO buffer x is disabled. 1: A DMA transfer request of receive FIFO buffer x is enabled. (x = 0 to 5)
6	RFDMAE6	
5	RFDMAE5	
4	RFDMAE4	
3	RFDMAE3	
2	RFDMAE2	
1	RFDMAE1	
0	RFDMAE0	

Modify the RCFDCnCFDCDTCT register in global operating mode or global test mode.

CFDMAEm Bit

This bit is used to enable DMA transfer for transmit/receive FIFO buffer 3 × m (the first transmit/ receive FIFO buffer allocated to channel m). DMA transfer is enabled only for transmit/receive FIFO buffers for which the CFM[1:0] bits in the RCFDCnCFDCFCCk register is set to 00_B (receive mode).

Set this bit to 0 when the CFM[1:0] value is 01_B (transmit mode) or 10_B (gateway mode).

RFDMAEx Bit

This bit is used to enable DMA transfer for receive FIFO buffer x.

24.3.9.2 RCFDCnCFDCDTSTS — DMA Status Register

Access: RCFDCnCFDCDTSTS register is a read-only register that can be read in 32-bit units

RCFDCnCFDCDTSTSL register is a read-only register that can be read in 16-bit units

RCFDCnCFDCDTSTSLL, RCFDCnCFDCDTSTSLH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDCDTSTS: <RCFDCn_base> + 0644_H

RCFDCnCFDCDTSTSL: <RCFDCn_base> + 0644_H

RCFDCnCFDCDTSTSLL: <RCFDCn_base> + 0644_H,

RCFDCnCFDCDTSTSLH: <RCFDCn_base> + 0645_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDMASTS7	CFDMASTS6	CFDMASTS5	CFDMASTS4	CFDMASTS3	CFDMASTS2	CFDMASTS1	CFDMASTS0	RFDMASTS7	RFDMASTS6	RFDMASTS5	RFDMASTS4	RFDMASTS3	RFDMASTS2	RFDMASTS1	RFDMASTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.84 RCFDCnCFDCDTSTS Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15	CFDMASTS7	Transmit/Receive FIFO Buffer 21 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 21 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 21 is in progress.
14	CFDMASTS6	Transmit/Receive FIFO Buffer 18 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 18 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 18 is in progress.
13	CFDMASTS5	Transmit/Receive FIFO Buffer 15 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 15 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 15 is in progress.
12	CFDMASTS4	Transmit/Receive FIFO Buffer 12 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 12 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 12 is in progress.
11	CFDMASTS3	Transmit/Receive FIFO Buffer 9 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 9 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 9 is in progress.
10	CFDMASTS2	Transmit/Receive FIFO Buffer 6 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 6 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 6 is in progress.
9	CFDMASTS1	Transmit/Receive FIFO Buffer 3 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 3 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 3 is in progress.

Table 24.84 RCFDCnCFDCDTSTS Register Contents

Bit Position	Bit Name	Function
8	CFDMASTS0	Transmit/Receive FIFO Buffer 0 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 0 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 0 is in progress.
7	RFDMASTS7	RH850/F1KH-D8, RH850/F1KM-S4: Receive FIFO Buffer x DMA Status 0: DMA transfer of receive FIFO buffer x is not in progress. 1: DMA transfer of receive FIFO buffer x is in progress. (x = 0 to 7) RH850/F1KM-S1: Receive FIFO Buffer x DMA Status 0: DMA transfer of receive FIFO buffer x is not in progress. 1: DMA transfer of receive FIFO buffer x is in progress. (x = 0 to 5)
6	RFDMASTS6	
5	RFDMASTS5	
4	RFDMASTS4	
3	RFDMASTS3	
2	RFDMASTS2	
1	RFDMASTS1	
0	RFDMASTS0	

CFDMASTSm Bit

When DMA transfer is enabled (CFDMAEm bit in the RCFDCnCFDCDTCT register is 1) for the transmit/receive FIFO buffer 3 × m (the first transmit/receive FIFO buffer allocated to channel m) while the transmit/receive FIFO buffer contains one of more messages, the CFDMASTSm bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the transmit/receive FIFO buffer have been transferred or DMA transfer is disabled (CFDMAEm bit is 0), the CFDMASTSm bit is cleared to 0 indicating that DMA transfer has been completed. If the CFDMAEm bit is set to 0 during DMA transfer, the CFDMASTSm bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area).

These bits are cleared to 0 in global reset mode.

RFDMASTSx Bit

When DMA transfer is enabled (corresponding RFDMAEx bit in the RCFDCnCFDCDTCT register is 1) for the receive FIFO buffer x and the receive FIFO buffer contains one of more messages, the RFDMASTSx bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the receive FIFO buffer x have been transferred or DMA transfer is disabled (RFDMAEx bit = 0), the RFDMASTSx bit is cleared to 0 indicating that DMA transfer has been completed. If the RFDMAEx bit is set to 0 during DMA transfer, the RFDMASTSx bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area) These bits are cleared to 0 in global reset mode.

24.3.10 Details of Transmit Buffer-related Registers

24.3.10.1 RCFDCnCFDTMCp — Transmit Buffer Control Register (p = 0 to 255)

Access: RCFDCnCFDTMCp register can be read or written in 8-bit units

Address: RCFDCnCFDTMCp: <RCFDCn_base> + 0250_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*[1]	R/W*[1]

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 24.85 RCFDCnCFDTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RCFDCnCFDTMCp register meets any of the following conditions, set it to 00_H.

- The RCFDCnCFDTMCp register corresponds to the transmit buffer number selected by the CFTML[4:0] bits in the RCFDCnCFDCFCCk register (p = m × 32 + the value of CFTML[4:0] bits).
- The RCFDCnCFDTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[4:0] bits in the RCFDCnCFDTXQCCm (m = 0 to 7) register (p = (m × 32 + 31) to (m × 32 + 31 – the value of TXQDC[4:0] bits)).

All of the bits in the RCFDCnCFDTMCp register are cleared to 0 in channel reset mode. Modify the RCFDCnCFDTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. If transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RCFDCnCFDTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RCFDCnCFDTMSTSp register is 00_B.

24.3.10.2 RCFDCnCFDTMSTSp — Transmit Buffer Status Register (p = 0 to 255)

Access: RCFDCnCFDTMSTSp register can be read or written in 8-bit units

Address: RCFDCnCFDTMSTSp: <RCFDCn_base> + 0350_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 24.86 RCFDCnCFDTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

All of the bits in the RCFDCnCFDTMSTSp register are cleared to 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RCFDCnCFDTMCp register is set to 1. The TMTARM flag is set to 0 when the TMTAR bit in the RCFDCnCFDTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RCFDCnCFDTMCp register is set to 1. The TMTRM flag is set to 0 when the TMTR bit in the RCFDCnCFDTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RCFDCnCFDTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RCFDCnCFDTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

24.3.10.3 RCFDCnCFDTMIDp — Transmit Buffer ID Register (p = 0 to 255)

Access:	RCFDCnCFDTMIDp register can be read or written in 32-bit units RCFDCnCFDTMIDpL, RCFDCnCFDTMIDpH registers can be read or written in 16-bit units RCFDCnCFDTMIDpLL, RCFDCnCFDTMIDpLH, RCFDCnCFDTMIDpHL, RCFDCnCFDTMIDpHH registers can be read or written in 8-bit units
Address:	RCFDCnCFDTMIDp: <RCFDCn_base> + 8000_H + (80_H × p) RCFDCnCFDTMIDpL: <RCFDCn_base> + 8000_H + (80_H × p), RCFDCnCFDTMIDpH: <RCFDCn_base> + 8002_H + (80_H × p) RCFDCnCFDTMIDpLL: <RCFDCn_base> + 8000_H + (80_H × p), RCFDCnCFDTMIDpLH: <RCFDCn_base> + 8001_H + (80_H × p), RCFDCnCFDTMIDpHL: <RCFDCn_base> + 8002_H + (80_H × p), RCFDCnCFDTMIDpHH: <RCFDCn_base> + 8003_H + (80_H × p)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.87 RCFDCnCFDTMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR/RRS ● When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame ● When the transmit message is a CAN FD frame Write 0 to this bit.
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RCFDCnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 32 + 31) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

Set this bit to 0 when the TMFDF bit in the RCFDCnCFDTMFDCTRp register is 1 (CAN FD frame).

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

24.3.10.4 RCFDCnCFDTMPTRp — Transmit Buffer Pointer Register (p = 0 to 255)

Access: RCFDCnCFDTMPTRp register can be read or written in 32-bit units

RCFDCnCFDTMPTRpH register can be read or written in 16-bit units

RCFDCnCFDTMPTRpHH register can be read or written in 8-bit units

Address: RCFDCnCFDTMPTRp: <RCFDCn_base> + 8004_H + (80_H × p)

RCFDCnCFDTMPTRpH: <RCFDCn_base> + 8006_H + (80_H × p)

RCFDCnCFDTMPTRpHH: <RCFDCn_base> + 8007_H + (80_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.88 RCFDCnCFDTMPTRp Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>TMDLC[3:0]</td><td>Transmit Buffer DLC Data
<table>
<tr><th>b31</th><th>b30</th><th>b29</th><th>b28</th><th>Classical CAN Frame</th><th>CAN FD Frame</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td></tr>
</table>
</td></tr>
<tr><td>27 to 0</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to these bits, write the value after reset.</td></tr>
</table>

Modify this register when the TMTRM bit in the corresponding RCFDCnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 32 + 31) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RCFDCnCFDTMIDp register is set to 0 (data frame).

When the TMDLC[3:0] bits are set to 1001_B or more while the TMFDF bit in the RCFDCnCFDTMFDCTRp register is 0 (classical CAN frame), 8-byte data is transmitted actually. When the TMRTR bit is 1 (remote frame), these bits set the length of the message to be requested.

24.3.10.5 RCFDCnCFDTMFDCTRp — Transmit Buffer CAN FD Configuration Register (p = 0 to 255)

Access:	RCFDCnCFDTMFDCTRp register can be read or written in 32-bit units RCFDCnCFDTMFDCTRpL, RCFDCnCFDTMFDCTRpH registers can be read or written in 16-bit units RCFDCnCFDTMFDCTRpLL, RCFDCnCFDTMFDCTRpHL, RCFDCnCFDTMFDCTRpHH registers can be read or written in 8-bit units
Address:	RCFDCnCFDTMFDCTRp: <RCFDCn_base> + 8008_H + (80_H × p) RCFDCnCFDTMFDCTRpL: <RCFDCn_base> + 8008_H + (80_H × p), RCFDCnCFDTMFDCTRpH: <RCFDCn_base> + $800A_H$ + (80_H × p) RCFDCnCFDTMFDCTRpLL: <RCFDCn_base> + 8008_H + (80_H × p), RCFDCnCFDTMFDCTRpHL: <RCFDCn_base> + $800A_H$ + (80_H × p), RCFDCnCFDTMFDCTRpHH: <RCFDCn_base> + $800B_H$ + (80_H × p)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24.89 RCFDCnCFDTMFDCTRp Register Contents

Bit Position	Bit Name	Function
31 to 16	TMPTR[15:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	TMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	TMESI	ESI 0: Error active node 1: Error passive node

Modify this register when the TMTRM bit in the corresponding RCFDCnCFDTMSTSp register is set to 0 (transmission not requested). When this register is linked to the transmit/receive FIFO buffer, do not write data to this register. When this register is allocated to the transmit queue, write data only to transmit buffer p (p = m × 32 + 31) of the corresponding channel.

TMPTR[15:0] Bits

When message transmission has been completed, the TMPTR[15:0] value is stored in the transmit history buffer.

TMFDF Bit

This bit is used to set the FD format of the message to be transmitted from the transmit buffer.

TMBRS Bit

When this bit is set to 1 while the TMFDF bit is set to 1, the data area of a transmit message is transmitted at the data bit rate. When the TMFDF bit is set to 0, write 0 to this bit.

TMESI Bit

This bit is used to set the ESI bit value of the message to be transmitted from the transmit buffer when the TMFDF bit is set to 1. The set value is transmitted when the ESIC bit in the RCFDCnCFDCmFDCFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the TMFDF bit is set to 0, write 0 to this bit.

24.3.10.6 RCFDCnCFDTMDFb_p — Transmit Buffer Data Field Register (b = 0 to 15, p = 0 to 255)

Access: RCFDCnCFDTMDFb_p register can be read or written in 32-bit units

RCFDCnCFDTMDFb_pL, RCFDCnCFDTMDFb_pH registers can be read or written in 16-bit units

RCFDCnCFDTMDFb_pLL, RCFDCnCFDTMDFb_pLH, RCFDCnCFDTMDFb_pHL, RCFDCnCFDTMDFb_pHH registers can be read or written in 8-bit units

Address: RCFDCnCFDTMDFb_p: <RCFDCn_base> + $800C_H$ + (04_H × b) + (80_H × p)

RCFDCnCFDTMDFb_pL: <RCFDCn_base> + $800C_H$ + (04_H × b) + (80_H × p),

RCFDCnCFDTMDFb_pH: <RCFDCn_base> + $800E_H$ + (04_H × b) + (80_H × p)

RCFDCnCFDTMDFb_pLL: <RCFDCn_base> + $800C_H$ + (04_H × b) + (80_H × p),

RCFDCnCFDTMDFb_pLH: <RCFDCn_base> + $800D_H$ + (04_H × b) + (80_H × p),

RCFDCnCFDTMDFb_pHL: <RCFDCn_base> + $800E_H$ + (04_H × b) + (80_H × p),

RCFDCnCFDTMDFb_pHH: <RCFDCn_base> + $800F_H$ + (04_H × b) + (80_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB4 × b + 3 [7:0]								TMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB4 × b + 1 [7:0]								TMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.90 RCFDCnCFDTMDFb_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB4 × b + 3 [7:0]	Transmit Buffer Data Byte 4 × b + 3 Transmit Buffer Data Byte 4 × b + 2 Transmit Buffer Data Byte 4 × b + 1 Transmit Buffer Data Byte 4 × b + 0 Set the transmit buffer data.
23 to 16	TMDB4 × b + 2 [7:0]	
15 to 8	TMDB4 × b + 1 [7:0]	
7 to 0	TMDB4 × b + 0 [7:0]	

Modify this register when the TMTRM bit in the corresponding RCFDCnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 32 + 31) for the corresponding channel.

24.3.10.7 RCFDCnCFDTMIECm — Transmit Buffer Interrupt Enable Configuration Register (m = 0 to 7)

Access: RCFDCnCFDTMIECm register can be read or written in 32-bit units

RCFDCnCFDTMIECmL, RCFDCnCFDTMIECmH registers can be read or written in 16-bit units

RCFDCnCFDTMIECmLL, RCFDCnCFDTMIECmLH, RCFDCnCFDTMIECmHL, RCFDCnCFDTMIECmHH registers can be read or written in 8-bit units

Address: RCFDCnCFDTMIECm: <RCFDCn_base> + $04D0_H$ + (04_H × m)

RCFDCnCFDTMIECmL: <RCFDCn_base> + $04D0_H$ + (04_H × m),

RCFDCnCFDTMIECmH: <RCFDCn_base> + $04D2_H$ + (04_H × m)

RCFDCnCFDTMIECmLL: <RCFDCn_base> + $04D0_H$ + (04_H × m),

RCFDCnCFDTMIECmLH: <RCFDCn_base> + $04D1_H$ + (04_H × m),

RCFDCnCFDTMIECmHL: <RCFDCn_base> + $04D2_H$ + (04_H × m),

RCFDCnCFDTMIECmHH: <RCFDCn_base> + $04D3_H$ + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = m × 32 + 31 to m × 32 + 16 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = m × 32 + 15 to m × 32 + 0 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.91 RCFDCnCFDTMIECm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = m × 32 + 31 to m × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = m × 32 + 15 to m × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 255)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RCFDCnCFDTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 24.92, TMIEp Bit Assignment shows the bit assignment.

Table 24.92 TMIEp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
31	0	31
32	1	0
. .	. .	. .
62	1	30
63	1	31
64	2	0
65	2	1
. .	. .	. .
95	2	31
96	3	0
. .	. .	. .
126	3	30
127	3	31
128	4	0
129	4	1
. .	. .	. .
158	4	30
159	4	31
160	5	0
161	5	1
. .	. .	. .
191	5	31
192	6	0
. .	. .	. .
222	6	30
223	6	31
224	7	0
225	7	1
. .	. .	. .
254	7	30
255	7	31

24.3.11 Details of Transmit Buffer Status-related Registers

24.3.11.1 RCFDCnCFDTMTRSTSm — Transmit Buffer Transmit Request Status Register (m = 0 to 7)

Access: RCFDCnCFDTMTRSTSm register is a read-only register that can be read in 32-bit units

RCFDCnCFDTMTRSTSmL, RCFDCnCFDTMTRSTSmH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDTMTRSTSmLL, RCFDCnCFDTMTRSTSmLH, RCFDCnCFDTMTRSTSmHL, RCFDCnCFDTMTRSTSmHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTMTRSTSm: <RCFDCn_base> + 0450_H + (04_H × m)
RCFDCnCFDTMTRSTSmL: <RCFDCn_base> + 0450_H + (04_H × m),
RCFDCnCFDTMTRSTSmH: <RCFDCn_base> + 0452_H + (04_H × m)
RCFDCnCFDTMTRSTSmLL: <RCFDCn_base> + 0450_H + (04_H × m),
RCFDCnCFDTMTRSTSmLH: <RCFDCn_base> + 0451_H + (04_H × m),
RCFDCnCFDTMTRSTSmHL: <RCFDCn_base> + 0452_H + (04_H × m),
RCFDCnCFDTMTRSTSmHH: <RCFDCn_base> + 0453_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = m × 32 + 31 to m × 32 + 16 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = m × 32 + 15 to m × 32 + 0 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.93 RCFDCnCFDTMTRSTSm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = m × 32 + 31 to m × 32 + 16) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = m × 32 + 15 to m × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 255)

These flags indicate the status of the TMTR bit in the RCFDCnCFDTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 24.94, TMTRSTSp Bit Assignment shows the bit assignment.

Table 24.94 TMTRSTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
31	0	31
32	1	0
. .	. .	. .
62	1	30
63	1	31
64	2	0
65	2	1
. .	. .	. .
95	2	31
96	3	0
. .	. .	. .
126	3	30
127	3	31
128	4	0
129	4	1
. .	. .	. .
158	4	30
159	4	31
160	5	0
161	5	1
. .	. .	. .
191	5	31
192	6	0
. .	. .	. .
222	6	30
223	6	31
224	7	0
225	7	1
. .	. .	. .
254	7	30
255	7	31

24.3.11.2 RCFDCnCFDTMTARSTSm — Transmit Buffer Transmit Abort Request Status Register (m = 0 to 7)

Access: RCFDCnCFDTMTARSTSm register is a read-only register that can be read in 32-bit units

RCFDCnCFDTMTARSTSmL, RCFDCnCFDTMTARSTSmH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDTMTARSTSmLL, RCFDCnCFDTMTARSTSmLH, RCFDCnCFDTMTARSTSmHL, RCFDCnCFDTMTARSTSmHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTMTARSTSm: <RCFDCn_base> + 0470_H + (04_H × m)

RCFDCnCFDTMTARSTSmL: <RCFDCn_base> + 0470_H + (04_H × m),

RCFDCnCFDTMTARSTSmH: <RCFDCn_base> + 0472_H + (04_H × m)

RCFDCnCFDTMTARSTSmLL: <RCFDCn_base> + 0470_H + (04_H × m),

RCFDCnCFDTMTARSTSmLH: <RCFDCn_base> + 0471_H + (04_H × m),

RCFDCnCFDTMTARSTSmHL: <RCFDCn_base> + 0472_H + (04_H × m),

RCFDCnCFDTMTARSTSmHH: <RCFDCn_base> + 0473_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = m × 32 + 31 to m × 32 + 16 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = m × 32 + 15 to m × 32 + 0 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.95 RCFDCnCFDTMTARSTSm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = m × 32 + 31 to m × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = m × 32 + 15 to m × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 255)

These flags indicate the status of the TMTAR bit in the RCFDCnCFDTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 24.96, TMTARSTSp Bit Assignment shows the bit assignment.

Table 24.96 TMTARSTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
31	0	31
32	1	0
. .	. .	. .
62	1	30
63	1	31
64	2	0
65	2	1
. .	. .	. .
95	2	31
96	3	0
. .	. .	. .
126	3	30
127	3	31
128	4	0
129	4	1
. .	. .	. .
158	4	30
159	4	31
160	5	0
161	5	1
. .	. .	. .
191	5	31
192	6	0
. .	. .	. .
222	6	30
223	6	31
224	7	0
225	7	1
. .	. .	. .
254	7	30
255	7	31

24.3.11.3 RCFDCnCFDTMTCSTSm — Transmit Buffer Transmit Complete Status Register (m = 0 to 7)

Access: RCFDCnCFDTMTCSTSm register is a read-only register that can be read in 32-bit units

RCFDCnCFDTMTCSTSmL, RCFDCnCFDTMTCSTSmH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDTMTCSTSmLL, RCFDCnCFDTMTCSTSmLH, RCFDCnCFDTMTCSTSmHL, RCFDCnCFDTMTCSTSmHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTMTCSTSm: <RCFDCn_base> + 0490_H + (04_H × m)
RCFDCnCFDTMTCSTSmL: <RCFDCn_base> + 0490_H + (04_H × m),
RCFDCnCFDTMTCSTSmH: <RCFDCn_base> + 0492_H + (04_H × m)
RCFDCnCFDTMTCSTSmLL: <RCFDCn_base> + 0490_H + (04_H × m),
RCFDCnCFDTMTCSTSmLH: <RCFDCn_base> + 0491_H + (04_H × m),
RCFDCnCFDTMTCSTSmHL: <RCFDCn_base> + 0492_H + (04_H × m),
RCFDCnCFDTMTCSTSmHH: <RCFDCn_base> + 0493_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = m × 32 + 31 to m × 32 + 16 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = m × 32 + 15 to m × 32 + 0 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.97 RCFDCnCFDTMTCSTSm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = m × 32 + 31 to m × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = m × 32 + 15 to m × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 255)

When the TMTRF[1:0] flag in the RCFDCnCFDTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

To clear the TMTCSTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 24.98, TMTCSTSp Bit Assignment shows the bit assignment.

Table 24.98 TMTCSTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
31	0	31
32	1	0
. .	. .	. .
62	1	30
63	1	31
64	2	0
65	2	1
. .	. .	. .
95	2	31
96	3	0
. .	. .	. .
126	3	30
127	3	31
128	4	0
129	4	1
. .	. .	. .
158	4	30
159	4	31
160	5	0
161	5	1
. .	. .	. .
191	5	31
192	6	0
. .	. .	. .
222	6	30
223	6	31
224	7	0
225	7	1
. .	. .	. .
254	7	30
255	7	31

24.3.11.4 RCFDCnCFDTMTASTSm — Transmit Buffer Transmit Abort Status Register (m = 0 to 7)

Access: RCFDCnCFDTMTASTSm register is a read-only register that can be read in 32-bit units

RCFDCnCFDTMTASTSmL, RCFDCnCFDTMTASTSmH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDTMTASTSmLL, RCFDCnCFDTMTASTSmLH, RCFDCnCFDTMTASTSmHL, RCFDCnCFDTMTASTSmHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTMTASTSm: <RCFDCn_base> + $04B0_H$ + (04_H × m)
RCFDCnCFDTMTASTSmL: <RCFDCn_base> + $04B0_H$ + (04_H × m),
RCFDCnCFDTMTASTSmH: <RCFDCn_base> + $04B2_H$ + (04_H × m)
RCFDCnCFDTMTASTSmLL: <RCFDCn_base> + $04B0_H$ + (04_H × m),
RCFDCnCFDTMTASTSmLH: <RCFDCn_base> + $04B1_H$ + (04_H × m),
RCFDCnCFDTMTASTSmHL: <RCFDCn_base> + $04B2_H$ + (04_H × m),
RCFDCnCFDTMTASTSmHH: <RCFDCn_base> + $04B3_H$ + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = m × 32 + 31 to m × 32 × 16 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = m × 32 + 15 to m × 32 + 0 (m = 0 to 7))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.99 RCFDCnCFDTMTASTSm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = m × 32 + 31 to m × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = m × 32 + 15 to m × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 255)

When the TMTRF[1:0] flag in the RCFDCnCFDTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

To clear the TMTASTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 24.100, TMTASTSp Bit Assignment shows the bit assignment.

Table 24.100 TMTASTSp Bit Assignment

Bit Position	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
31	0	31
32	1	0
. .	. .	. .
62	1	30
63	1	31
64	2	0
65	2	1
. .	. .	. .
95	2	31
96	3	0
. .	. .	. .
126	3	30
127	3	31
128	4	0
129	4	1
. .	. .	. .
158	4	30
159	4	31
160	5	0
161	5	1
. .	. .	. .
191	5	31
192	6	0
. .	. .	. .
222	6	30
223	6	31
224	7	0
225	7	1
. .	. .	. .
254	7	30
255	7	31

24.3.12 Details of Transmit Queue-related Registers

24.3.12.1 RCFDCnCFDTXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 7)

Access:	RCFDCnCFDTXQCCm register can be read or written in 32-bit units RCFDCnCFDTXQCCmL register can be read or written in 16-bit units RCFDCnCFDTXQCCmLL, RCFDCnCFDTXQCCmLH registers can be read or written in 8-bit units
Address:	RCFDCnCFDTXQCCm: <RCFDCn_base> + 0550_H + (04_H × m) RCFDCnCFDTXQCCmL: <RCFDCn_base> + 0550_H + (04_H × m) RCFDCnCFDTXQCCmLL: <RCFDCn_base> + 0550_H + (04_H × m), RCFDCnCFDTXQCCmLH: <RCFDCn_base> + 0551_H + (04_H × m)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R/W

Table 24.101 RCFDCnCFDTXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	TXQDC[4:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 31) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	TXQTXIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
4 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQDC[4:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 32 + 31) to (m × 32 + 0) (See **Table 24.30, Transmit Buffer p Allocated to the Transmit Queue of Each Channel**). For examples of how buffer allocation is done, see **Figure 24.9, Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links**.

Modify these bits only in channel reset mode.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQTXIE Bit

When the TXQTXIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQTXIE bit.

TXQE Bit

Setting this bit to 1 enables the transmit queue. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[4:0] bits to 00010_B or more.

24.3.12.2 RCFDCnCFDTXQSTSm — Transmit Queue Status Register (m = 0 to 7)

Access: RCFDCnCFDTXQSTSm register can be read or written in 32-bit units

RCFDCnCFDTXQSTSmL register can be read or written in 16-bit units

RCFDCnCFDTXQSTSmLL register can be read or written in 8-bit units

Address: RCFDCnCFDTXQSTSm: <RCFDCn_base> + 0570_H + (04_H × m)

RCFDCnCFDTXQSTSmL: <RCFDCn_base> + 0570_H + (04_H × m)

RCFDCnCFDTXQSTSmLL: <RCFDCn_base> + 0570_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W^{*1}	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.102 RCFDCnCFDTXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13 to 8	Reserved	When read, the undefined value is returned. When writing to these bits, write the value after reset.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQTXIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQTXIF Flag

The TXQTXIF flag is set to 1 when the interrupt source specified by the TXQIM bit in the RCFDCnCFDTXQCCm register has occurred.

The TXQTXIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RCFDCnCFDTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages stored in the transmit queue matches the transmit queue depth set by the TXQDC[4:0] bits in the RCFDCnCFDTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[4:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is stored in the transmit queue. This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

24.3.12.3 RCFDCnCFDTXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 7)

Access: RCFDCnCFDTXQPCTRm register is a write-only register that can be written in 32-bit units
RCFDCnCFDTXQPCTRmL register is a write-only register that can be written in 16-bit units
RCFDCnCFDTXQPCTRmLL register is a write-only register that can be written in 8-bit units

Address: RCFDCnCFDTXQPCTRm: <RCFDCn_base> + 0590_H + (04_H × m)
RCFDCnCFDTXQPCTRmL: <RCFDCn_base> + 0590_H + (04_H × m)
RCFDCnCFDTXQPCTRmLL: <RCFDCn_base> + 0590_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 24.103 RCFDCnCFDTXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request for the message. Write transmit messages to the RCFDCnCFDTMIDp, RCFDCnCFDTMPTRp, RCFDCnCFDTMFDCTRp, and RCFDCnCFDTMDFb_p registers (p = 31, 63, 95, 127, 159, 191, 223 and 255) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RCFDCnCFDTXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RCFDCnCFDTXQSTSm register is 0 (he transmit queue is not full).

24.3.13 Details of Transmit History-related Registers

24.3.13.1 RCFDCnCFDTHLCCm — Transmit History Configuration and Control Register (m = 0 to 7)

Access:	RCFDCnCFDTHLCCm register can be read or written in 32-bit units RCFDCnCFDTHLCCmL register can be read or written in 16-bit units RCFDCnCFDTHLCCmLL, RCFDCnCFDTHLCCmLH registers can be read or written in 8-bit units
Address:	RCFDCnCFDTHLCCm: <RCFDCn_base> + $05B0_H$ + (04_H × m) RCFDCnCFDTHLCCmL: <RCFDCn_base> + $05B0_H$ + (04_H × m) RCFDCnCFDTHLCCmLL: <RCFDCn_base> + $05B0_H$ + (04_H × m), RCFDCnCFDTHLCCmLH: <RCFDCn_base> + $05B1_H$ + (04_H × m)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 24.104 RCFDCnCFDTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entries from transmit/receive FIFO buffers and transmit queue 1: Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 24 items of data have been stored in the transmit history buffer 1: Each time transmit history data is stored in the transmit history buffer
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source. Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit is set to 0.

THLE Bit

Setting this bit to 1 enables the transmit history buffer. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

24.3.13.2 RCFDCnCFDTHLSTSm — Transmit History Status Register (m = 0 to 7)

Access: RCFDCnCFDTHLSTSm register can be read or written in 32-bit units
RCFDCnCFDTHLSTSmL register can be read or written in 16-bit units
RCFDCnCFDTHLSTSmLL register can be read or written in 8-bit units
RCFDCnCFDTHLSTSmLH register is a read-only register that can be read in 8-bit units

Address: RCFDCnCFDTHLSTSm: <RCFDCn_base> + $05D0_H + (04_H \times m)$
RCFDCnCFDTHLSTSmL: <RCFDCn_base> + $05D0_H + (04_H \times m)$
RCFDCnCFDTHLSTSmLL: <RCFDCn_base> + $05D0_H + (04_H \times m)$,
RCFDCnCFDTHLSTSmLH: <RCFDCn_base> + $05D1_H + (04_H \times m)$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	THLMC[5:0]						—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 24.105 RCFDCnCFDTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13 to 8	THLMC[5:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[5:0] Bits

These bits indicate the number of unread data stored in the transmit history buffer. These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RCFDCnCFDTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 32 items of data have been stored in the transmit history buffer, and is cleared to 0 when the number of data stored in the transmit history buffer has decreased to less than 32. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RCFDCnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single item of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RCFDCnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

24.3.13.3 RCFDCnCFDTHLPCTRm — Transmit History Pointer Control Register (m = 0 to 7)

Access: RCFDCnCFDTHLPCTRm register is a write-only register that can be written in 32-bit units

RCFDCnCFDTHLPCTRmL register is a write-only register that can be written in 16-bit units

RCFDCnCFDTHLPCTRmLL register is a write-only register that can be written in 8-bit units

Address: RCFDCnCFDTHLPCTRm: <RCFDCn_base> + 05F0$_H$ + (04$_H$ × m)

RCFDCnCFDTHLPCTRmL: <RCFDCn_base> + 05F0$_H$ + (04$_H$ × m)

RCFDCnCFDTHLPCTRmLL: <RCFDCn_base> + 05F0$_H$ + (04$_H$ × m)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 24.106 RCFDCnCFDTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF$_H$ to these bits moves the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF$_H$, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[5:0] (transmit history buffer unread data counter) value in the RCFDCnCFDTHLSTSm register is decremented by 1. Write FF$_H$ to the THLPC[7:0] bits after reading from the RCFDCnCFDTHLACC0m register and the RCFDCnCFDTHLACC1m register.

When writing FF$_H$ to these bits, make sure that the THLE bit in the RCFDCnCFDTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RCFDCnCFDTHLSTSm register is 0.

24.3.13.4 RCFDCnCFDTHLACC0m — Transmit History Access Register 0 (m = 0 to 7)

Access: RCFDCnCFDTHLACC0m register is a read-only register that can be read in 32-bit units

RCFDCnCFDTHLACC0mL, RCFDCnCFDTHLACC0mH registers are read-only registers that can be read in 16-bit units

RCFDCnCFDTHLACC0mLL, RCFDCnCFDTHLACC0mHL, RCFDCnCFDTHLACC0mHH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTHLACC0m: <RCFDCn_base> + 10000_H + (08_H × m)

RCFDCnCFDTHLACC0mL: <RCFDCn_base> + 10000_H + (08_H × m),

RCFDCnCFDTHLACC0mH: <RCFDCn_base> + 10002_H + (08_H × m)

RCFDCnCFDTHLACC0mLL: <RCFDCn_base> + 10000_H + (08_H × m),

RCFDCnCFDTHLACC0mHL: <RCFDCn_base> + 10002_H + (08_H × m),

RCFDCnCFDTHLACC0mHH: <RCFDCn_base> + 10003_H + (08_H × m)

Value after reset: $0000\ 0000_H$

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td></td><td colspan="16">TMTS[15:0]</td></tr>
<tr><td>Value after reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr>
</table>

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td colspan="5">BN[4:0]</td><td colspan="3">BT[2:0]</td></tr>
<tr><td>Value after reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr>
</table>

Table 24.107 RCFDCnCFDTHLACC0m Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTS[15:0]	Timestamp Data The timestamp data of stored data can be read.
15 to 8	Reserved	When read, the value after reset is returned.
7 to 3	BN[4:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TMTS[15:0] Bits

Timestamp values in transmit history data stored in the transmit history buffer are displayed.

BN[4:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

24.3.13.5 RCFDCnCFDTHLACC1m — Transmit History Access Register 1 (m = 0 to 7)

Access: RCFDCnCFDTHLACC1m register is a read-only register that can be read in 32-bit units

RCFDCnCFDTHLACC1mL register is a read-only register that can be read in 16-bit units

RCFDCnCFDTHLACC1mLL, RCFDCnCFDTHLACC1mLH registers are read-only registers that can be read in 8-bit units

Address: RCFDCnCFDTHLACC1m: <RCFDCn_base> + 10004_H + (08_H × m)

RCFDCnCFDTHLACC1mL: <RCFDCn_base> + 10004_H + (08_H × m)

RCFDCnCFDTHLACC1mLL: <RCFDCn_base> + 10004_H + (08_H × m),

RCFDCnCFDTHLACC1mLH: <RCFDCn_base> +10005_H + (08_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.108 RCFDCnCFDTHLACC1m Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15 to 0	TID[15:0]	Label Data The label information of stored data can be read.

TID[15:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

24.3.14 Details of Test-related Registers

24.3.14.1 RCFDCnCFDGTSTCFG — Global Test Configuration Register

Access: RCFDCnCFDGTSTCFG register can be read or written in 32-bit units

RCFDCnCFDGTSTCFGL, RCFDCnCFDGTSTCFGH registers can be read or written in 16-bit units

RCFDCnCFDGTSTCFGLL, RCFDCnCFDGTSTCFGHL, RCFDCnCFDGTSTCFGHH registers can be read or written in 8-bit units

Address: RCFDCnCFDGTSTCFG: <RCFDCn_base> + 0618_H

RCFDCnCFDGTSTCFGL: <RCFDCn_base> + 0618_H,

RCFDCnCFDGTSTCFGH: <RCFDCn_base> + $061A_H$

RCFDCnCFDGTSTCFGLL: <RCFDCn_base> + 0618_H,

RCFDCnCFDGTSTCFGHL: <RCFDCn_base> + $061A_H$,

RCFDCnCFDGTSTCFGHH: <RCFDCn_base> + $061B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RTMPS[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C7ICBCE	C6ICBCE	C5ICBCE	C4ICBCE	C3ICBCE	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.109 RCFDCnCFDGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
24 to 16	RTMPS[8:0]	RAM Test Page Configuration Set a value within a range of page 0 to page 33 × (m + 1).
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	C7ICBCE	CAN7 Inter-channel Communication Test Enable 0: CAN7 inter-channel communication test is disabled. 1: CAN7 inter-channel communication test is enabled.
6	C6ICBCE	CAN6 Inter-channel Communication Test Enable 0: CAN6 inter-channel communication test is disabled. 1: CAN6 inter-channel communication test is enabled.
5	C5ICBCE	CAN5 Inter-channel Communication Test Enable 0: CAN5 inter-channel communication test is disabled. 1: CAN5 inter-channel communication test is enabled.
4	C4ICBCE	CAN4 Inter-channel Communication Test Enable 0: CAN4 inter-channel communication test is disabled. 1: CAN4 inter-channel communication test is enabled.

Table 24.109 RCFDCnCFDGTSTCFG Register Contents

Bit Position	Bit Name	Function
3	C3ICBCE	CAN3 Inter-channel Communication Test Enable 0: CAN3 inter-channel communication test is disabled. 1: CAN3 inter-channel communication test is enabled.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled. 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RCFDCnCFDGTSTCFG register only in global test mode.

RTMPS[8:0] Bits

- RCFDC0 with 8 channels

  These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 108_H, inclusive.

- RCFDC0 with 6 channels

  These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to $C6_H$, inclusive. Should not access more than 192 Bytes in the last page.

- RCFDC1 with 4 channels

  These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 84_H, inclusive. Should not access more than 128 Bytes in the last page.

C7ICBCE Bit

Setting this bit to 1 enables the channel 7 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C6ICBCE Bit

Setting this bit to 1 enables the channel 6 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C5ICBCE Bit

Setting this bit to 1 enables the channel 5 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C4ICBCE Bit

Setting this bit to 1 enables the channel 4 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C3ICBCE Bit

Setting this bit to 1 enables the channel 3 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test. This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test. This bit is cleared to 0 in global reset mode.

24.3.14.2 RCFDCnCFDGTSTCTR — Global Test Control Register

Access: RCFDCnCFDGTSTCTR register can be read or written in 32-bit units
RCFDCnCFDGTSTCTRL register can be read or written in 16-bit units
RCFDCnCFDGTSTCTRLL register can be read or written in 8-bit units

Address: RCFDCnCFDGTSTCTR: <RCFDCn_base> + $061C_H$
RCFDCnCFDGTSTCTRL: <RCFDCn_base> + $061C_H$
RCFDCnCFDGTSTCTRLL: <RCFDCn_base> + $061C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 24.110 RCFDCnCFDGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ICBCTME	Inter-channel Communication Test Enable 0: Inter-channel communication test is disabled 1: Inter-channel communication test is enabled

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode (See **Figure 24.37, RAM Test Setting Procedure**).

1. Set the GMDC[1:0] bits in the RCFDCnCFDGCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 7) in the RCFDCnCFDGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

24.3.14.3 RCFDCnCFDGLOCKK — Global Lock Key Register

Access: RCFDCnCFDGLOCKK register is a write-only register that can be written in 32-bit units.

RCFDCnCFDGLOCKKL register is a write-only register that can be written in 16-bit units.

Address: RCFDCnCFDGLOCKK: <RCFDCn_base> + $062C_H$

RCFDCnCFDGLOCKKL: <RCFDCn_base> + $062C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}	W^{*1}

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 24.111 RCFDCnCFDGLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RCFDCnCFDGLOCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 24.10.4.2, Procedure for Releasing the Protection**.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RCFDCnCFDGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RCFDCn_base> + 0000_H to <RCFDCn_base> + $07FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

24.3.14.4 RCFDCnCFDRPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access:	RCFDCnCFDRPGACCr register can be read or written in 32-bit units RCFDCnCFDRPGACCrL, RCFDCnCFDRPGACCrH registers can be read or written in 16-bit units RCFDCnCFDRPGACCrLL, RCFDCnCFDRPGACCrLH, RCFDCnCFDRPGACCrHL, RCFDCnCFDRPGACCrHH registers can be read or written in 8-bit units
Address:	RCFDCnCFDRPGACCr: <RCFDCn_base> + 10400_H + (04_H × r) RCFDCnCFDRPGACCrL: <RCFDCn_base> + 10400_H + (04_H × r), RCFDCnCFDRPGACCrH: <RCFDCn_base> + 10402_H + (04_H × r) RCFDCnCFDRPGACCrLL: <RCFDCn_base> + 10400_H + (04_H × r), RCFDCnCFDRPGACCrLH: <RCFDCn_base> + 10401_H + (04_H × r), RCFDCnCFDRPGACCrHL: <RCFDCn_base> + 10402_H + (04_H × r), RCFDCnCFDRPGACCrHH: <RCFDCn_base> + 10403_H + (04_H × r)
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.112 RCFDCnCFDRPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA[31:0]	RAM Data Test Access RAM data for CAN can be read and written.

Modify the RCFDCnCFDRPGACCr register in global test mode with the RTME bit in the RCFDCnCFDGTSTCTR register set to 1 (RAM test is enabled).

The RCFDCnCFDRPGACCr register can be read and written when the RTME bit is set to 1.

24.4 Interrupt Sources and DMA Trigger

24.4.1 Interrupt Sources

The RS-CANFD module has 26 interrupts for unit 0 and 14 interrupts for unit 1 are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources/unit): Receive FIFO interrupt Global error interrupt
- Channel interrupts (3 sources/channel): (m = 0 to 7 (n = 0), m = 0 to 3 (n = 1))

1. CANm transmit interrupt
    - CANm transmit complete interrupt
    - CANm transmit abort interrupt
    - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
    - CANm transmit history interrupt
    - CANm transmit queue Interrupt
2. CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)
3. CANm error interrupt

CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In this case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CANFD module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 24.113, List of CAN Interrupt Sources lists the CAN interrupt sources. **Figure 24.2, CAN Global Interrupt Block Diagram** shows the CAN global interrupt block diagram. **Figure 24.3, CAN Channel Interrupt Block Diagram** shows the CAN channel interrupt block diagram.

Table 24.113 List of CAN Interrupt Sources

	Interrupt Source		Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RCFDCnCFDRFSTS0 register	RFIE in the RCFDCnCFDRFCC0 register
		Receive FIFO 1	RFIF in the RCFDCnCFDRFSTS1 register	RFIE in the RCFDCnCFDRFCC1 register
		Receive FIFO 2	RFIF in the RCFDCnCFDRFSTS2 register	RFIE in the RCFDCnCFDRFCC2 register
		Receive FIFO 3	RFIF in the RCFDCnCFDRFSTS3 register	RFIE in the RCFDCnCFDRFCC3 register
		Receive FIFO 4	RFIF in the RCFDCnCFDRFSTS4 register	RFIE in the RCFDCnCFDRFCC4 register
		Receive FIFO 5	RFIF in the RCFDCnCFDRFSTS5 register	RFIE in the RCFDCnCFDRFCC5 register
		Receive FIFO 6	RFIF in the RCFDCnCFDRFSTS6 register	RFIE in the RCFDCnCFDRFCC6 register
		Receive FIFO 7	RFIF in the RCFDCnCFDRFSTS7 register	RFIE in the RCFDCnCFDRFCC7 register
	Global error		DEF in the RCFDCnCFDGERFL register	DEIE in the RCFDCnCFDGCTR register
			MES in the RCFDCnCFDGERFL register	MEIE in the RCFDCnCFDGCTR register
			THLES in the RCFDCnCFDGERFL register	THLEIE in the RCFDCnCFDGCTR register
			CMPOF in the RCFDCnCFDGERFL register	CMPOFIE in the RCFDCnCFDGCTR register
Channel interrupts (m = 0 to 7 (n = 0), m = 0 to 3 (n = 1))	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RCFDCnCFDTMSTSp register	TMIEp in the RCFDCnCFDTMIECm register
		CANm transmit abort	TMTRF[1:0] in the RCFDCnCFDTMSTSp register	TAIE in the RCFDCnCFDCmCTR register
		CANm transmit/ receive FIFO transmit complete	CFTXIF in the RCFDCnCFDCFSTSk register	CFTXIE in the RCFDCnCFDCFCCk register
		CANm transmit queue	TXQTXIF in the RCFDCnCFDTXQSTSm register	TXQTXIE in the RCFDCnCFDTXQCCm register
		CANm transmit history	THLIF in the RCFDCnCFDTHLSTSm register	THLIE in the RCFDCnCFDTHLCCm register
	CANm transmit/receive FIFO receive complete		CFRXIF in the RCFDCnCFDCFSTSk register	CFRXIE in the RCFDCnCFDCFCCk register
	CANm error		BEF in the RCFDCnCFDCmERFL register	BEIE in the RCFDCnCFDCmCTR register
			ALF in the RCFDCnCFDCmERFL register	ALIE in the RCFDCnCFDCmCTR register
			BLF in the RCFDCnCFDCmERFL register	BLIE in the RCFDCnCFDCmCTR register
			OVLF in the RCFDCnCFDCmERFL register	OLIE in the RCFDCnCFDCmCTR register
			BORF in the RCFDCnCFDCmERFL register	BORIE in the RCFDCnCFDCmCTR register
			BOEF in the RCFDCnCFDCmERFL register	BOEIE in the RCFDCnCFDCmCTR register
			EPF in the RCFDCnCFDCmERFL register	EPIE in the RCFDCnCFDCmCTR register
			EWF in the RCFDCnCFDCmERFL register	EWIE in the RCFDCnCFDCmCTR register
			SOCO in the RCFDCnCFDCmFDSTS register	SOCOIE in the RCFDCnCFDCmCTR register
			EOCO in the RCFDCnCFDCmFDSTS register	EOCOIE in the RCFDCnCFDCmCTR register
			TDCVF in the RCFDCnCFDCmFDSTS register	TDCVFIE in the RCFDCnCFDCmCTR register

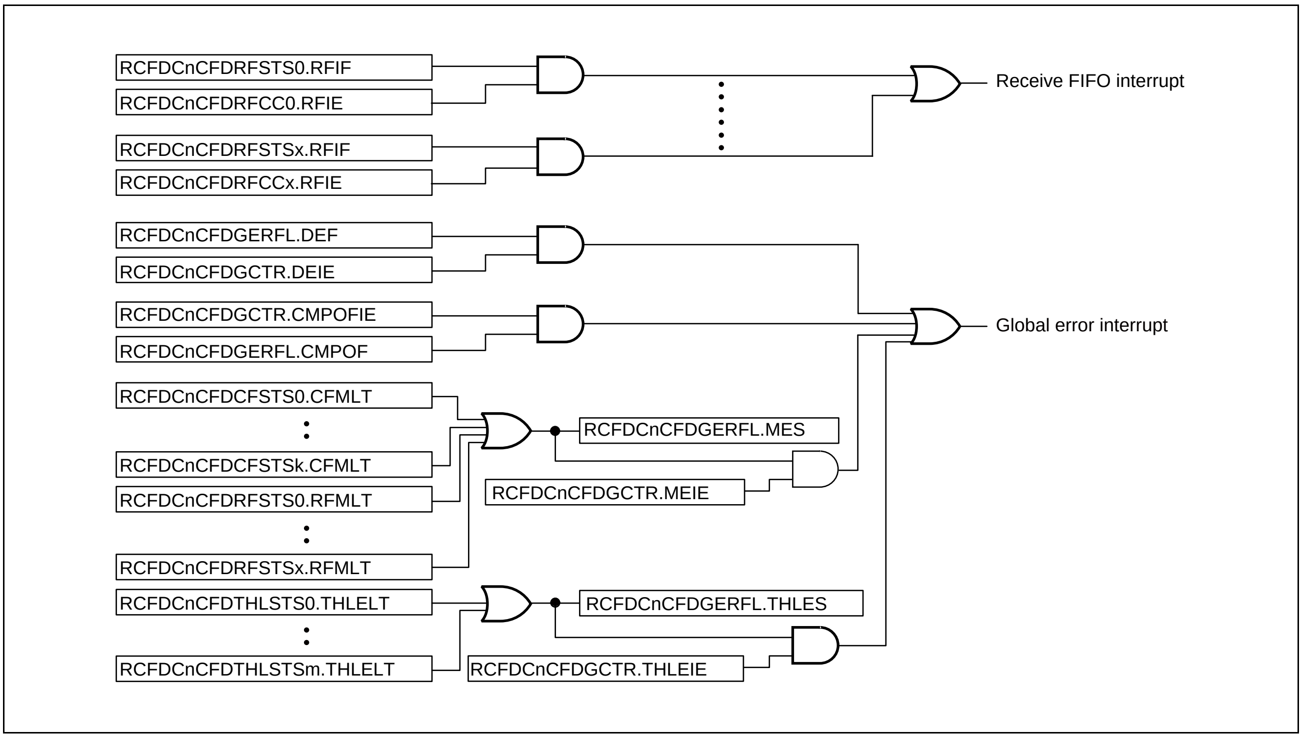


Figure 24.2 CAN Global Interrupt Block Diagram

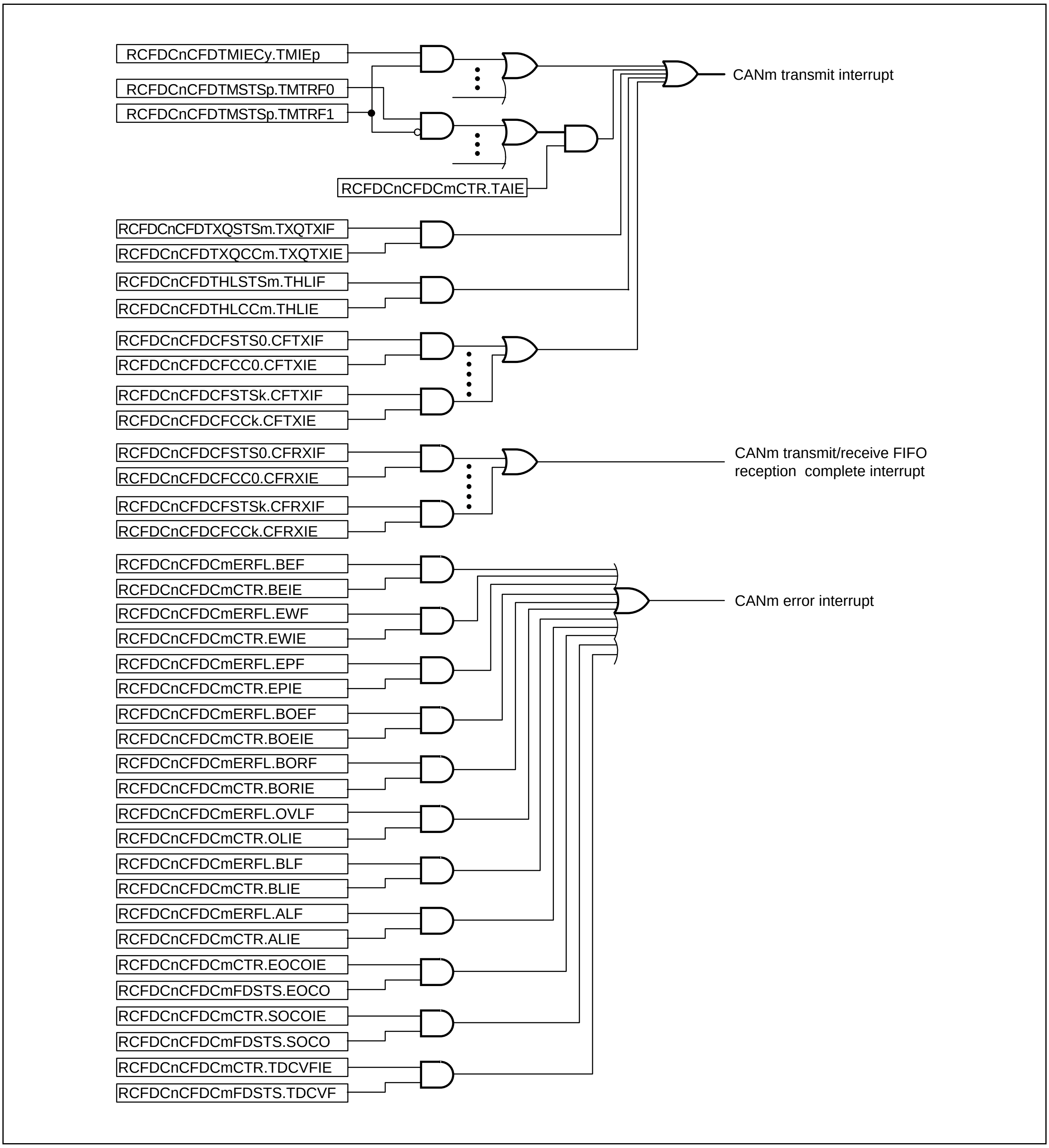


Figure 24.3 CAN Channel Interrupt Block Diagram

24.4.2 DMA Trigger

FIFO buffers used for reception can be related to DMA channels. The following 16 FIFO buffers can be related.

- All receive FIFO buffers x (x = 0 to 7)
- The first transmit/receive FIFO buffer k (k = 3 × m, m = 0 to 7) allocated to channel m

When the DMA enable bit (RFDMAEx or CFDMAEm bit in the RCFDCnCFDCDTCT register) is set to 1 and an unread message is remaining in the related FIFO, a DMA transfer request trigger is generated.

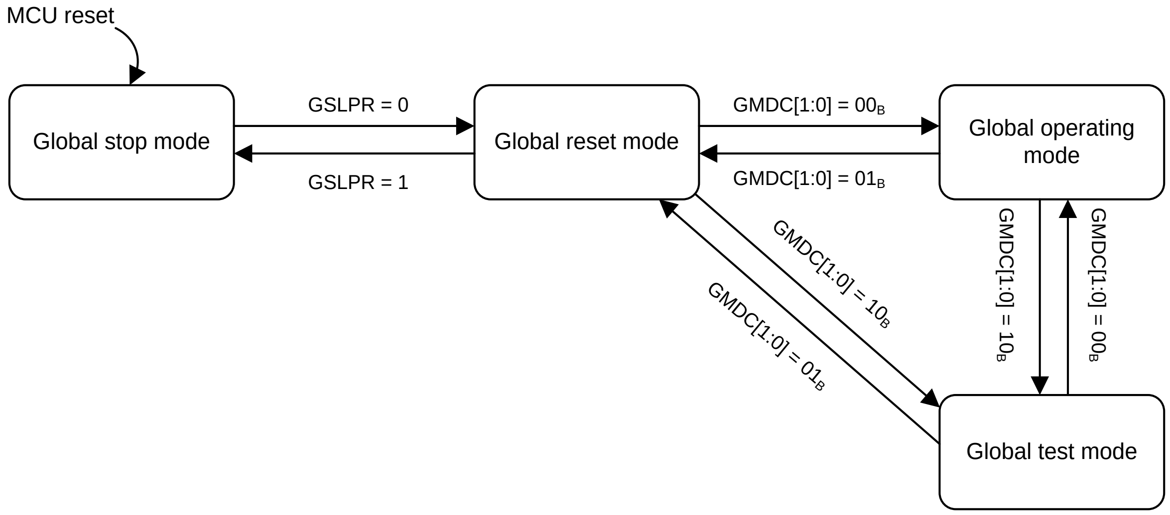
24.5 CAN Modes

The RS-CANFD module has four global modes to control the entire RS-CANFD module status and four channel modes to control individual channel status. Details of global modes are described in **Section 24.5.1, Global Modes**, and details of channel modes are described in **Section 24.5.2, Channel Modes**.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

24.5.1 Global Modes

Figure 24.4, Transitions of Global Modes shows the transitions of global modes.



Note: GSLPR, GMDC[1:0]: Bits in the RCFDCnCFDGCTR register

Figure 24.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 24.114, Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 24.114 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

	Channel Mode after Setting			
Channel Mode before Setting	GMDC[1:0] = 00_B GSLPR = 0 (Global Operation)	GMDC[1:0] = 10_B GSLPR = 0 (Global Test)	GMDC[1:0] = 01_B GSLPR = 0 (Global Reset)	GMDC[1:0] = 01_B GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RCFDCnCFDGCTR register

Table 24.115, Global Mode Transition Time shows the global mode transition time.

Table 24.115 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Two CAN bit times*1,*2
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Two CAN bit times*1,*2
Global operating	Global test	Three CAN frames*1

Note 1. CAN frame time and CAN bit time of the lowest communication speed of the channels in use

Note 2. This time value is the CAN bit time of the nominal bit rate.

24.5.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RCFDCnCFDGCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each RCFDCnCFDCmCTR register to 1 (channel stop mode). Afterwards, if all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

24.5.1.2 Global Reset Mode

In global reset mode, RS-CANFD module settings are performed. When the RS-CANFD module transitions to global reset mode, some registers are initialized. For registers to be initialized, see **Table 24.118, Registers Initialized in Global Reset Mode or Channel Reset Mode** and **Table 24.119, Registers Initialized Only in Global Reset Mode**.

Setting the GMDC[1:0] bits in the RCFDCnCFDGCTR register to 01_B sets the CHMDC[1:0] bits in each RCFDCnCFDCmCTR registers (m = 0 to 7) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode.

Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

24.5.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RCFDCnCFDGCTR register to 10_B sets the CHMDC[1:0] bits in each RCFDCnCFDCmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

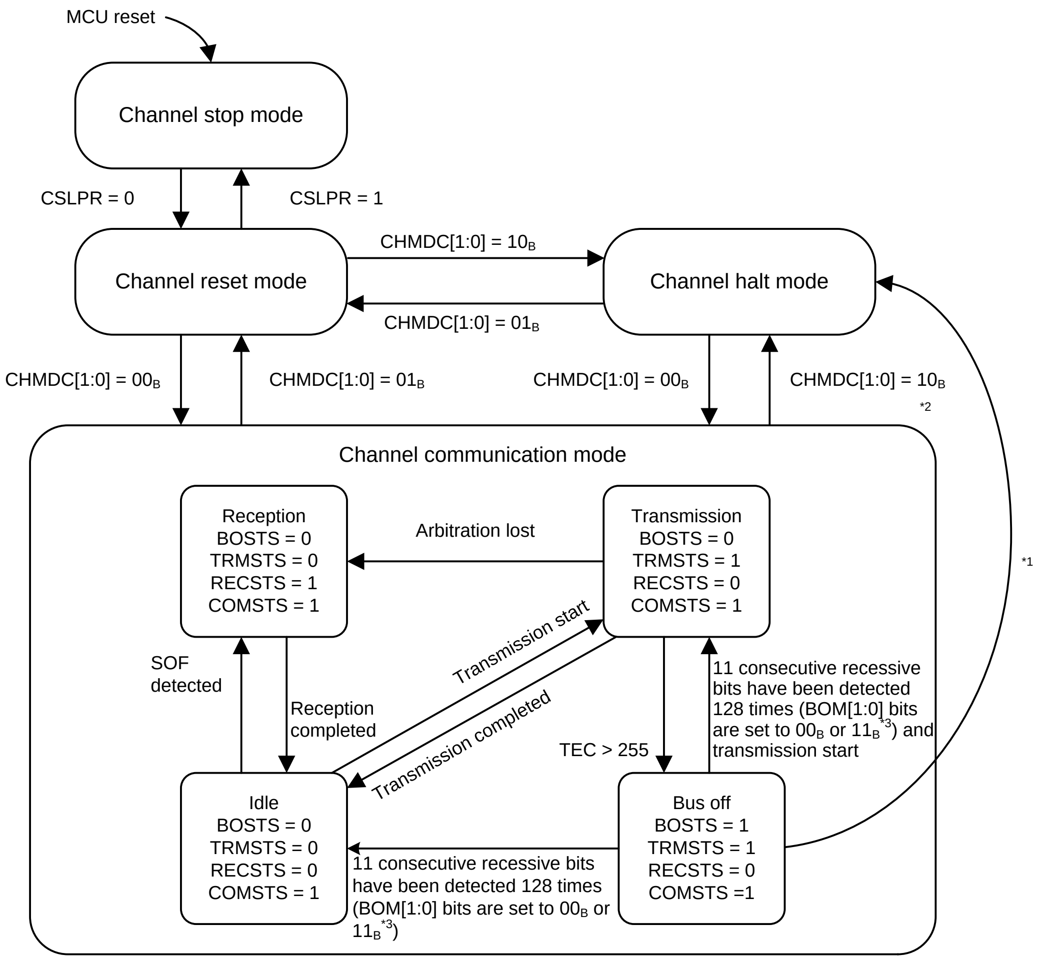
24.5.1.4 Global Operating Mode

The RS-CANFD module operates in global operating mode.

When the GMDC[1:0] bits in the RCFDCnCFDGCTR register are set to 00_B, the RS-CANFD module transitions to global operating mode.

24.5.2 Channel Modes

Figure 24.5, Channel Mode State Transition Chart shows a channel mode state transition chart. **Table 24.116, Channel Mode Transition Time** shows the channel mode transition time.



Note: CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RCFDCnCFDCmCTR register (m = 0 to 7) BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RCFDCnCFDCmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Note 3. When 11 consecutive recessive bits are detected 128 times before the CHMDC[1:0] bits are set to 10_B.

Figure 24.5 Channel Mode State Transition Chart

Table 24.116 Channel Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times*1
Channel reset	Channel communication	Four CANm bit times*1
Channel halt	Channel reset	Two CANm bit times*1
Channel halt	Channel communication	Four CANm bit times*1
Channel communication	Channel reset	Two CANm bit times*1
Channel communication	Channel halt	Two CANm frames

Note 1. This time value is the CANm bit time of the nominal bit rate.

24.5.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. Channel-related registers can be read, but writing data to them is prohibited (except write to CSLPR bit). Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RCFDCnCFDCmCTR register (m = 0 to 7) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

24.5.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. For registers to be initialized, see **Table 24.118, Registers Initialized in Global Reset Mode or Channel Reset Mode**.

When the CHMDC[1:0] bits in the RCFDCnCFDCmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 24.117, Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

24.5.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 24.117, Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 24.117 Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01_B)	Transitions to channel reset mode before reception is completed.*[1]	Transitions to channel reset mode before transmission is completed.*[1]	Transitions to channel reset mode before bus off recovery.
Channel halt*[3] (CHMDC[1:0] = 10_B)	Transitions to channel halt mode after reception is completed.*[2]	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11_B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10_B before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RCFDCnCFDCmERFL register that becomes 1 when dominant lock is detected.

Note 3. When the transition from channel reset mode to channel halt mode is to be made, set the RCFDCnCFDCmNCFG register and the RCFDCnCFDCmDCFG register, and then make a transition.

24.5.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.
- Bus off: Isolated from CAN communication.

When the CHMDC[1:0] bits in the RCFDCnCFDCmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RCFDCnCFDCmSTS register (m = 0 to 7) is set to 1

(communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

24.5.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RCFDCnCFDCmCTR register.

- When BOM[1:0] = 00_B:

  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RCFDCnCFDCmSTS register are initialized to 00_H, the BORF flag in the

  RCFDCnCFDCmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RCFDCnCFDCmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).

- When BOM[1:0] = 01_B:

  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H. The BORF flag is not set to 1, and bus off recovery interrupt request is not generated.

- When BOM[1:0] = 10_B:

  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, the BORF flag is set to 1, and a bus off recovery interrupt request is generated.

- When BOM[1:0] = 11_B:

  When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt is not generated.

  However, the BORF flag becomes 1 and a bus off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 11 consecutive recessive bits 128 times) before CHMDC[1:0] bits are set to 10_B.

If the RS-CANFD module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RCFDCnCFDCmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] value is 00_B. Writing 1 to the RTBO bit in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

24.5.3 Initializing Registers by Transition to CAN Mode

Table 24.118, Registers Initialized in Global Reset Mode or Channel Reset Mode lists bits and flags to be initialized by a transition to channel reset mode. These bits and flags are also initialized by a transition to global reset mode. Furthermore, **Table 24.119, Registers Initialized Only in Global Reset Mode** lists bits and flags to be initialized only by a transition to global reset mode.

Table 24.118 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit / Flag
RCFDCnCFDCmCTR register	ROM, CRCT, CTMS[1:0], CTME, CHMDC[1:0]
RCFDCnCFDCmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, ESIF, REC[7:0], TEC[7:0]
RCFDCnCFDCmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RCFDCnCFDCmFDCTR register	EOCCLR, SOCCLR
RCFDCnCFDCmFDSTS register	SOC[7:0], EOC[7:0], SOCO, EOCO, TDCVF, TDCR[6:0]
RCFDCnCFDCmFDCRC register	CRCREG[20:0],SCNT[3:0]
RCFDCnCFDCFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RCFDCnCFDCFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RCFDCnCFDCFTISTS register	CFkTXIF
RCFDCnCFDTMCp register	TMOM, TMTAR, TMTR
RCFDCnCFDTMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RCFDCnCFDTMTRSTSm register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RCFDCnCFDTMTARSTSm register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RCFDCnCFDTMTCSTSm register	TMTCSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RCFDCnCFDTMTASTSm register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RCFDCnCFDTXQCCm register	TXQE
RCFDCnCFDTXQSTSm register	TXQTXIF, TXQFLL, TXQEMP
RCFDCnCFDTHLCCm register	THLE
RCFDCnCFDTHLSTSm register	THLMC[5:0], THLIF, THLELT, THLFLL, THLEMP
RCFDCnCFDGTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 3)
RCFDCnCFDGTINTSTS1 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 4 to 7)

Table 24.119 Registers Initialized Only in Global Reset Mode

Register	Bit / Flag
RCFDCnCFDGSTS register	GHLTSTS
RCFDCnCFDGERFL register	EEF0, EEF1, EEF2, EEF3, EEF4, EEF5, CMPOF, THLES, MES, DEF
RCFDCnCFDGTSC register	TS[15:0]
RCFDCnCFDRMNDy register	RMNSq
RCFDCnCFDRFCCx register	RFE
RCFDCnCFDRFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RCFDCnCFDCFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RCFDCnCFDCFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RCFDCnCFDFESTS register	CFkEMP, RFxEMP
RCFDCnCFDFFSTS register	CFkFLL, RFxFLL
RCFDCnCFDFMSTS register	CFkMLT, RFxMLT
RCFDCnCFDRFISTS register	RFxIF
RCFDCnCFDCFRISTS register	CFkRXIF
RCFDCnCFDCDTCT register	CFDMAEm, RFDMAEx
RCFDCnCFDCDTSTS register	CFDMASTSm, RFDMASTSx
RCFDCnCFDGTSTCFG register	RTMPS[8:0], C0ICBCE, C1ICBCE, C2ICBCE, C3ICBCE, C4ICBCE, C5ICBCE, C6ICBCE, C7ICBCE
RCFDCnCFDGTSTCTR register	RTME, ICBCTME

24.6 Reception Functions

There are two reception types.

- Reception by receive buffers:

  Zero to 128 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.

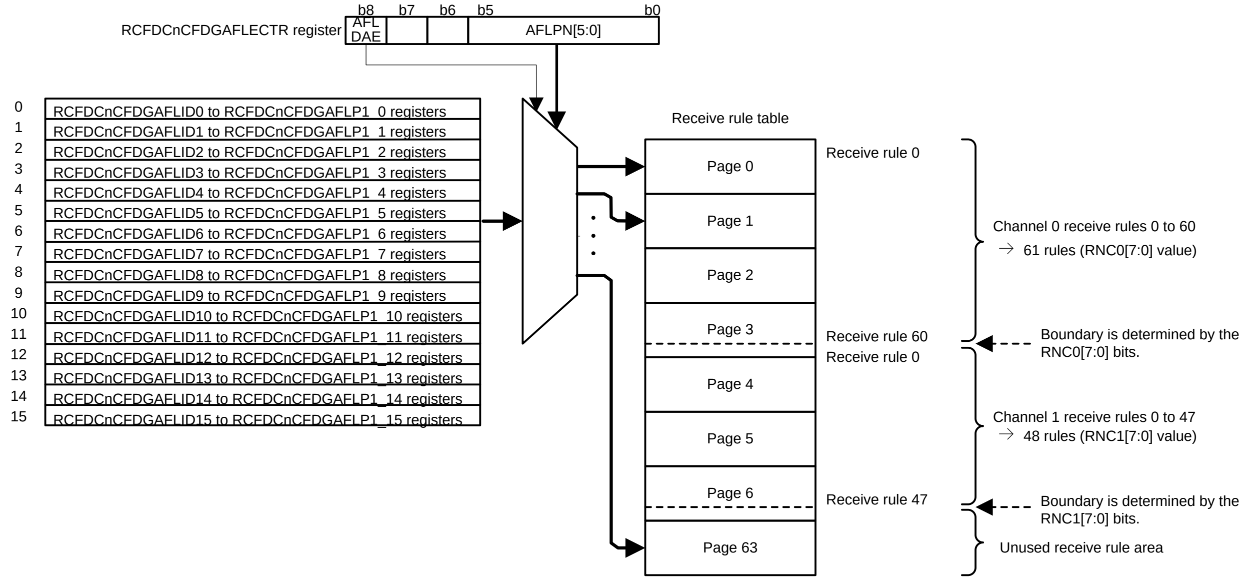
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):

  Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

24.6.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 255 receive rules can be registered per channel and up to (128 × number of channels) total receive rules can be registered in the entire module. (Up to 1024 receive rules can be registered in this module that has eight channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 24.6, Registration of Receive Rules (for Setting Channel 0 and 1)** illustrates how receive rules are registered.



Note: RNC0[7:0], RNC1[7:0]: Bits in the RCFDCnCFDGAFLCFG0 register

Figure 24.6 Registration of Receive Rules (for Setting Channel 0 and 1)

CAUTION

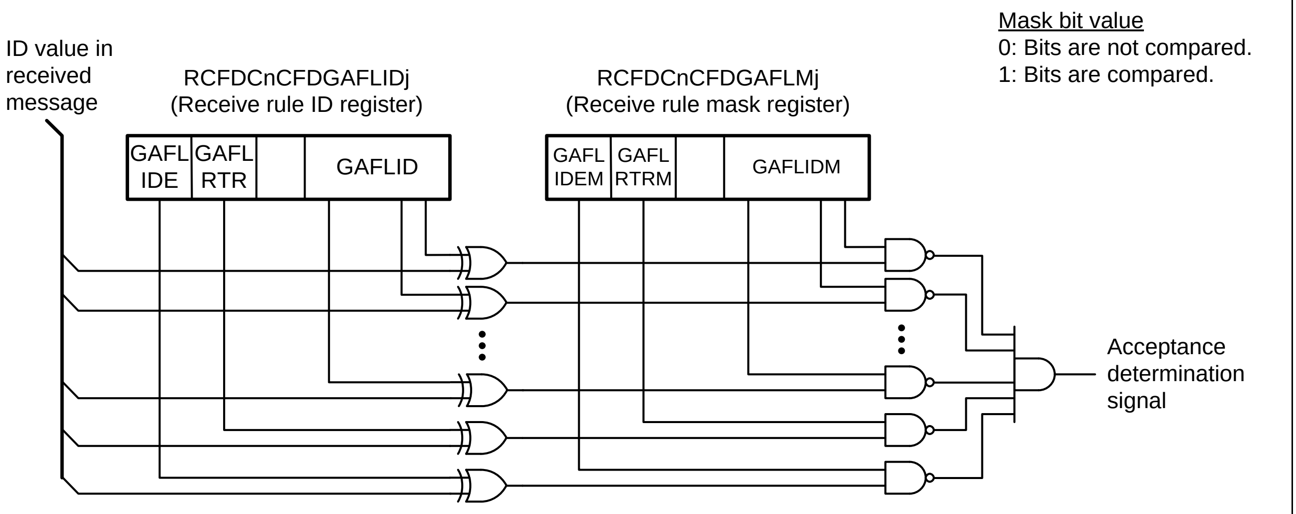
Receive rules for each channel must be set in contiguous blocks. Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RCFDCnCFDGAFLIDj, RCFDCnCFDGAFLMj, RCFDCnCFDGAFLP0_j, and RCFDCnCFDGAFLP1_j registers (j = 0 to 15). The RCFDCnCFDGAFLIDj register is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function, the RCFDCnCFDGAFLMj register is used to set mask, the RCFDCnCFDGAFLP0_j register is used to set label information to be added, DLC value, and storage receive buffer, and the RCFDCnCFDGAFLP1_j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

24.6.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RCFDCnCFDGAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule with the smallest number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.



j = 0 to 15

GAFLIDE, GAFLRTR, GAFLID: Bits in the RCFDCnCFDGAFLIDj register
GAFLIDEM, GAFLRTRM, GAFLIDM: Bits (mask bits) in the RCFDCnCFDGAFLMj register

Acceptance determination signal
0: Does not pass the acceptance filter processing.
(Not stored in the buffer)
1: Passes the acceptance filter processing.

Figure 24.7 Acceptance Filter Function

24.6.1.2 DLC Filter Processing

When the DCE bit in the RCFDCnCFDGCFG register is set to 1 (DLC check is enabled), DLC filter processing is performed for messages that pass through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RCFDCnCFDGCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RCFDCnCFDGCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RCFDCnCFDGERFL register is set to 1 (a DLC error is present).

24.6.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, and/or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RCFDCnCFDGAFLP0_j register (j = 0 to 15) and by the RCFDCnCFDGAFLP1_j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

If the payload length of the received message exceeds the payload storage size of the storage buffer, the CMPOF flag in the RCFDCnCFDGERFL register is set to 1 (payload overflow) and the processing is handled according to the CMPOC bit in the RCFDCnCFDGCFG register. When the CMPOC bit is 0, the received message which exceeds the payload storage size is not stored in the buffer. When the CMPOC bit is 1, the received message is stored in the buffer with payloads exceeding the storage size being discarded, and depending on the DRE bit in the RCFDCnCFDGCFG register the received DLC value or the DLC value of the receive rule is stored in the buffer.

24.6.1.4 Label Addition Processing

It is possible to add 16-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[15:0] bits in the RCFDCnCFDGAFLP0_j register.

24.6.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is enabled by setting the MME bit in the RCFDCnCFDGCFG register to 1 (mirror function is enabled).

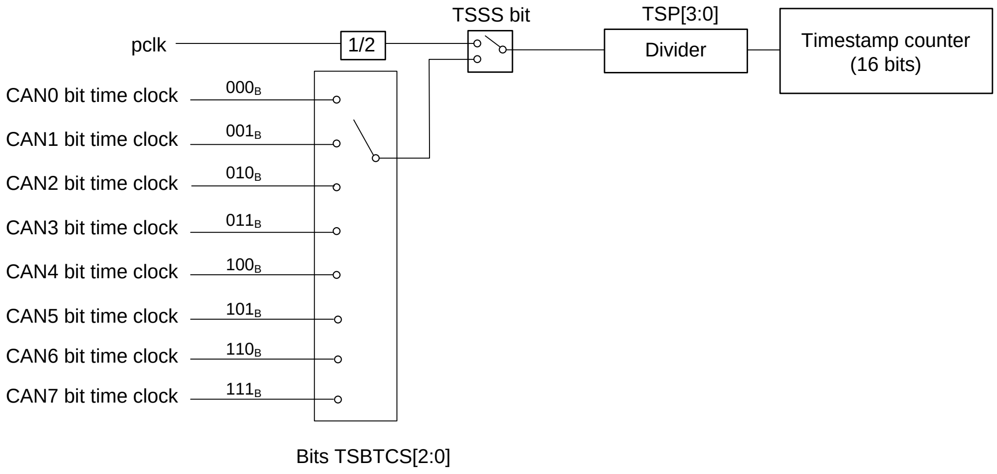
When the mirror function is in use, receive rules for which the GAFLLB bit in the RCFDCnCFDGAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

24.6.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time and transmission time. The timestamp counter value is fetched at the timing set with the TSCCFG[1:0] bits in the RCFDCnCFDGCFG register and is then stored in a receive buffer or a FIFO buffer together with the message ID and data during data reception. The clock source of the timestamp counter is selected by the TSBTCS[2:0] and TSSS bits in the RCFDCnCFDGCFG register. The clock source is selectable from pclk/2 or nominal CANm bit time clock(m = 0 to 7). However, do not select the nominal CANm bit time clock of channels that handle CAN FD frames. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RCFDCnCFDGCFG register.

When the CANm bit time clock or nominal CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the pclk/2 is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RCFDCnCFDGCTR register to 1.



Note: TSBTCS[2:0], TSSS, TSP[3:0]: Bits in the RCFDCnCFDGCFG register
When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B

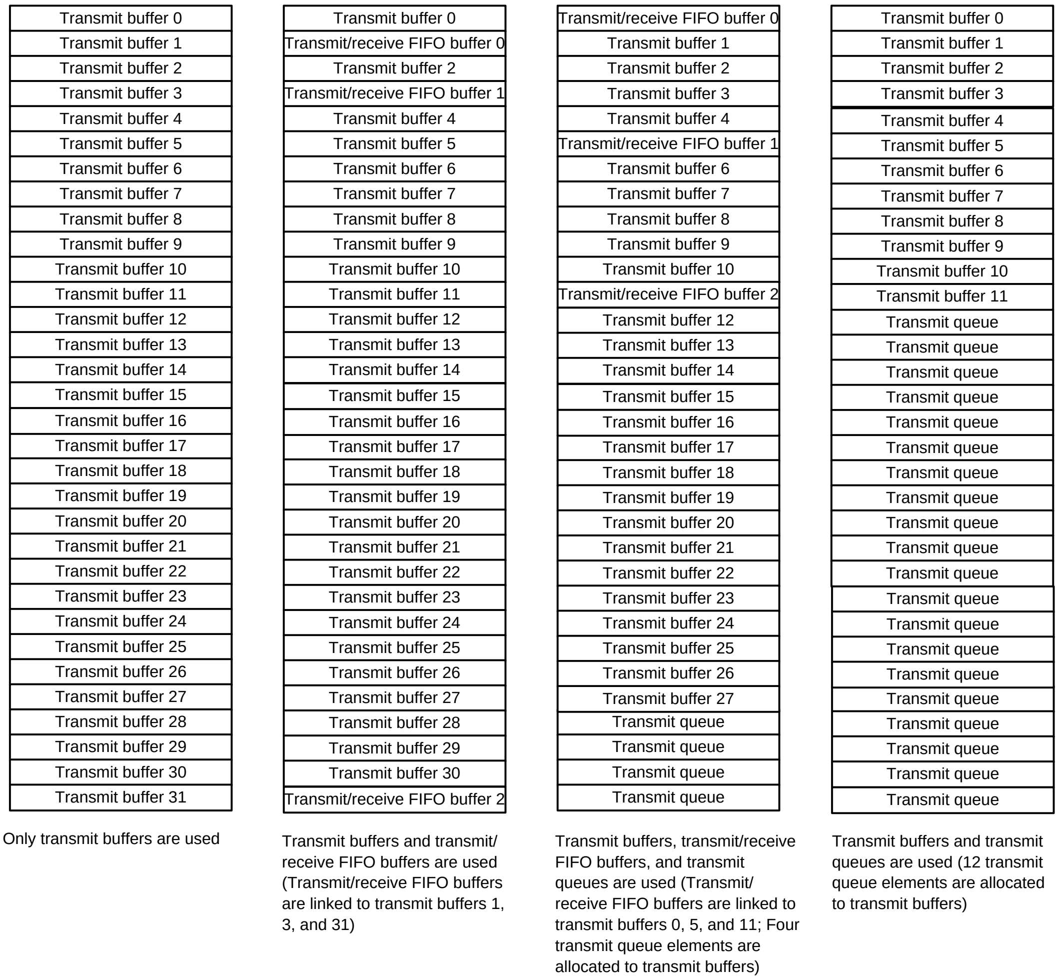
Figure 24.8 Timestamp Function Block Diagram

24.7 Transmission Functions

There are three types of transmission. The transmittable payload length is 64 bytes in every transmission types.

- Transmission using transmit buffers:
  Each channel has 32 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first- out basis.
- Transmission using transmit queues:
  Up to 32 transmit buffers per channel can be allocated to the transmit queues. Transmit buffer ((32 × m) + 31) is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 24.9, Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links shows the allocation of transmit queues and transmit/receive FIFO buffer link.



Transmit buffer 0	Transmit buffer 0	Transmit/receive FIFO buffer 0	Transmit buffer 0
Transmit buffer 1	Transmit/receive FIFO buffer 0	Transmit buffer 1	Transmit buffer 1
Transmit buffer 2	Transmit buffer 2	Transmit buffer 2	Transmit buffer 2
Transmit buffer 3	Transmit/receive FIFO buffer 1	Transmit buffer 3	Transmit buffer 3
Transmit buffer 4	Transmit buffer 4	Transmit buffer 4	Transmit buffer 4
Transmit buffer 5	Transmit buffer 5	Transmit/receive FIFO buffer 1	Transmit buffer 5
Transmit buffer 6	Transmit buffer 6	Transmit buffer 6	Transmit buffer 6
Transmit buffer 7	Transmit buffer 7	Transmit buffer 7	Transmit buffer 7
Transmit buffer 8	Transmit buffer 8	Transmit buffer 8	Transmit buffer 8
Transmit buffer 9	Transmit buffer 9	Transmit buffer 9	Transmit buffer 9
Transmit buffer 10	Transmit buffer 10	Transmit buffer 10	Transmit buffer 10
Transmit buffer 11	Transmit buffer 11	Transmit/receive FIFO buffer 2	Transmit buffer 11
Transmit buffer 12	Transmit buffer 12	Transmit buffer 12	Transmit queue
Transmit buffer 13	Transmit buffer 13	Transmit buffer 13	Transmit queue
Transmit buffer 14	Transmit buffer 14	Transmit buffer 14	Transmit queue
Transmit buffer 15	Transmit buffer 15	Transmit buffer 15	Transmit queue
Transmit buffer 16	Transmit buffer 16	Transmit buffer 16	Transmit queue
Transmit buffer 17	Transmit buffer 17	Transmit buffer 17	Transmit queue
Transmit buffer 18	Transmit buffer 18	Transmit buffer 18	Transmit queue
Transmit buffer 19	Transmit buffer 19	Transmit buffer 19	Transmit queue
Transmit buffer 20	Transmit buffer 20	Transmit buffer 20	Transmit queue
Transmit buffer 21	Transmit buffer 21	Transmit buffer 21	Transmit queue
Transmit buffer 22	Transmit buffer 22	Transmit buffer 22	Transmit queue
Transmit buffer 23	Transmit buffer 23	Transmit buffer 23	Transmit queue
Transmit buffer 24	Transmit buffer 24	Transmit buffer 24	Transmit queue
Transmit buffer 25	Transmit buffer 25	Transmit buffer 25	Transmit queue
Transmit buffer 26	Transmit buffer 26	Transmit buffer 26	Transmit queue
Transmit buffer 27	Transmit buffer 27	Transmit buffer 27	Transmit queue
Transmit buffer 28	Transmit buffer 28	Transmit queue	Transmit queue
Transmit buffer 29	Transmit buffer 29	Transmit queue	Transmit queue
Transmit buffer 30	Transmit buffer 30	Transmit queue	Transmit queue
Transmit buffer 31	Transmit/receive FIFO buffer 2	Transmit queue	Transmit queue
Only transmit buffers are used	Transmit buffers and transmit/receive FIFO buffers are used (Transmit/receive FIFO buffers are linked to transmit buffers 1, 3, and 31)	Transmit buffers, transmit/receive FIFO buffers, and transmit queues are used (Transmit/receive FIFO buffers are linked to transmit buffers 0, 5, and 11; Four transmit queue elements are allocated to transmit buffers)	Transmit buffers and transmit queues are used (12 transmit queue elements are allocated to transmit buffers)

Figure 24.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

24.7.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RCFDCnCFDGCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the smallest buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again regardless of the TPRI bit setting. When a 2-bit ECC error is detected in the priority determination processing, no message is transmitted.

24.7.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RCFDCnCFDTMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RCFDCnCFDTMSTSp register (p = 0 to 255). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

24.7.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RCFDCnCFDTMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RCFDCnCFDTMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RCFDCnCFDTMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

24.7.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RCFDCnCFDTMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RCFDCnCFDTMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] lag is set to 01_B (transmit abort has been completed).

24.7.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffer, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RCFDCnCFDCFCCk register (k = 0 to 23). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[4:0] bits in the RCFDCnCFDCFCCk register. When the CFE bit in the RCFDCnCFDCFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, arbitration-lost or the transition to channel halt mode in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

24.7.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RCFDCnCFDCFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RCFDCnCFDCFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RCFDCnCFDCFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits × 10). When the CFITR and CFITSS bits are set to $x1_B$, the CANm bit time clock becomes a count source in Classical CAN only mode and the nominal CANm bit time clock becomes a count source in CAN FD mode. (Use this count source only for the channel which does not handle the CAN FD frames in CAN FD mode.)

The interval time is calculated by the following equations where M is the value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00_B

$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times N$$

- When CFITR and CFITSS = 10_B:

$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = $x1_B$:

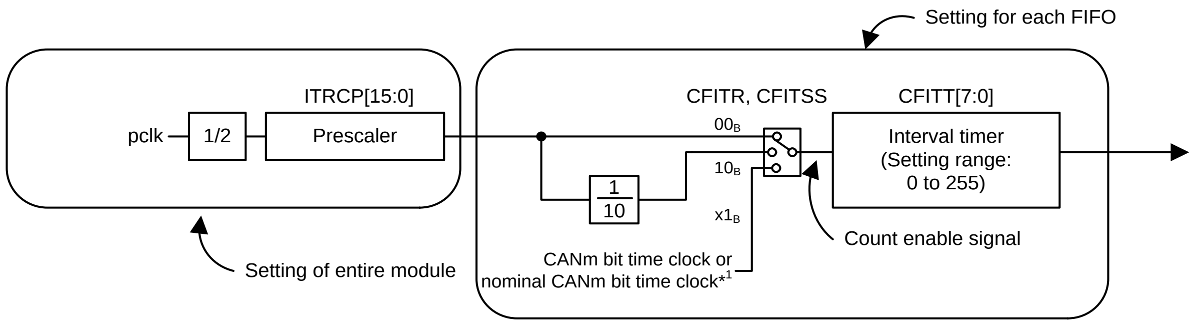
  Classical CAN only mode:

$$\frac{1}{\text{CANm bit time clock frequency}} \times N$$

  CAN FD mode and CAN FD only mode:

$$\frac{1}{\text{Nominal CANm bit time clock frequency}} \times N$$

Figure 24.10, Interval Timer Block Diagram shows the interval timer block diagram.

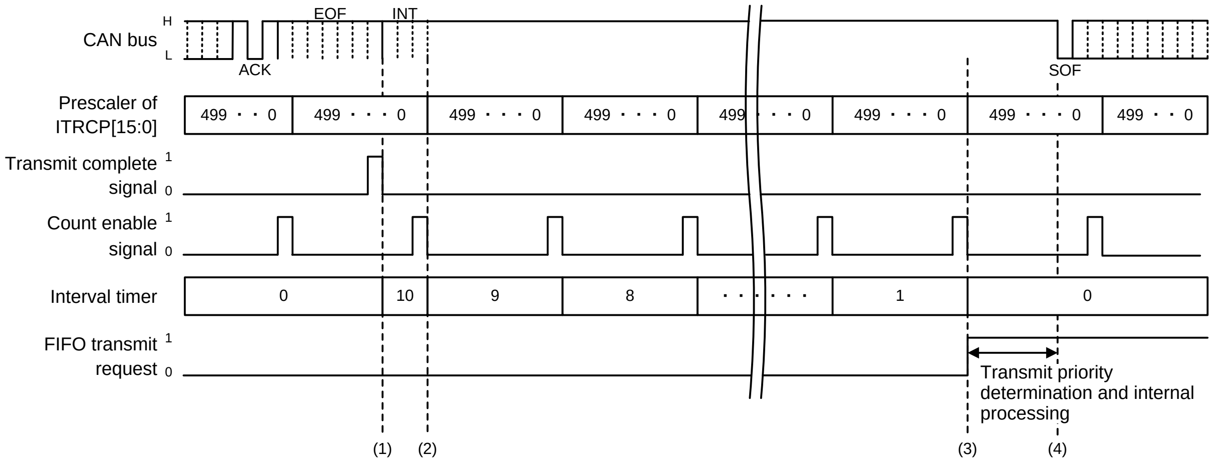


Note: ITRCP[15:0]: Bits in the RCFDCnCFDGCFG register
CFITR, CFITSS, CFITT[7:0]: Bits in the RCFDCnCFDCFCCk register
m = 0 to 7, k = 0 to 23

Note 1. Use this count source only for the channel does not handle the CAN FD frames in CAN FD mode.

Figure 24.10 Interval Timer Block Diagram

Figure 24.11, Interval Timer Timing Chart shows the interval timer timing diagram.



$$\text{Interval time (theoretical value)} = \frac{2}{\text{pclk frequency}} \times \text{(set ITRCP[15:0] value)} \times \text{set CFITT[7:0] value}$$

Note: ITRCP[15:0]: Bits in the RCFDCnCFDGCFG register (The set value is 500 in this figure.)
CFITT[7:0]: Bits in the RCFDCnCFDCFCCk register (The set value is 10 in this figure.)

Figure 24.11 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.

(2) The interval timer is decremented by 1 upon the next count enable signal.

(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.

(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 1164 pclk cycles may be generated.

24.7.4 Transmission Using Transmit Queues

3 to 32 buffers are allocated to a transmit queue for each channel, and transmit buffer ((32 × m) + 31) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RCFDCnCFDTXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RCFDCnCFDTXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, arbitration-lost, or the transition to channel halt mode when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

24.7.5 Transmit Data Padding (in CAN FD Mode and in CAN FD only Mode)

When the payload length indicated by the set DLC value in a transmit message exceeds the payload storage area size of a buffer to be used for transmission, excessive payloads are padded by CC_H.

This processing is performed in the following cases.

- Transmit/receive FIFO set to transmission or gateway mode:

When the payload length of the transmit DLC exceeds the transmit/receive FIFO payload storage area size set by the CFPLS[2:0] bits in the RCFDCnCFDCFCCk register.

This processing is also performed in FD only mode, if a Classical Frame is configured with a DLC bigger than 8.

24.7.6 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 32 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RCFDCnCFDTHLCCm register. The THLEN bit in the RCFDCnCFDCFIDk register (k = 0 to 23) and the RCFDCnCFDTMIDp register (p = 0 to 255)determines whether transmit history data is stored for each message.

A timestamp value is always included in the transmit history data..

The following information on a transmitted message will be stored in the transmit history buffer after the successful completion of transmission.

Storage of the transmit history data after the successful completion of transmission may take up to 448 pclk cycles.

- Buffer type

  001_B: Transmit buffer

  010_B: Transmit/receive FIFO buffer

  100_B: Transmit queue

- Buffer number:

  Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 24.120, Transmit History Data Buffer Numbers**.
- Label data: Label information of the transmit message
- Timestamp: Timestamp value of the transmit message

Table 24.120 Transmit History Data Buffer Numbers

Buffer Type / Buffer No.	001_B	010_B	100_B
00000_B	Transmit buffer 16 × m + 0	Buffer numbers of the transmit buffer linked to the transmit/receive FIFO buffer by the CFTML[4:0] bits in the RCFDCnCFDCFCCk register (k = 0 to 23)	Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission
00001_B	Transmit buffer 16 × m + 1		
00010_B	Transmit buffer 16 × m + 2		
00011_B	Transmit buffer 16 × m + 3		
00100_B	Transmit buffer 16 × m + 4		
00101_B	Transmit buffer 16 × m + 5		
00110_B	Transmit buffer 16 × m + 6		
00111_B	Transmit buffer 16 × m + 7		
01000_B	Transmit buffer 16 × m + 8		
01001_B	Transmit buffer 16 × m + 9		
01010_B	Transmit buffer 16 × m + 10		
01011_B	Transmit buffer 16 × m + 11		
01100_B	Transmit buffer 16 × m + 12		
01101_B	Transmit buffer 16 × m + 13		
01110_B	Transmit buffer 16 × m + 14		
01111_B	Transmit buffer 16 × m + 15		
10000_B	Transmit buffer 16 × m + 16		
10001_B	Transmit buffer 16 × m + 17		
10010_B	Transmit buffer 16 × m + 18		
10011_B	Transmit buffer 16 × m + 19		
10100_B	Transmit buffer 16 × m + 20		
10101_B	Transmit buffer 16 × m + 21		
10110_B	Transmit buffer 16 × m + 22		
10111_B	Transmit buffer 16 × m + 23		
11000_B	Transmit buffer 16 × m + 24		
11001_B	Transmit buffer 16 × m + 25		
11010_B	Transmit buffer 16 × m + 26		
11011_B	Transmit buffer 16 × m + 27		
11100_B	Transmit buffer 16 × m + 28		
11101_B	Transmit buffer 16 × m + 29		
11110_B	Transmit buffer 16 × m + 30		
11111_B	Transmit buffer 16 × m + 31		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

The timestamp value is fetched from the timestamp counter at the SOF (start of frame) timing of the message. For details about the timestamp counter, see **Section 24.6.1.6, Timestamp**.

Transmit history data can be read from the RCFDCnCFDTHLACC0m register and the RCFDCnCFDTHLACC1m register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

24.8 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive messages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RCFDCnCFDCFCCk register are set to 10_B (gateway mode) for the transmit/receive FIFO buffer selected by the RCFDCnCFDGAFLP1_j register of a channel being used for transmission, messages that pass through filter processing according to the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the

RCFDCnCFDCFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

24.8.1 CAN-CAN FD Gateway

When the gateway function is used, a frame to be transmitted can be replaced with a classical CAN frame or a CAN FD frame.

Setting the GWEN bit in the RCFDCnCFDCmFDCFG register to 1 enables the CAN-CAN FD gateway. The FDF and BRS bits in the transmit frame can be selected by the GWFDF and GWBRS bits in the RCFDCnCFDCmFDCFG register. When the DLC value of the received CAN frame is 1001_B or more and the GWFDF bit is 1 (CAN FD frame), the DLC value is replaced with 1000_B.

When the CAN-CAN FD gateway is enabled, do not perform routing for the following frames.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

24.9 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
  - Restricted operation mode
- Global tests: Performed for the entire module
  - RAM test (read/write test)
  - Inter-channel communication test [CRC error test enabled]

24.9.1 Standard Test Mode

CRC tests are enabled in standard test mode. The CRC value calculated by the RS-CANFD module based on the transmit message or receive message is stored in the register. This CRC value is stored in the CRCREG[14:0] bits in the RCFDCnCFDCmERFL register when the message is a classical CAN frame (CRC length = 15 bits) or in the CRCREG[20:0] bits in the RCFDCnCFDCmFDCRC register when the message is a CAN FD frame (CRC length = 17 or 21 bits). Use the inter-channel communication test function for CRC error tests. For details, see **Section 24.9.6.1, CRC Error Test**.

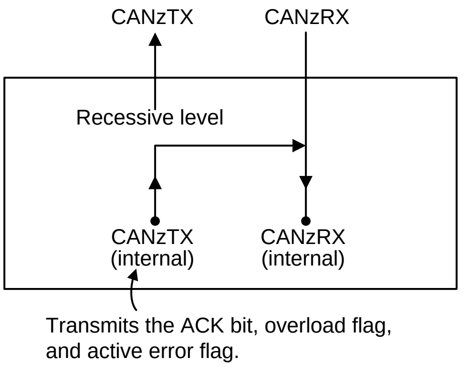
24.9.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 24.12, Connection when Listen-Only Mode is Selected shows the connection when listen-only mode is selected.



Note: z = 0 to 11

Figure 24.12 Connection when Listen-Only Mode is Selected

24.9.3 Self-Test Mode (Loopback Mode)

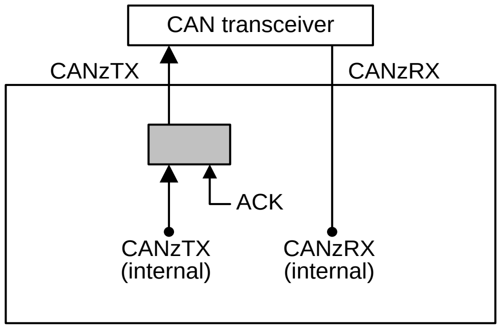
In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RCFDCnCFDGAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

24.9.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver. In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 24.13, Connection when Self-Test Mode 0 is Selected shows the connection when self-test mode 0 is selected.



Note: z = 0 to 11

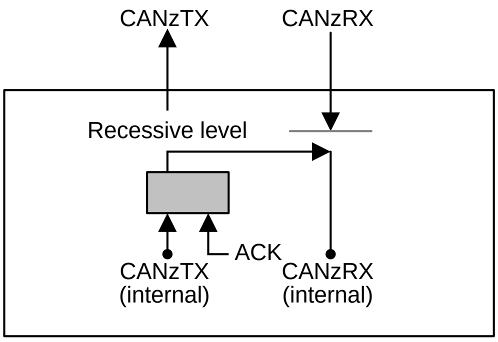
Figure 24.13 Connection when Self-Test Mode 0 is Selected

24.9.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANzTX pin (z = 0 to 11) to the internal CANzRX pin is performed. The external CANzRX pin input is isolated. The external CANzTX pin outputs only recessive bits.

Figure 24.14, Connection when Self-Test Mode 1 is Selected shows the connection when self-test mode 1 is selected.



Note: z = 0 to 11

Figure 24.14 Connection when Self-Test Mode 1 is Selected

24.9.4 Restricted Operation Mode (CAN FD Mode and CAN FD Only Mode)

In restricted operation mode, an ACK bit is generated when a valid data frame and a remote frame have been received, but these frames are not transmitted even if an error frame or an overload frame transmit condition is detected. When a condition is detected, operation is suspended until the bus idle state comes for resynchronization with the CAN communication. The receive error counter (REC) and the transmit error counter (TEC) do not change due to an error.

A desired transmission request can be made for transmission without restrictions.

24.9.5 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page are set by the RTMPS[8:0] bits in the RCFDCnCFDGTSTCFG register. Data in the set page can be read from and written to the RCFDCnCFDRPGACCr register (r = 0 to 63).

The available total RAM size:

- RCFDC0 with 8 channels: 67840 bytes (10900_H).
- RCFDC0 with 6 channels: 50880 bytes ($C6C0_H$).
- RCFDC1 with 4 channels: 33920 bytes (8480_H).

24.9.6 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel.

Set the channels not participating in test to Channel halt mode.

Figure 24.15, Connection for Inter-Channel Communication Test shows the connection for inter-channel communication test.

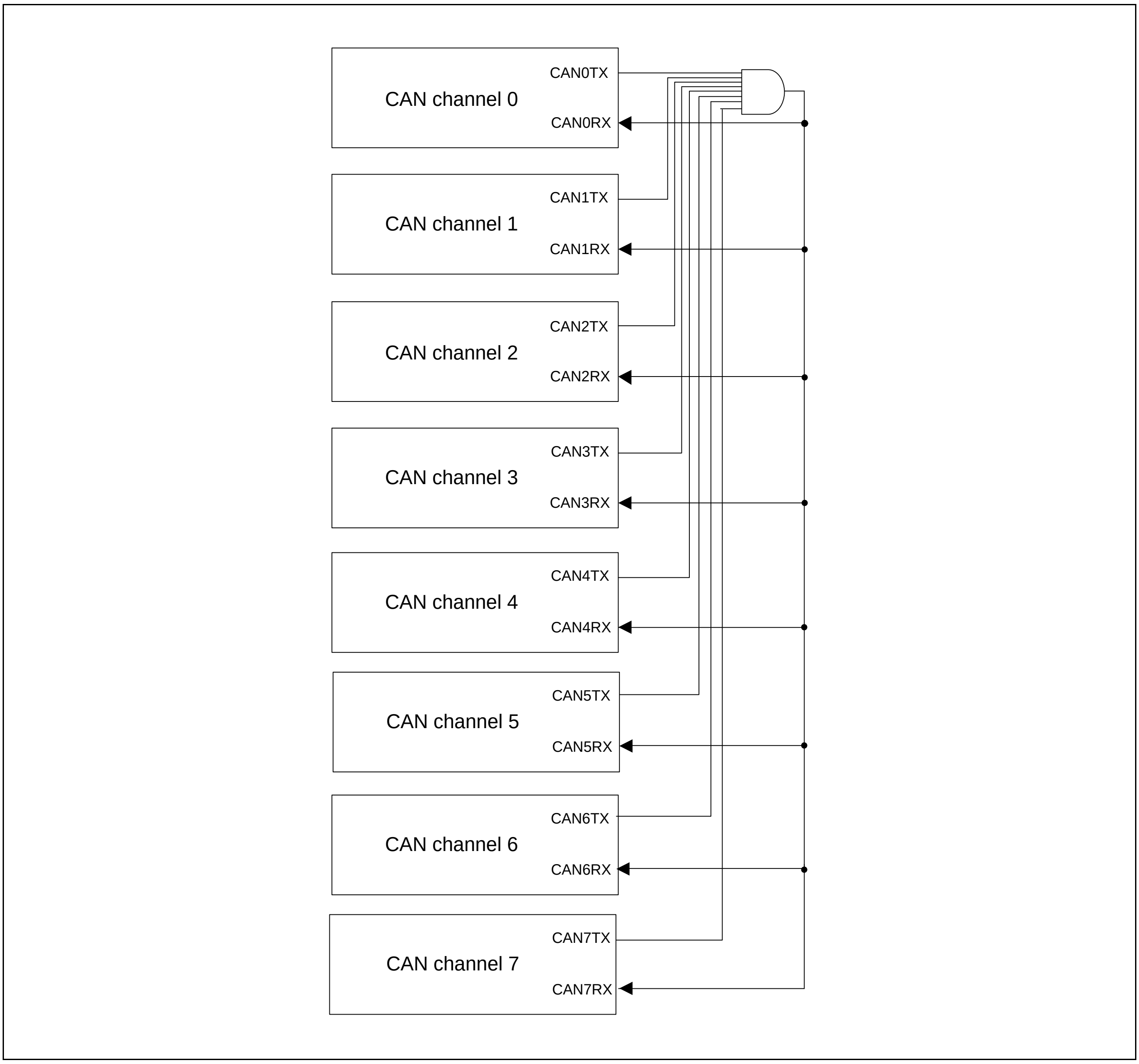


Figure 24.15 Connection for Inter-Channel Communication Test

24.9.6.1 CRC Error Test

A CRC error test is enabled during an inter-channel communication test. The following shows an example of channel 0 CRC error test procedure during a communication test between channel 0 and channel 1.

Preconditions

- Inter-channel communication test is enabled.
- Channel 0 and channel 1 are in standard test mode.

Procedure

1. Make a setting to send a message from the transmit buffer p of channel 1.
2. Set the CRCT bit in the RCFDCnCFDC0CTR register to 1 (to enable inversion of the first bit in the received ID field).
3. Set the TMTR bit in the RCFDCnCFDTMCp register to 1 (to issue a transmission request to the transmit buffer p of channel 1).
4. Wait for occurrence of a CAN0 error interrupt due to a channel bus error.
5. Read the CRCREG[14:0] bits in the RCFDCnCFDCmERFL register or the CRCREG[20:0] bits in the RCFDCnCFDCmFDCRC register of channel 0 and channel 1, and confirm that the CRC values are different on the transmission and the reception side.
6. Confirm that the CERR bit in RCFDCnCFDC0ERFL is 1 (CRC error detected).

The CRC error test function generates an incorrect CRC value by inverting the first bit in the received ID field. Therefore, note that not a CRC error but a stuff error (continuous 6-bit data of the same level) is detected when a message in which ID’s upper 5-bit value is 10000_B or ID’s upper 6-bit value is 011111_B is received.

The CRC generation circuit of the RS-CANFD module is contained in the protocol controller of each channel. Another CRC calculation test is not necessary during transmission because the same circuit is used for both transmission and reception.

24.10 RS-CANFD Setting Procedure

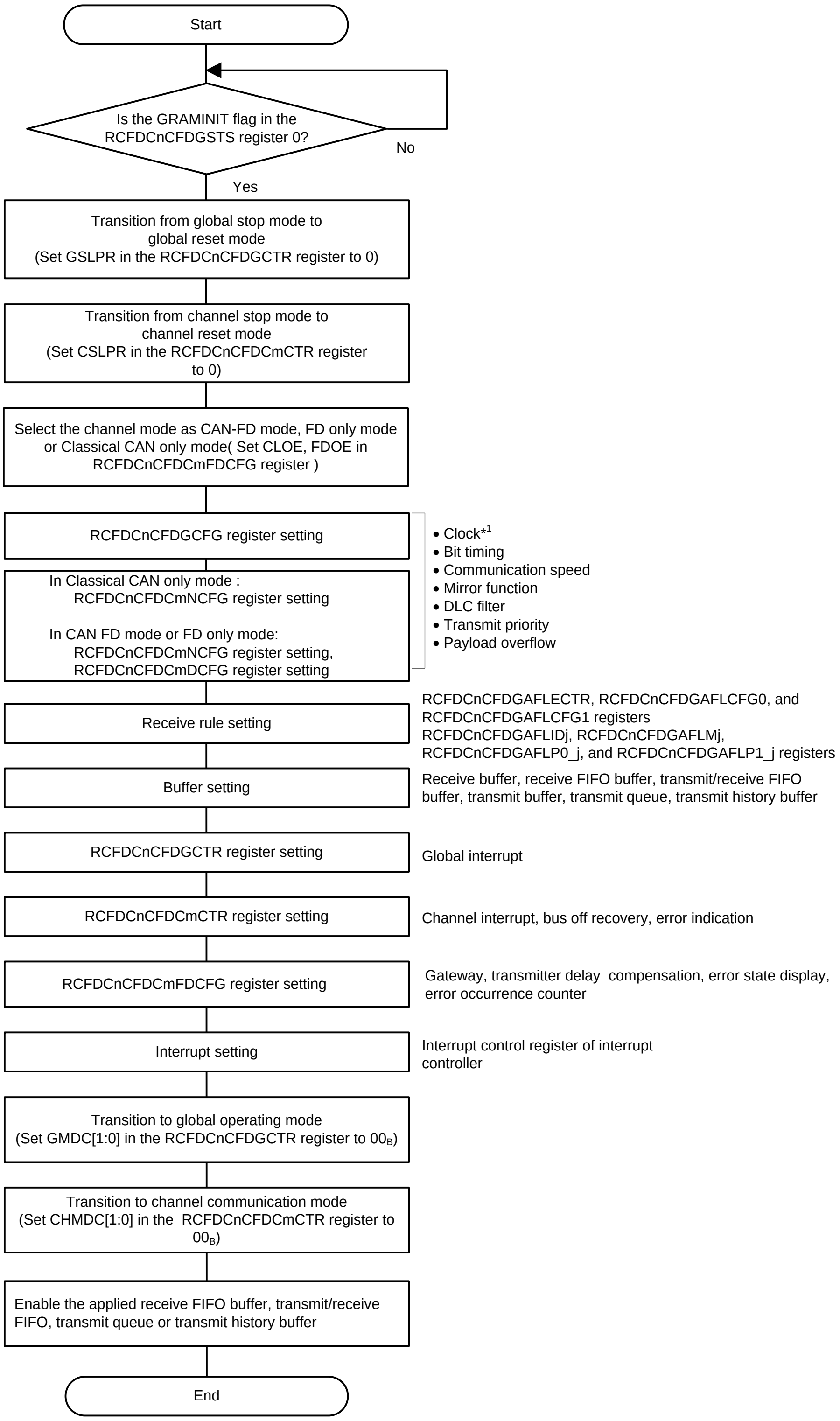
24.10.1 Initial Settings

The RS-CANFD module initializes the CAN RAM after the MCU is reset.

The RAM initialization time:

- RCFDC0 with 8 channels: 25730 pclk cycles.
- RCFDC0 with 6 channels: 19298 pclk cycles.
- RCFDC1 with 4 channels: 12866 pclk cycles.

The GRAMINIT flag in the RCFDCnCFDGSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 24.16, CAN Setting Procedure after the MCU is Reset** shows the CAN setting procedure after the MCU is reset.



Note: m = 0 to 7, j = 0 to 15

Note 1. For the setting of CAN clock frequency, see **Table 24.16, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1KH and RH850/F1KM**.

Figure 24.16 CAN Setting Procedure after the MCU is Reset

24.10.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RS-CANFD module. Select the clk_xincan or clkc using the DCS bit in the RCFDCnCFDGCFG register.

24.10.1.2 Bit Timing Setting

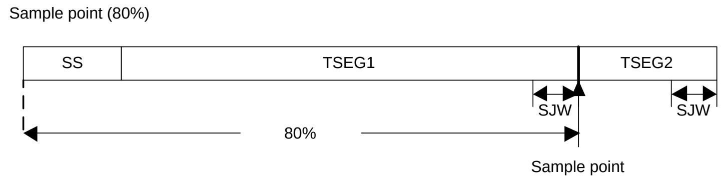
In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the corresponding registers for each channel. Two bit rates (nominal bit rate and data bit rate) are provided. Set the nominal bit rate in the RCFDCnCFDCmNCFG register and set the data bit rate in the RCFDCnCFDCmDCFG register. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). A single Tq is the cycle of clock obtained by dividing the clock selected by the DCS bit in the RCFDCnCFDGCFG register. Set a division ratio by the NBRP[9:0] bits in the RCFDCnCFDCmNCFG register and the DBRP[7:0] bits in the RCFDCnCFDCmDCFG register (CANmTq(N) clock and CANmTq(D) clock).

Be sure to specify the same values for both NBRP[9:0] and DBRP[7:0].

To specify different values for the nominal bit rate and the data bit rate, change the values of the RCFDCnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RCFDCnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RCFDCnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0] .

Figure 24.17, Bit Timing Chart shows the bit timing chart. **Table 24.121, Example of Bit Timing Settings** shows an example of bit timing setting.



Classical CAN only mode :

SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 128 Tq
Set TSEG2 to a range of 2 Tq to 32 Tq
Set SJW to a range of 1 Tq to 32 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 161 Tq
TSEG1 > TSEG2 > SJW

CAN FD mode and CAN FD only mode :
<Nominal bit rate>
SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 128 Tq
Set TSEG2 to a range of 2 Tq to 32 Tq
Set SJW to a range of 1 Tq to 32 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 161 Tq
TSEG1 > TSEG2 ≥ SJW

<Data bit rate>
SS = 1 Tq fixed
Set TSEG1 to a range of 2 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 8 Tq
Set SS + TSEG1 + TSEG2 to a range of 5 Tq to 25 Tq
TSEG1 ≥ TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space. Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width)
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 24.17 Bit Timing Chart

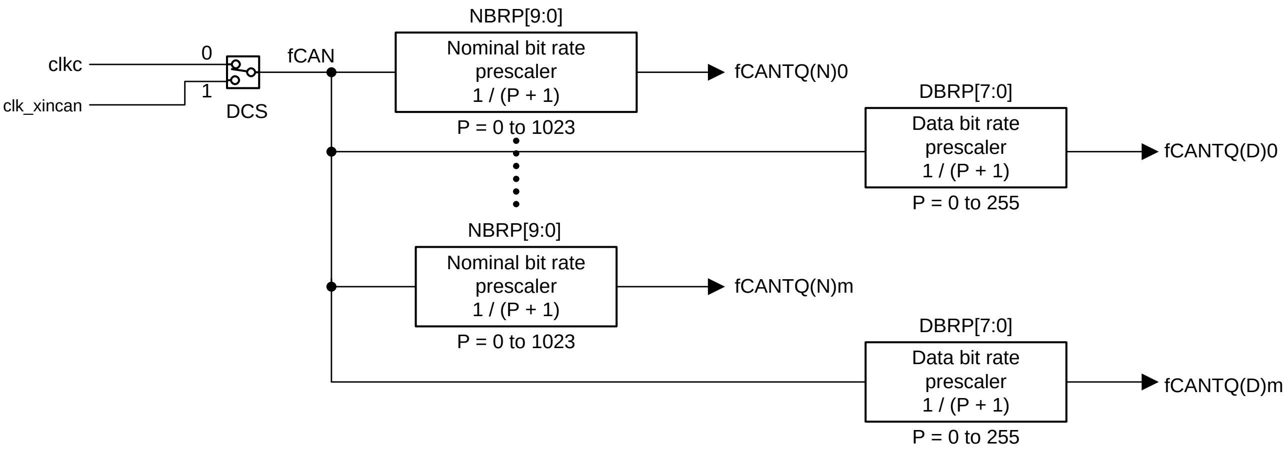
Table 24.121 Example of Bit Timing Settings

1 Bit	Set Value (Tq)				Sample Point (%) Note: See **Table 24.24, RS-CANFD Module Specifications**.
	SS	TSEG1	TSEG2	SJW	
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
50 Tq	1	39	10	4	80.00

24.10.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value, and Tq count per bit time. Set two types of transmission rate (arbitration phase and data phase) for each channel.

Figure 24.18, CAN Clock Control Block Diagram shows the CAN clock control block diagram, and **Table 24.122, Example of Communication Speed Setting (Classical CAN only mode)** shows an example of the communication speed setting.



$$\text{Nominal bit rate} = \frac{\text{fCAN}}{\text{Nominal bit rate prescaler division value} \times (\text{Tq count of 1 nominal bit time})}$$

$$\text{Data bit rate} = \frac{\text{fCAN}}{\text{Data bit rate prescaler division value} \times (\text{Tq count of 1 data bit time})}$$

Note: m = 0 to 7
DCS: Bit in the RCFDCnCFDGCFG register
NBRP[9:0]: Bits in the RCFDCnCFDCmNCFG register
DBRP[7:0]: Bits in the RCFDCnCFDCmDCFG register
fCAN: CAN clock
fCANTQm: CANmTq clock
fCANTQ(N)m: CANmTq(N) clock
fCANTQ(D)m: CANmTq(D) clock

Figure 24.18 CAN Clock Control Block Diagram

Table 24.122 Example of Communication Speed Setting (Classical CAN only mode)

fCAN / Communication Speed	40 MHz	32 MHz	24 MHz	20 MHz	16 MHz	8 MHz
1 Mbps	8 Tq (5) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (3) 12 Tq (2) 24 Tq (1)	10 Tq (2) 20 Tq (1)	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (6) 12 Tq (4) 24 Tq (2)	10 Tq (4) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (12) 12 Tq (8) 24 Tq (4)	10 Tq (8) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	8 Tq (32) 16 Tq (16)	8 Tq (24) 12 Tq (16) 24 Tq (8)	10 Tq (16) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (8) 16 Tq (4)

Table 24.123 Example of Transmission Rate Setting (Nominal Bit Rate and Data Bit Rate in CAN FD mode and CAN FD only mode)

fCAN / Communication Rate	40 MHz	20 MHz
Nominal bit rate 500 kbps Data bit rate 5 Mbps	Nominal bit rate 80 Tq (1) Data bit rate 8Tq (1)	None
Nominal bit rate 1M bps Data bit rate 2 Mbps	Nominal bit rate 40 Tq (1) Data bit rate 20Tq (1)	Nominal bit rate 20 Tq (1) Data bit rate 10 Tq (1)

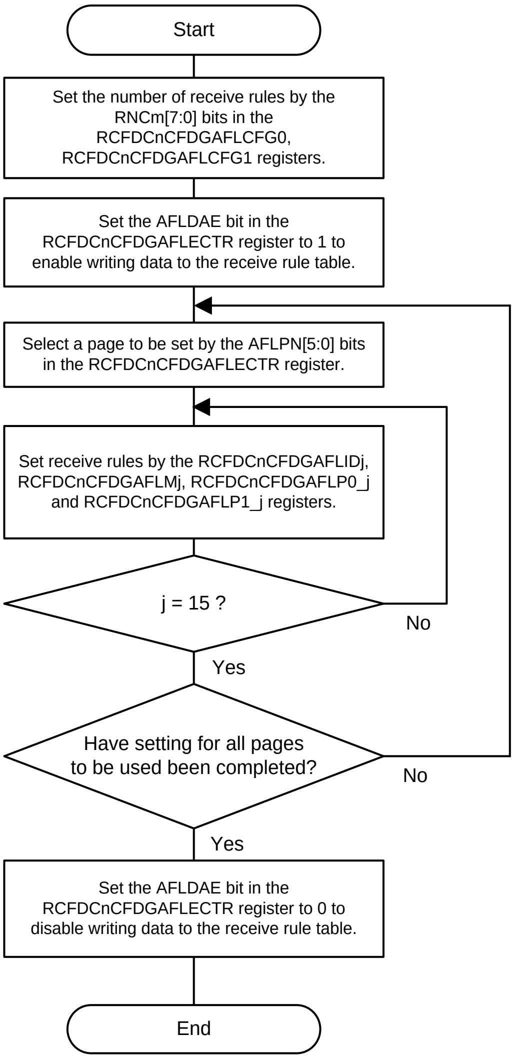
Note: Values in ( ) are baud rate prescaler division values.

24.10.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 63 (for 8-channel unit) by the AFLPN[5:0] bits in the RCFDCnCFDGAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

Figure 24.19, Receive Rule Setting Procedure shows the receive rule setting procedure.



Note: m = 0 to 7, j = 0 to 15

Figure 24.19 Receive Rule Setting Procedure

24.10.1.5 Buffer Setting

Set the number of buffers to be used (number of messages to be stored) and interrupt sources. And also set the payload storage size. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

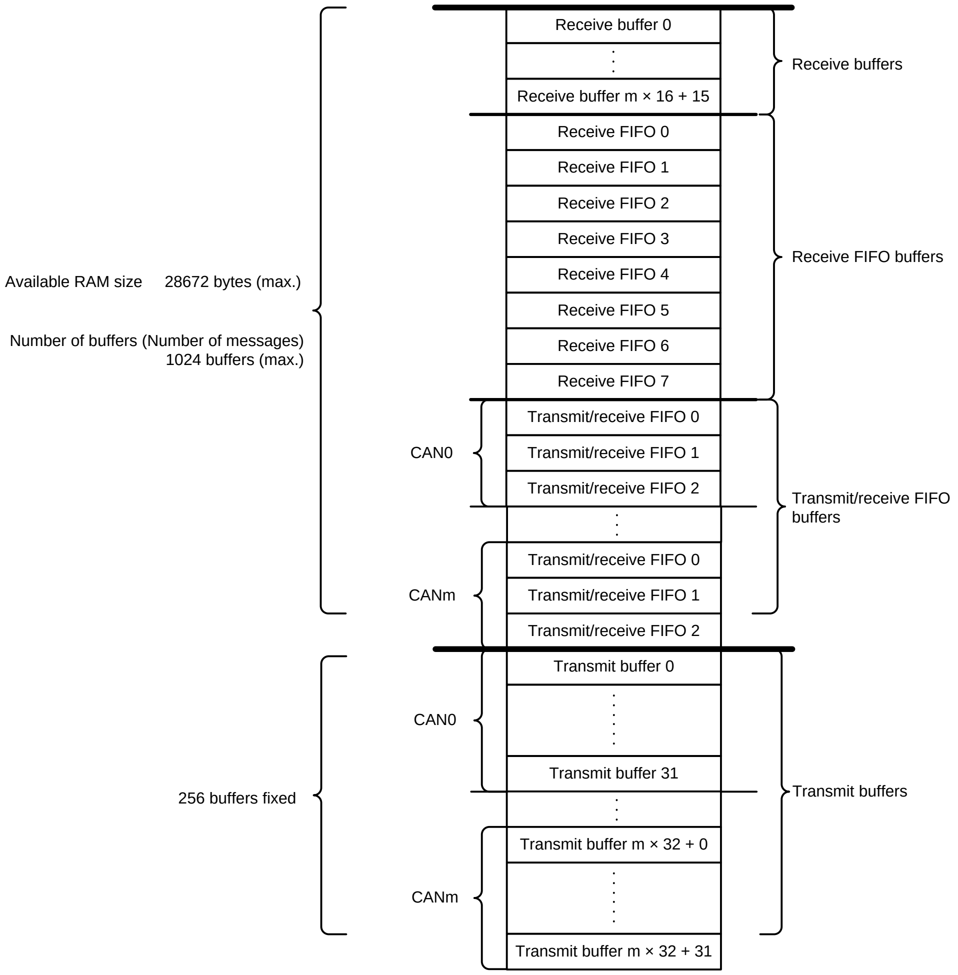
Up to 28672 bytes of the RAM can be used in receive buffers and FIFO buffers. Configure the buffers so that the following conditions are met.

Number of receive buffers × (12 + payload storage size) + total of (number of depth × (12 + payload storage size)) of receive FIFO buffers x + total of (number of depth × (12 + payload storage size)) of transmit/receive FIFO buffers k ≤ 28672 bytes

NOTE

The size of the RAM is for the RS-CANFDs that has 8 channels (m = 0 to 7) of IP design logic.
Regard the size of the RAM as the ones corresponding to your target product (3584 × (m+1) bytes).

Figure 24.20, Buffer Configuration shows the buffer configuration. **Figure 24.21, Buffer Setting Procedure** shows the buffer setting procedure.

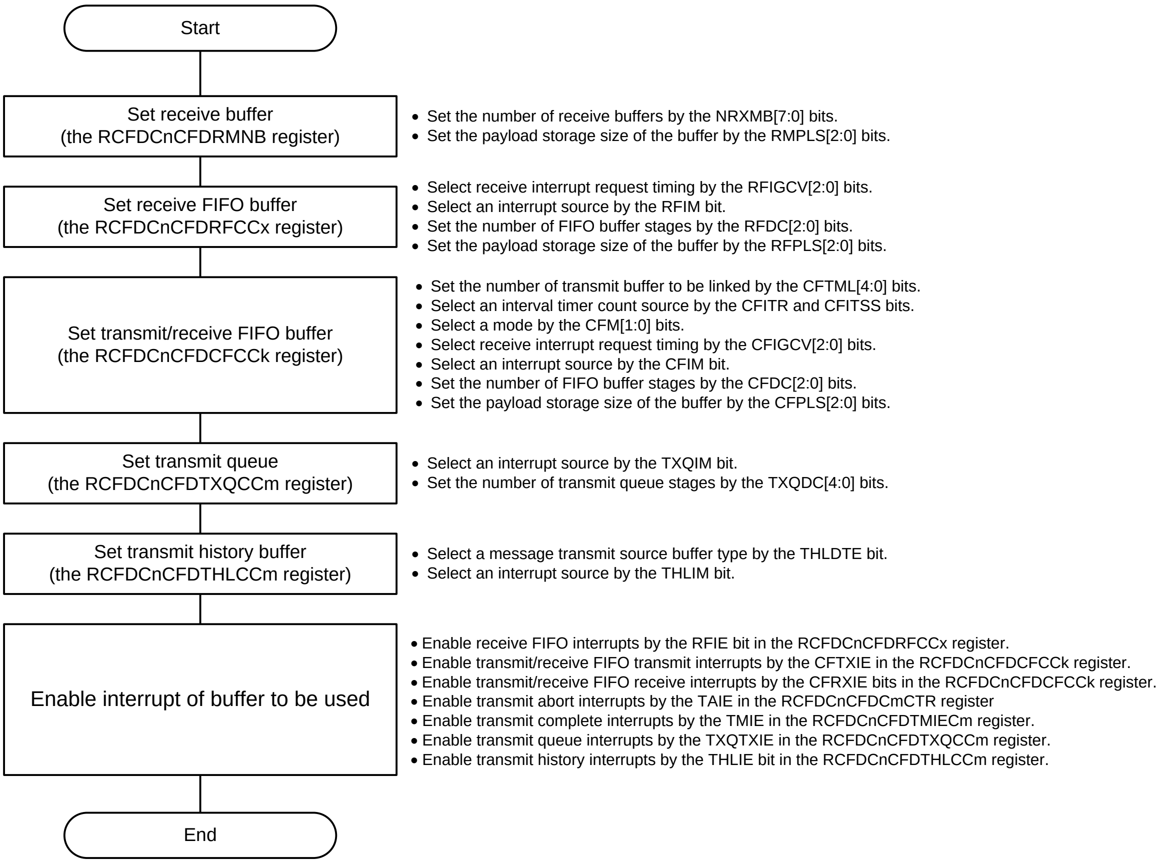


Note: m = 0 to 7

Figure 24.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.



Note: k = 0 to 23, x = 0 to 7

Figure 24.21 Buffer Setting Procedure

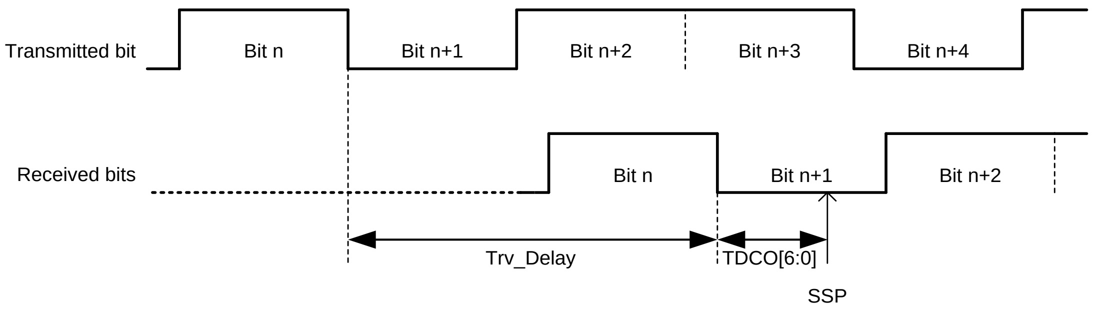
24.10.1.6 Transmitter Delay Compensation

A high baud rate is used in the data phase in CAN FD mode and CAN FD only mode. Transmitter delay compensation is provided as a function to accept propagation delay in data phase.

To use this function, set the TDCE bit in the RCFDCnCFDCmFDCFG register to 1. Also set the secondary sample point (SSP) timing used in the data phase by the TDCOC bit and TDCO[6:0] bits in the RCFDCnCFDCmFDCFG register.

When the TDCOC bit is 0, the SSP timing equals the total value of the delay measured by the RS- CANFD module and the TDCO[6:0] value. (This value is rounded off to the nearest integer of Tq.)

Usually, the TDCO[6:0] value must be equal to SS + TSEG1, the sample point timing



Note: Bit n: Bit in the data phase
TDCO[6:0]: Bits in the RCFDCnCFDCmFDCFG register

Figure 24.22 SSP timing

When the TDCOC bit is 1, the SSP timing is determined only by the TDCO[6:0] value. (When the DBRP[7:0] value in the RCFDCnCFDCmDCFG register is larger than 0, the TDCO[6:0] value is also rounded off to the nearest integer of Tq.)

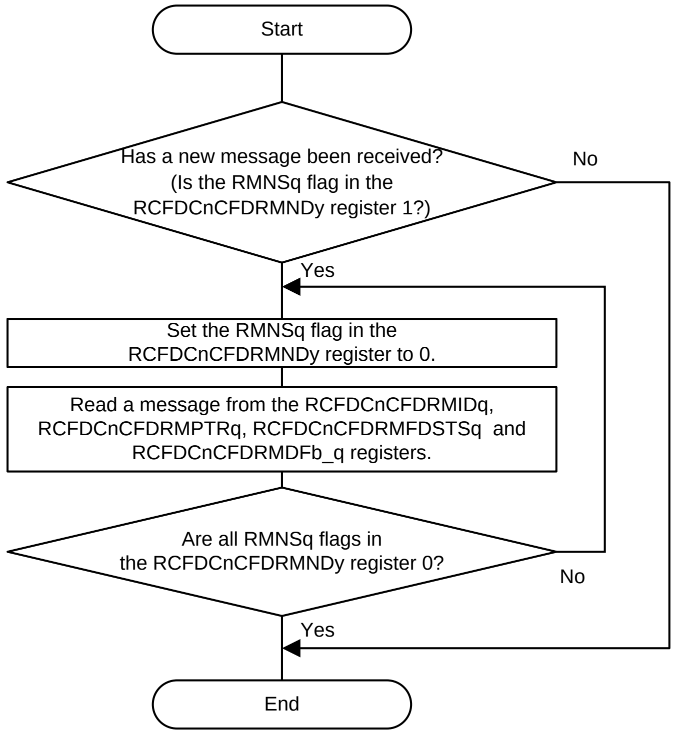
The RS-CANFD module compensates a delay up to (3 CANm bit time - 2 fCAN). (CANm bit time is data bit rate value.)

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RCFDCnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

24.10.2 Reception Procedure

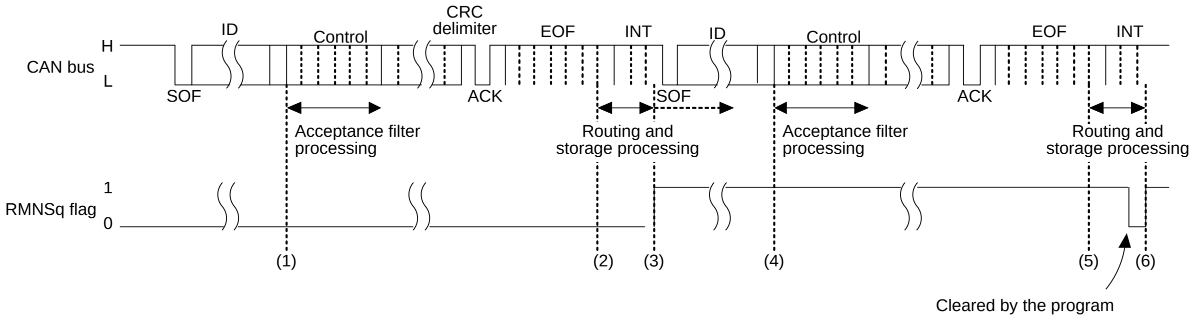
24.10.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RCFDCnCFDRMNDy register (y = 0 to 3, q = 0 to 127) is set to 1 (receive buffer q contains a new message). Messages can be read from registers RCFDCnCFDRMIDq, RCFDCnCFDRMPTRq, RCFDCnCFDRMFDSTSq, and RCFDCnCFDRMDFb_q (b = 0 to 15). If the next message is received before reading the message out of buffer, the message will be overwritten. **Figure 24.23, Receive Buffer Reading Procedure** shows the receive buffer reading procedure. This procedure ensures the consistency of messages read from registers RCFDCnCFDRMIDq, RCFDCnCFDRMPTRq, RCFDCnCFDRMFDSTSq, and RCFDCnCFDRMDFb_q.



Note: y = 0 to 3, q = 0 to 127, b = 0 to 15

Figure 24.23 Receive Buffer Reading Procedure



Note: y = 0 to 3, q = 0 to 127
RMNSq: Flag in the RCFDCnCFDRMNDy register

Figure 24.24 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RCFDCnCFDGCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding RCFDCnCFDRMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RCFDCnCFDGCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag can not be cleared to 0 during storage of messages.

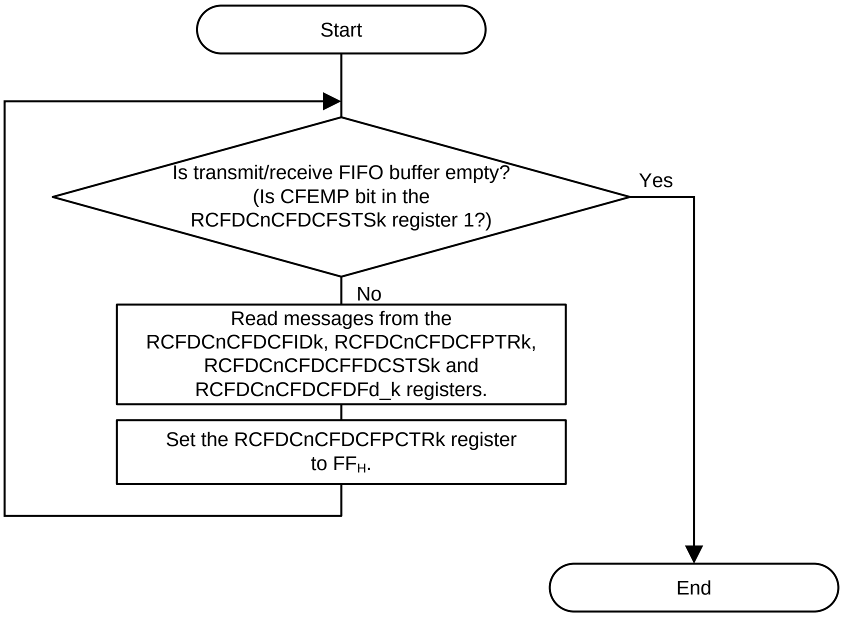
24.10.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message counter (RFMC[7:0] bits in the RCFDCnCFDRFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RCFDCnCFDCFSTSk register (k = 0 to 23)) is incremented by 1. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RCFDCnCFDRFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RCFDCnCFDCFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RCFDCnCFDRFIDx, RCFDCnCFDRFPTRx, RCFDCnCFDRFFDSTSx, and RCFDCnCFDRFDFd_x (d = 0 to 15) registers for receive FIFO buffers, or from the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk and RCFDCnCFDCFDFd_k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RCFDCnCFDRFCCx register or the CFDC[2:0] bits in the RCFDCnCFDCFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RCFDCnCFDRFSTSx register or the CFEMP flag in the RCFDCnCFDCFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RCFDCnCFDRFSTSx register or CFRXIF flag in the RCFDCnCFDCFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.



Note: k = 0 to 23, d = 0 to 15

Figure 24.25 Transmit/Receive FIFO Buffer Reading Procedure

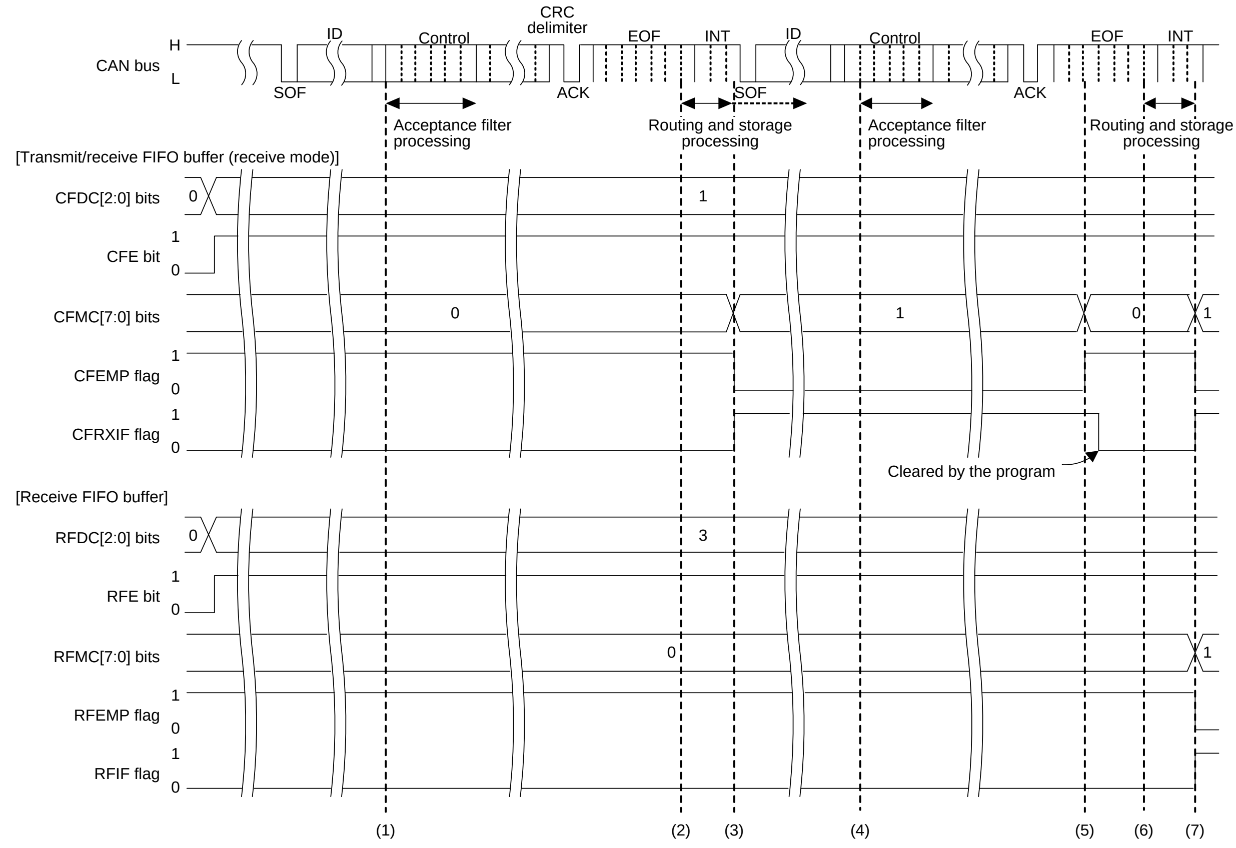
When reading a message, do not read the RCFDCnCFDRFDFd_x or RCFDCnCFDCFDFd_k register corresponding to the area exceeding the payload storage size specified by the RFPLS[2:0] bits in the RCFDCnCFDRFCCx register or the CFPLS[2:0] bits in the RCFDCnCFDCFCCk register.

Table 24.124 Payload Storage Area of Receive FIFO Buffer

Set RFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF1_x
001_B	12 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF2_x
010_B	16 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF3_x
011_B	20 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF4_x
100_B	24 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF5_x
101_B	32 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF7_x
110_B	48 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF11_x
111_B	64 bytes	RCFDCnCFDRFDF0_x to RCFDCnCFDRFDF15_x

Table 24.125 Payload Storage Area of Transmit/Receive FIFO Buffer

Set CFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF1_k
001_B	12 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF2_k
010_B	16 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF3_k
011_B	20 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF4_k
100_B	24 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF5_k
101_B	32 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF7_k
110_B	48 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF11_k
111_B	64 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF15_k



Note: CFDC[2:0], CFE: Bits in the RCFDCnCFDCFCCk register
CFMC[7:0], CFEMP, CFRXIF: Flags in the RCFDCnCFDCFSTSk register
RFDC[2:0], RFE: Bits in the RCFDCnCFDRFCCx register
RFMC[7:0], RFEMP, RFIF: Flags in the RCFDCnCFDRFSTSx register
k = 0 to 23, x = 0 to 7

Figure 24.26 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RCFDCnCFDGCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE bit in the RCFDCnCFDCFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RCFDCnCFDCFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RCFDCnCFDCFSTSk register is incremented and becomes 01_H. When the CFIM bit in the RCFDCnCFDCFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RCFDCnCFDCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, and RCFDCnCFDCFDFd_k registers and write FF_H to the RCFDCnCFDCFPCTRk register. This causes the CFMC[7:0] bits in the

RCFDCnCFDCFSTSk register to be decremented. When CFMC[7:0] becomes 00_H, the CFEMP flag in the RCFDCnCFDCFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RCFDCnCFDGCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RCFDCnCFDRFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RCFDCnCFDRFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RCFDCnCFDRFSTSx register are set to 01_H by being incremented by 1. When the RFIM bit in the RCFDCnCFDRFCCx register is set to 1 (an interrupt request occurs each time a message has been received), the RFIF flag in the RCFDCnCFDRFSTSx register is set to 1 (a receive FIFO interrupt request is present).

24.10.2.3 FIFO Buffer Reading Procedure by DMA Transfer

The following FIFO buffers can be read by DMA transfer.

- RH850/F1KH-D8, RH850/F1KM-S4:
  All receive FIFO buffers x (x = 0 to 7)
  RH850/F1KM-S1:
  The receive FIFO buffers x except for the following (x = 0 to 7)
  The receive FIFO buffer x DMA enable (RCFDCnCFDCDTCT) and the receive FIFO buffer x
  DMA status (RCFDCnCFDCDTSTS) (x = 0 to 5)
- The first transmit/receive FIFO buffer k allocated to channel m (k = 3 × m, m = 0 to 7)

The DMA enable bit (RFDMAEx or CFDMAEm bit in the RCFDCnCFDCDTCT register) can be set at any time. However, before setting the DMA enable bit to 1 (to enable DMA transfer requests), set the receive interrupt enable bit (RFIE bit in the RCFDCnCFDRFCCx register or CFRXIE bit in the RCFDCnCFDCFCCk register) of related FIFO buffers to 0 (to disable interrupts). When DMA transfer requests are enabled, do not write a value to the FIFO control register (RCFDCnCFDRFCCx register or RCFDCnCFDCFCCk register).

When an unread message is remaining in a DMA transfer-enabled FIFO buffer, a DMA transfer request trigger is generated. Specify the address of the FIFO access register address*[1] for the transfer source address, and adjust the transfer size so that data can be read to the end of the payload storage area with a single trigger. The end of the payload storage area depends on the payload storage size specified by the RFPLS[2:0] bits in the RCFDCnCFDRFCCx register or the CFPLS[2:0] bits in the RCFDCnCFDCFCCk register.

After the end of the payload stored in the FIFO buffer has been read, the RFMC[7:0] value in the RCFDCnCFDRFSTSx register or the CFMC[7:0] value in the RCFDCnCFDCFSTSk register is automatically decremented by 1. After that, if an unread message is remaining in the FIFO buffer, a trigger is generated again.

When the RFDMAEx or CFDMAEm bit is set to 0 (to disable DMA transfer requests) during DMA transfer, wait until the DMA transfer status (RFDMASTSx or CFDMASTSm bit in the RCFDCnCFDCDTSTS register) is cleared to 0 (DMA transfer disabled), and then start the next processing (enabling DMA transfer again etc.). When disabling DMA transfer, examine how to process a message remaining in the FIFO buffer and a newly arriving message. When the FIFO buffer is enabled, it continues to receive messages.

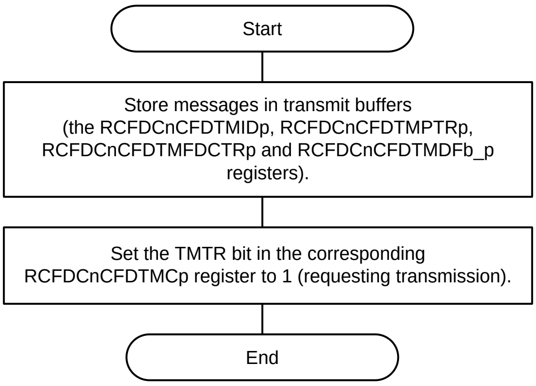
Note 1.
- Receive FIFO buffer
  RCFDCnCFDRFIDx, RCFDCnCFDRFPTRx, RCFDCnCFDRFFDSTSx, RCFDCnCFDRFDFd_x
- Transmit/Receive FIFO buffer
  RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, RCFDCnCFDCFDFd_k

24.10.3 Transmission Procedure

24.10.3.1 Procedure for Transmission from Transmit Buffers

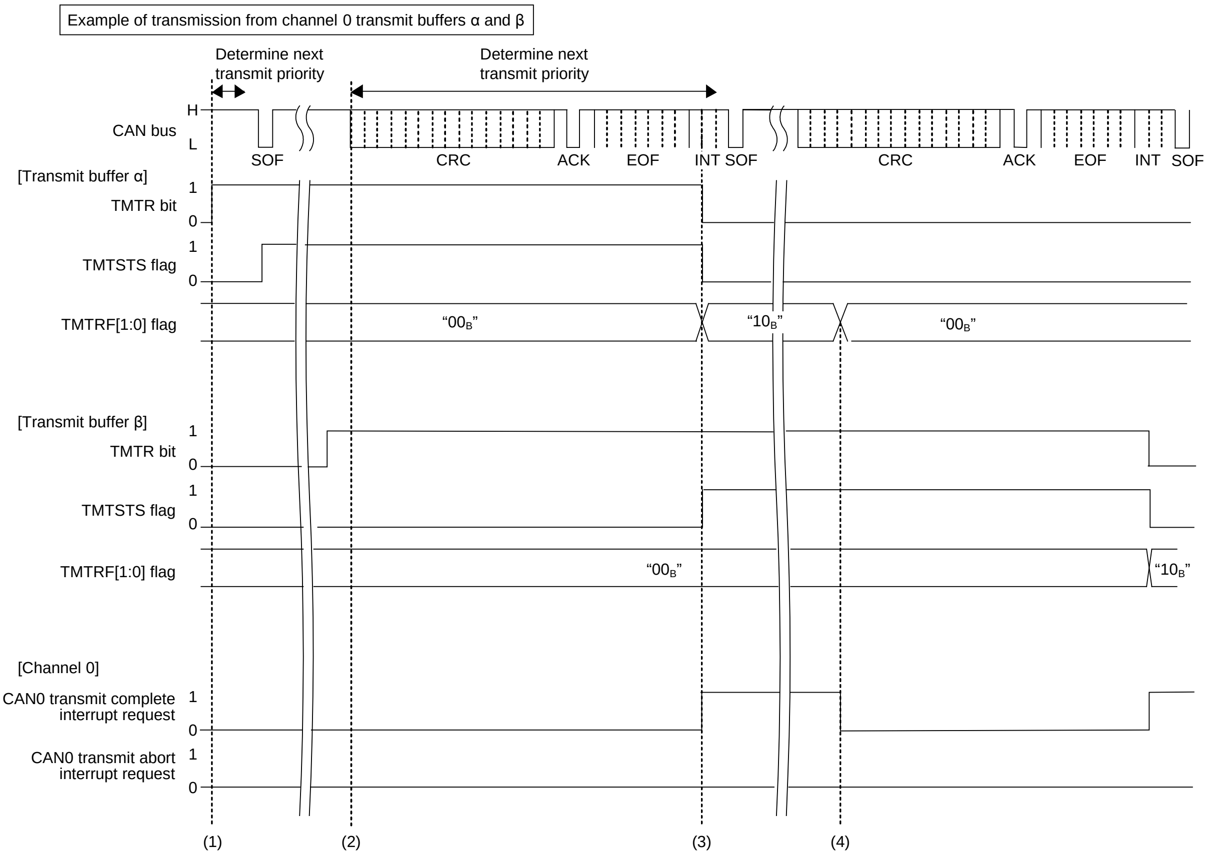
Figure 24.27, Procedure for Transmission from Transmit Buffers shows the procedure for transmission from transmit buffers.

Figure 24.28, Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully) shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 24.29, Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.



Note: p = 0 to 255, b = 0 to 15

Figure 24.27 Procedure for Transmission from Transmit Buffers



Note: α = 0 to 31, β = 0 to 31

TMTR: Bit in the RCFDCnCFDTMCp register

TMTSTS, TMTRF[1:0]: Flags in the RCFDCnCFDTMSTSp register

Figure 24.28 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) When the TMTR bit in the RCFDCnCFDTMCα register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer α is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RCFDCnCFDTMSTSα register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

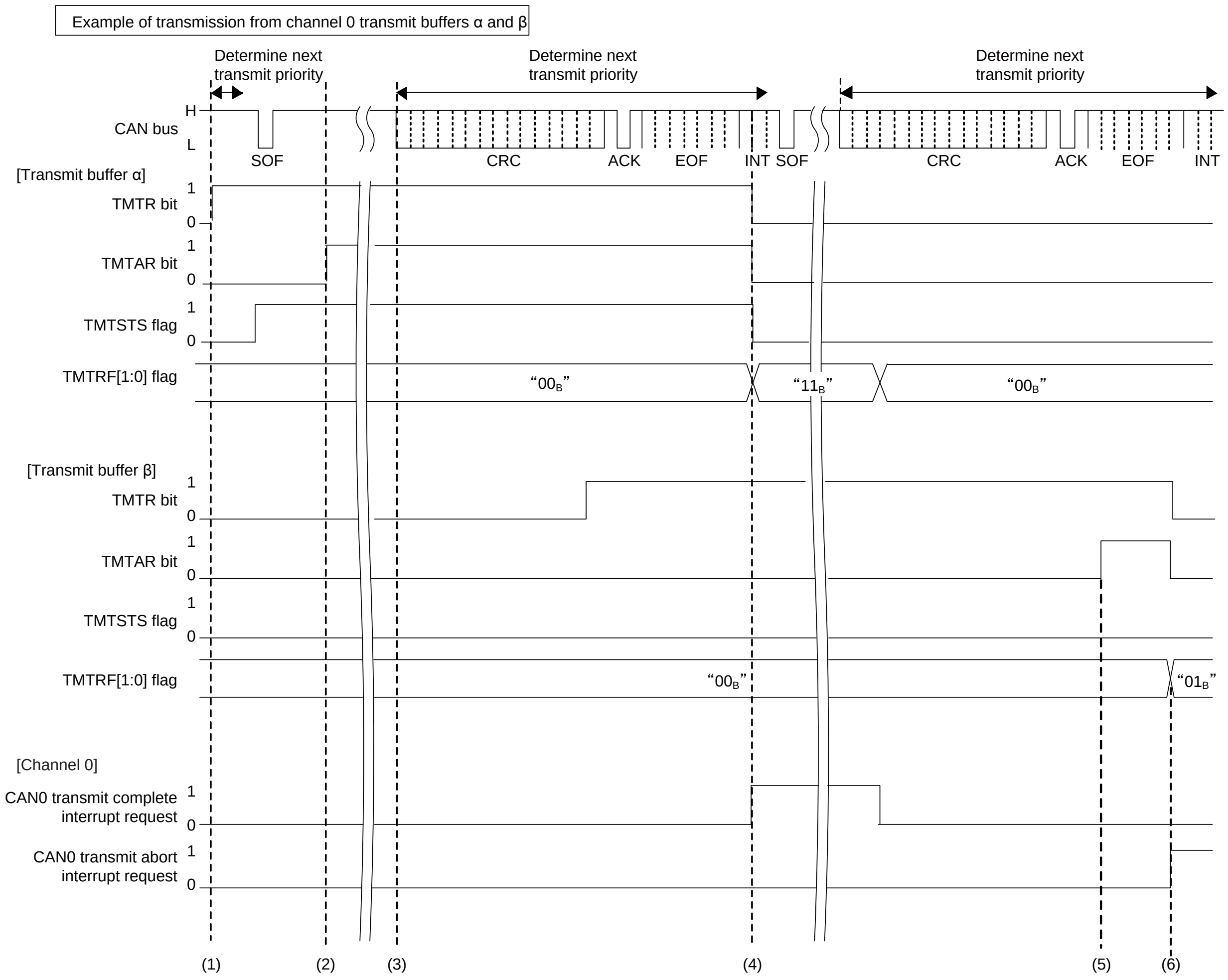
(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur between transmissions because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the TMTRF[1:0] flag in the RCFDCnCFDTMSTSα register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RCFDCnCFDTMCα register are cleared to 0. When the TMIEα bit in the RCFDCnCFDTMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

If an arbitration-lost occurs after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest- priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted.



Note: α = 0 to 31, β = 0 to 31

TMTR, TMTAR: Bits in the RCFDCnCFDTMCp register

TMTSTS, TMTRF[1:0]: Flags in the RCFDCnCFDTMSTSp register

Figure 24.29 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMTR bit in the RCFDCnCFDTMCα register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer α is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RCFDCnCFDTMSTSα register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts at the first bit of the CRC field for the next transmission. In this timing chart, buffer β is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmission completes successfully, the TMTRF[1:0] flag in the RCFDCnCFDTMSTSα register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RCFDCnCFDTMCα register are cleared to 0. When the TMIEα value in the RCFDCnCFDTMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RCFDCnCFDCmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

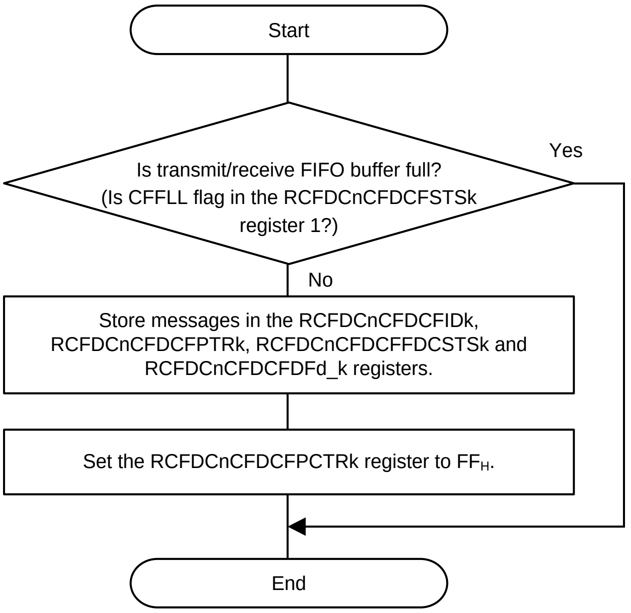
If an arbitration loss occurs after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted.

24.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 24.30, Procedure for Transmission from Transmit/Receive FIFO Buffers shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 24.31, Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully) shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 24.32, Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.



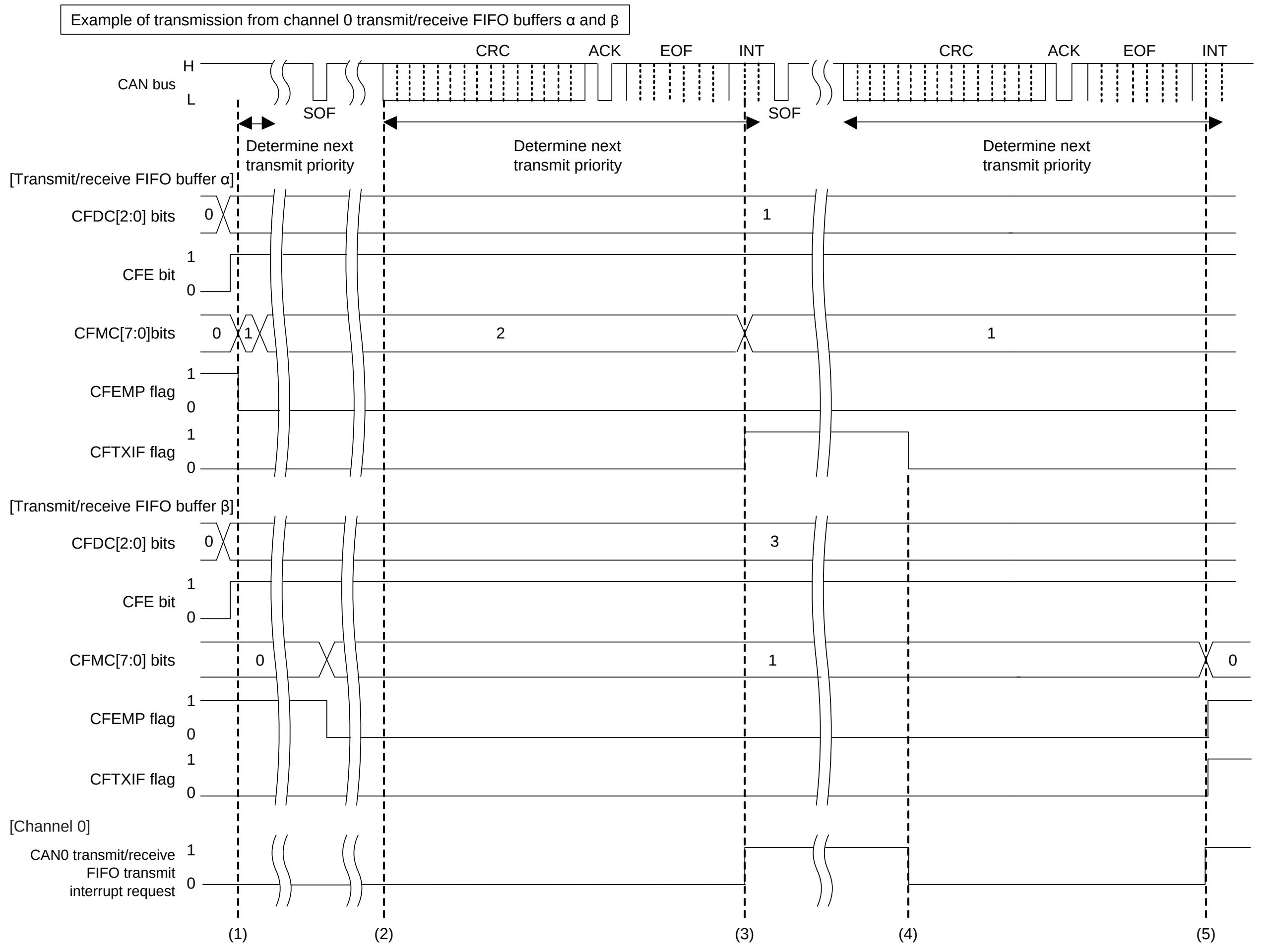
Note: k = 0 to 23, d = 0 to 15

Figure 24.30 Procedure for Transmission from Transmit/Receive FIFO Buffers

When storing a message, do not write a value to the RCFDCnCFDCFDFd_k register corresponding to the area exceeding the payload storage size specified by the CFPLS[2:0] bits in the RCFDCnCFDCFCCk register.

Table 24.126 Payload Storage Area of Transmit/Receive FIFO Buffer

Set CFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF1_k
001_B	12 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF2_k
010_B	16 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF3_k
011_B	20 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF4_k
100_B	24 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF5_k
101_B	32 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF7_k
110_B	48 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF11_k
111_B	64 bytes	RCFDCnCFDCFDF0_k to RCFDCnCFDCFDF15_k



Note: α = 0 to 2, β = 0 to 2

CFDC[2:0],CFE: Bits in the RCFDCnCFDCFCCk register

CFMC[7:0], CFEMP, CFTXIF: Flags in the RCFDCnCFDCFSTSk register

Figure 24.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFE bit in the RCFDCnCFDCFCCα register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RCFDCnCFDCFCCα register is 001_B (4 messages) or more and the CFMC[7:0] value in the RCFDCnCFDCFSTSα register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer α of channel 0.

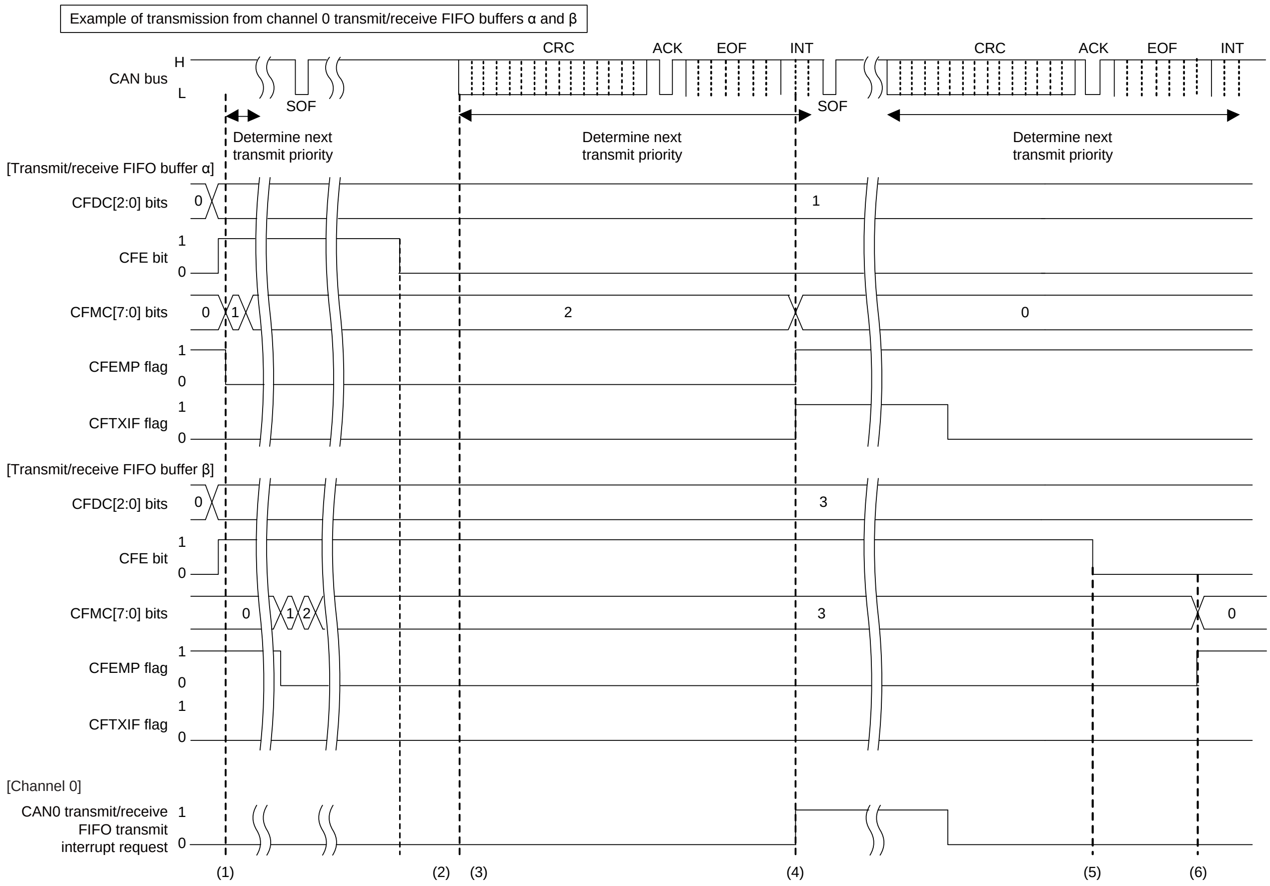
(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the CFMC[7:0] value in the RCFDCnCFDCFSTSα register is decremented by 1. Setting the CFIM bit in the RCFDCnCFDCFCCα register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RCFDCnCFDCFSTSα register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The program can clear the CFTXIF flag.

(5) Message transmission from transmit/receive FIFO buffer β of channel 0 completes and the CFMC[7:0] value in the RCFDCnCFDCFSTSβ register is decremented by 1. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RCFDCnCFDCFSTSβ register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RCFDCnCFDCFSTSα and RCFDCnCFDCFSTSβ registers is set to 1 (the transmit/receive FIFO buffer is full).



Note: α = 0 to 2, β = 0 to 2
CFDC[2:0], CFE: Bits in the RCFDCnCFDCFCCk register
CFMC[7:0], CFEMP, CFTXIF: Flags in the RCFDCnCFDCFSTSk register

Figure 24.32 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFE bit in the RCFDCnCFDCFCCα register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RCFDCnCFDCFCCα register is 001_B (4 messages) or more and the CFMC[7:0] value in the RCFDCnCFDCFSTSα register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer α of channel 0.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

(3) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. In this figure, transmit/receive FIFO buffer β is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

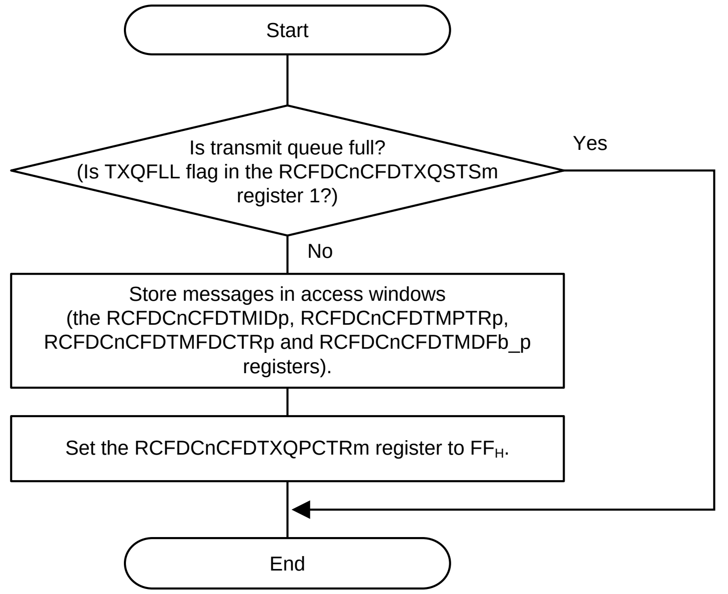
(4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RCFDCnCFDCFSTSα register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer β), transmit/receive FIFO buffers β cannot be disabled immediately even if the CFE bit in the RCFDCnCFDCFCCβ register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RCFDCnCFDCFSTSβ register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers β are disabled and the CFMC[7:0] bits in the RCFDCnCFDCFSTSβ register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer β is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer β is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

24.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 24.33, Procedure for Transmission from the Transmit Queue shows the procedure for transmission from the transmit queue.



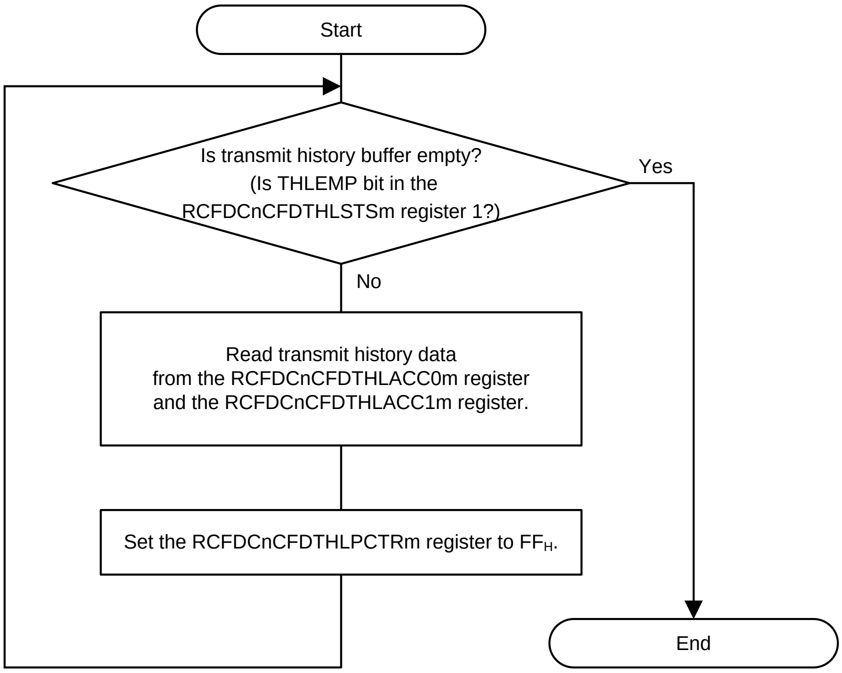
Note: p = m × 32 + 31, m = 0 to 7

Figure 24.33 Procedure for Transmission from the Transmit Queue

24.10.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RCFDCnCFDTHLACC0m register and the RCFDCnCFDTHLACC1m register.

The next data can be accessed by writing FF_H to the corresponding RCFDCnCFDTHLPCTRm register (m = 0 to 7) after reading a set of data. **Figure 24.34, Transmit History Buffer Reading Procedure** shows the transmit history buffer reading procedure.



Note: m = 0 to 7

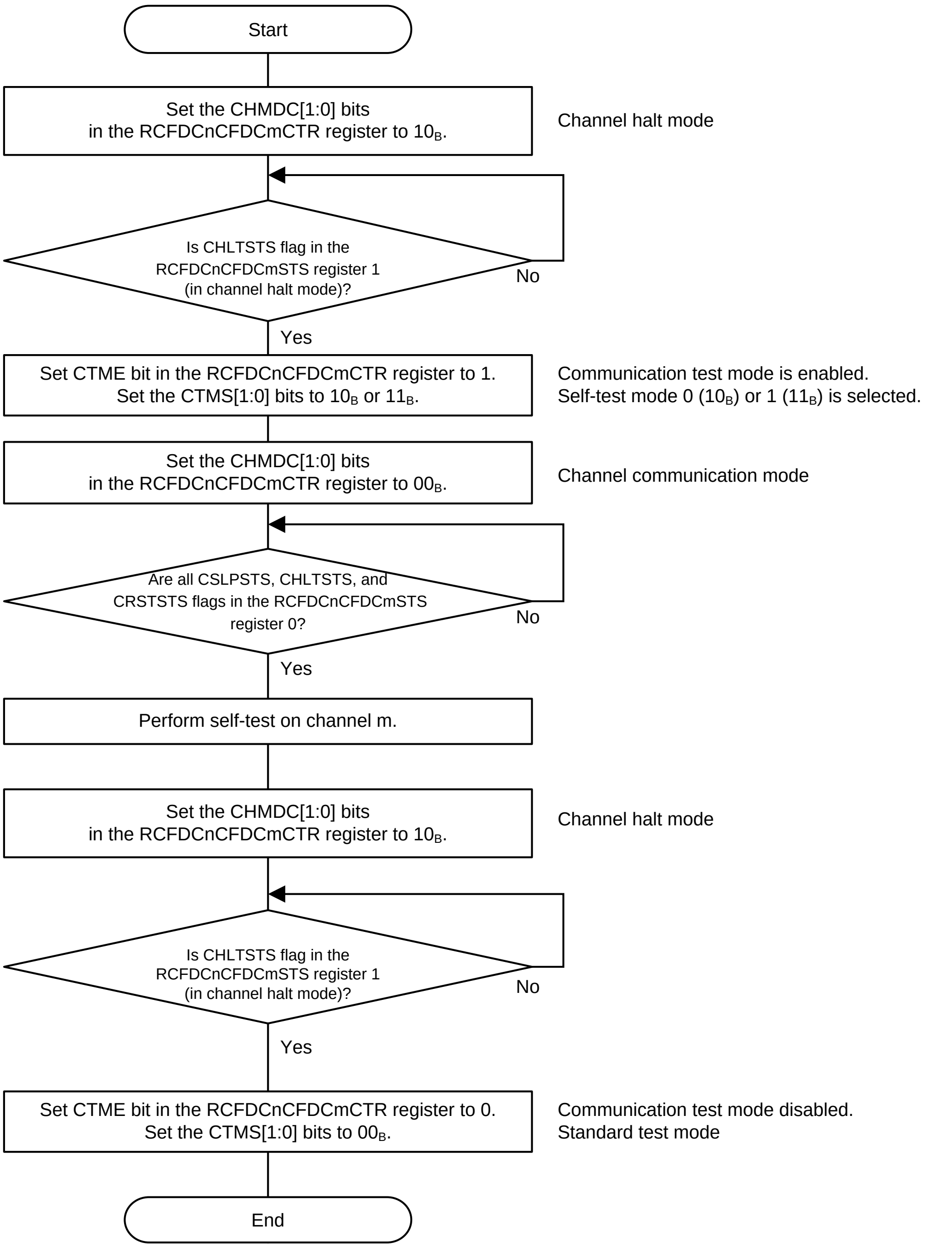
Figure 24.34 Transmit History Buffer Reading Procedure

24.10.4 Test Settings

24.10.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 24.35, Self-Test Mode Setting Procedure shows the self-test mode setting procedure.



Note: m = 0 to 7

Figure 24.35 Self-Test Mode Setting Procedure

24.10.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 24.127, Protection Release Data for Test Function** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RCFDCnCFDGLOCKK register, then set the target test bit to 1.

Table 24.127 Protection Release Data for Test Function

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575_H	$8A8A_H$	RTME bit in the RCFDCnCFDGTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data1.
Figure 24.36, Protection Release Procedure shows the procedure for releasing the protection.

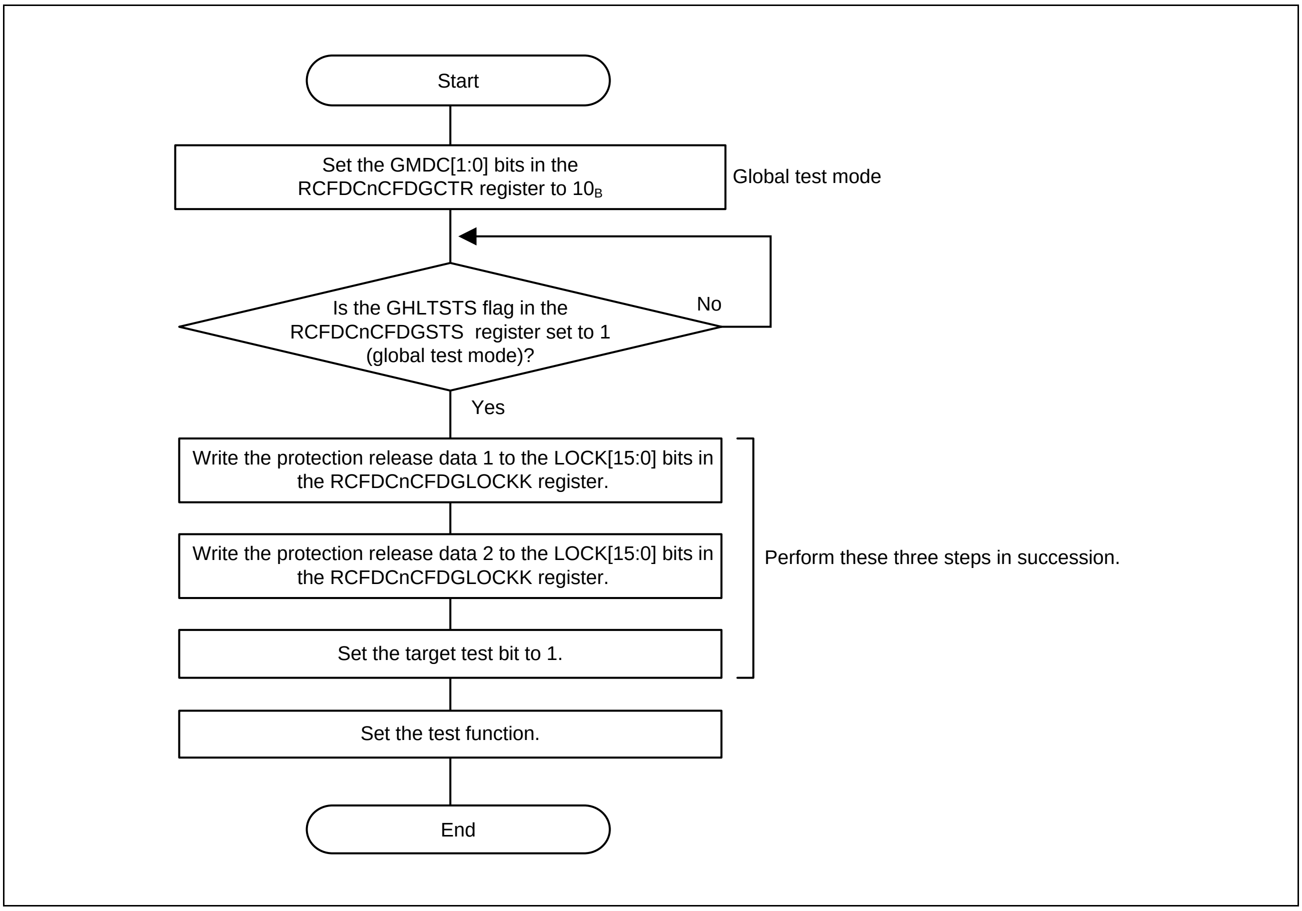
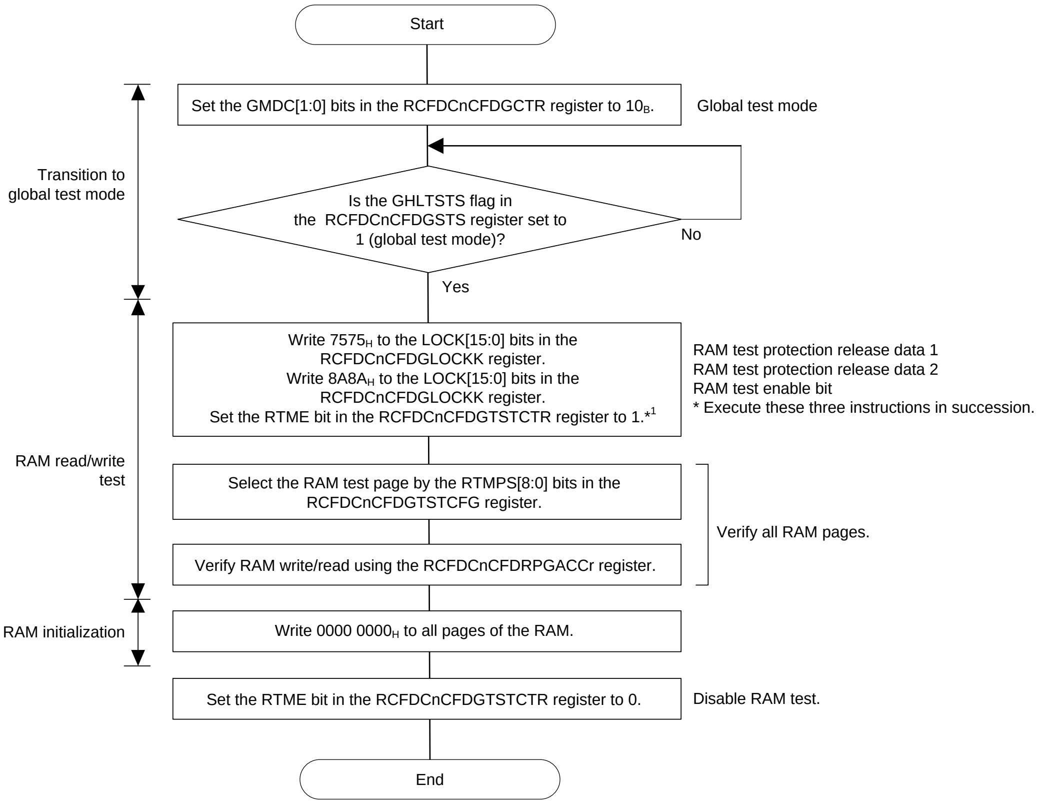


Figure 24.36 Protection Release Procedure

24.10.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before ending the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 24.37, RAM Test Setting Procedure shows the RAM test setting procedure.



Note: r = 0 to 63

Note 1. Before changing to the RAMTEST mode, make sure to:
- Cancel transmission request(s), if any,
- Disable all the FIFO and Transmit Queues, and,
- Clear the receiving flags of receiving buffers.

Figure 24.37 RAM Test Setting Procedure

24.10.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 24.38, Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1) shows the inter-channel communication test setting procedure.

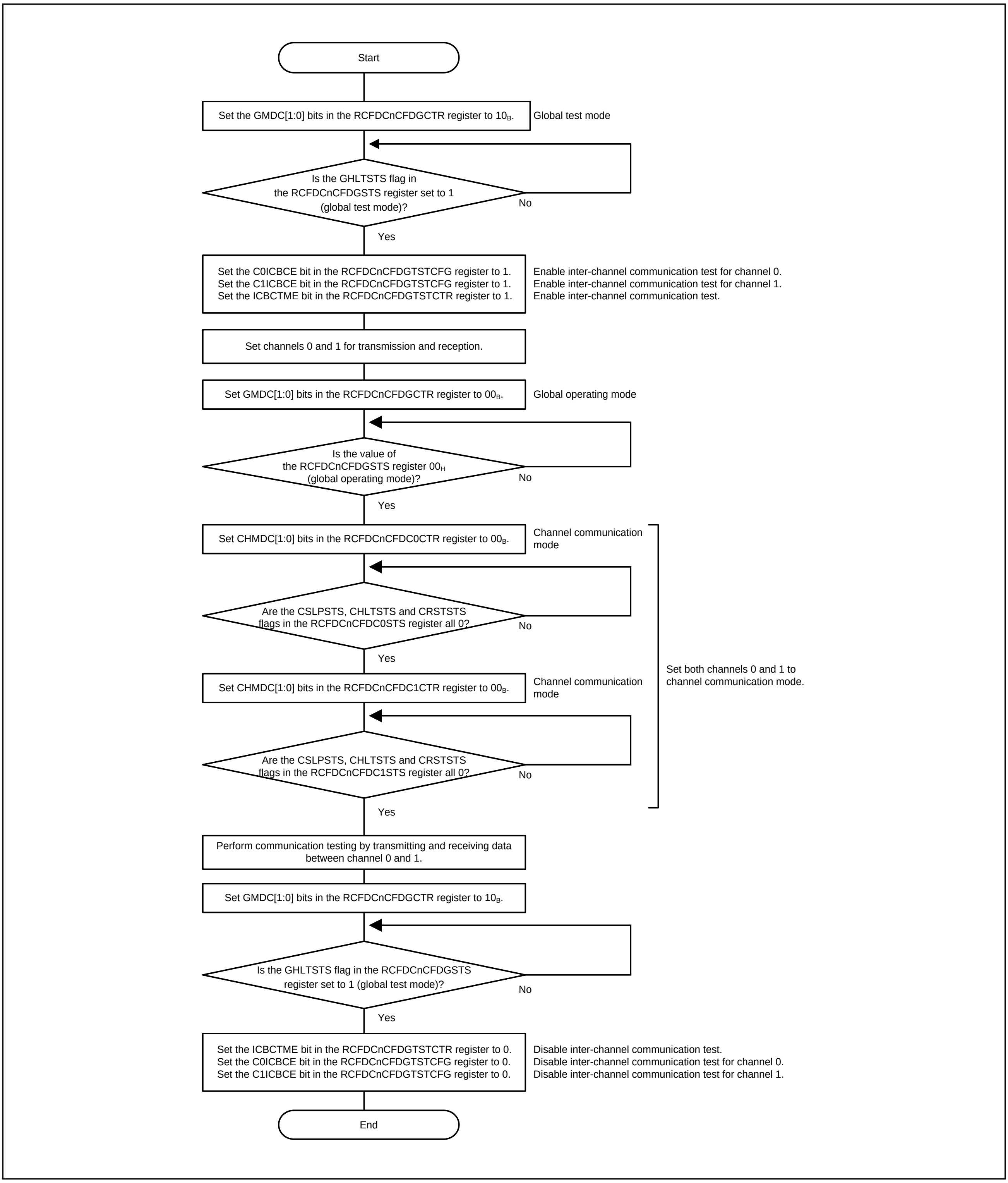


Figure 24.38 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

24.11 Notes on the RS-CANFD Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RCFDCnCFDGSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RCFDCnCFDCmSTS register (m = 0 to 7) for transitions.
- When only classical CAN frames are used in CAN FD mode, set the RCFDCnCFDCmDCFG register to the value equal to the set RCFDCnCFDCmNCFG register value.
- The acceptance filter processing checks receive rules sequentially in ascending order from the smallest rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the smallest number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RCFDCnCFDTMCp) of the corresponding transmit buffer to 00_H. The status register (RCFDCnCFDTMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RCFDCnCFDTMTRSTS0 to RCFDCnCFDTMTRSTS7, RCFDCnCFDTMTARSTS0 to RCFDCnCFDTMTARSTS7, RCFDCnCFDTMTCSTS0 to RCFDCnCFDTMTCSTS7, and RCFDCnCFDTMTASTS0 to RCFDCnCFDTMTASTS7), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RCFDCnCFDTMIEC0 to RCFDCnCFDTMIEC7) to 0 (transmit buffer interrupt is disabled).
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a newly received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- In the case of registers that access the RAM, the value after reset shown in **Section 24.3, Registers** indicate the values cleared by initialization of the CAN RAM. Values before clear are undefined. The following registers apply.
  - Receive rule (RCFDCnCFDGAFLIDj, RCFDCnCFDGAFLMj, RCFDCnCFDGAFLP0_j, RCFDCnCFDGAFLP1_j registers)
  - Receive buffers (RCFDCnCFDRMIDq, RCFDCnCFDRMPTRq, RCFDCnCFDRMFDSTSq, RCFDCnCFDRMDFb_q registers)
  - Receive FIFO buffer access registers (RCFDCnCFDRFIDx, RCFDCnCFDRFPTRx, RCFDCnCFDRFFDSTSx, and RCFDCnCFDRFDFd_x registers)
  - Transmit/receive FIFO buffer access registers (RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k registers)
  - Transmit buffers (RCFDCnCFDTMIDp, RCFDCnCFDTMPTRp, RCFDCnCFDTMFDCTRp, and RCFDCnCFDTMDFb_p registers)
  - Transmit history access registers (RCFDCnCFDTHLACC0m, RCFDCnCFDTHLACC1m registers)
  - RAM test page access register (RCFDCnCFDRPGACCr register)

- The values of unused receive buffers (RCFDCnCFDRMIDq, RCFDCnCFDRMPTRq, RCFDCnCFDRMFDSTSq, and RCFDCnCFDRMDFb_q registers), receive FIFO buffer access registers (RCFDCnCFDRFIDx, RCFDCnCFDRFPTRx, RCFDCnCFDRFFDSTSx, and RCFDCnCFDRFDFd_x registers) and transmit/receive FIFO buffer access registers (RCFDCnCFDCFIDk, RCFDCnCFDCFPTRk, RCFDCnCFDCFFDCSTSk, and RCFDCnCFDCFDFd_k registers) are undefined when the RS-CANFD module transitions to global operation mode or global test mode after exiting from global reset mode.

24.12 Detection and Correction of Errors in RS-CANFD RAM

For details of ECC, see **Section 40A.2.6, ECC for Peripheral RAM**, **Section 40B.2.6, ECC for Peripheral RAM**, **Section 40C.2.5, ECC for Peripheral RAM**.

Section 25 FlexRay (FLXA)

This section contains a generic description of the FlexRay (FLXA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of FLXA.

CAUTION

Please read “global RAM” described in this section as “global RAM/retention RAM”.

NOTE

The POC described in this section is an abbreviated form of Protocol Operation Control and differs from the POC (power-on clear) described in **Section 10A, Power Supply Circuit of RH850/F1KH-D8, Section 10B, Power Supply Circuit of RH850/F1KM-S4** and **Section 10C, Power Supply Circuit of RH850/F1KM-S1**.

25.1 Features of RH850/F1KH, RH850/F1KM FLXA

25.1.1 Number of Units and Channels

This microcontroller has the following number of FLXA units and channels.

Table 25.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	FLXAn (n = 0)	FLXAn (n = 0)	FLXAn (n = 0)

Table 25.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	FLXAn (n = 0)	FLXAn (n = 0)	FLXAn (n = 0)	FLXAn (n = 0)	FLXAn (n = 0)

Table 25.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	—	—
Name	—	—	—	—

Table 25.4 FLXA Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name	Number of Channels per Unit	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
FLXA0	2 (A ch, B ch)	✓	✓	✓

Table 25.5 FLXA Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name	Number of Channels per Unit	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
FLXA0	2 (A ch, B ch)	✓	✓	✓	✓	✓

Table 25.6 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual FLXA units are identified by the index “n”; for example, FLXAnFROC is the FlexRay control register.

Table 25.7 Index (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual FLXA units are identified by the index “n”; for example, FLXAnFROC is the FlexRay control register.

25.1.2 Register Base Address

FLXAn base address is listed in the following table.

FLXAn register addresses are given as an offset from the base address in general.

Table 25.8 Register Base Address (RH850/F1KH-D8)

Base Address Name	Base Address
<FLXA0_base>	1002 0000_H

Table 25.9 Register Base Address (RH850/F1KM-S4)

Base Address Name	Base Address
<FLXA0_base>	1002 0000_H

25.1.3 Clock Supply

The FLXAn clock supply is shown in the following table.

Table 25.10 Clock Supply (RH850/F1KH-D8)

Unit Name	Clock Name for the Unit	Supply Clock Name
FLXAn	hclk	CPUCLK_L
	clkc	CKSCLK_PPLLCLK*[1]
	Register access clock	CPUCLK_L

Note 1. FLXA0 clock can stop by OPBT0. FLXA0EN, refer to **Section 44.9.2, OPBT0 — Option Byte 0**.

Table 25.11 Clock Supply (RH850/F1KM-S4)

Unit Name	Clock Name for the Unit	Supply Clock Name
FLXAn	hclk	CPUCLK_L
	clkc	CKSCLK_PPLLCLK*[1]
	Register access clock	CPUCLK_L

Note 1. FLXA0 clock can stop by OPBT0. FLXA0EN, refer to **Section 44.9.2, OPBT0 — Option Byte 0**.

Table 25.12 Range of Operating Frequency Depending on the Transfer Rate

	Range of Operating Frequency	
Data Transfer Rate	hclk	clkc
10 Mbps	45 ≤ hclk ≤ 60 MHz	80 MHz
5 Mbps	27 ≤ hclk ≤ 60 MHz	80 MHz
2.5 Mbps	27 ≤ hclk ≤ 60 MHz	80 MHz

25.1.4 Interrupt Requests

The FLXAn interrupt requests are listed in the following table.

Table 25.13 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
FLXA0			
INTFLXA0LINE0	FlexRay0 interrupt	179	—
INTFLXA0LINE1	FlexRay1 interrupt	180	—
INTFLXA0TIM0	Timer 0 interrupt	181	—
INTFLXA0TIM1	Timer 1 interrupt	182	—
INTFLXA0TIM2	Timer 2 interrupt	183	—
INTFLXA0FDA	FIFO transfer interrupt	173	—
INTFLXA0FW	FIFO transfer warning interrupt	174	—
INTFLXA0OW	Output transfer warning interrupt	178	—
INTFLXA0OT	Output transfer end interrupt	177	—
INTFLXA0IQF	Input queue full interrupt	176	—
INTFLXA0IQE	Input queue empty interrupt	175	—

Table 25.14 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
FLXA0			
INTFLXA0LINE0	FlexRay0 interrupt	179	—
INTFLXA0LINE1	FlexRay1 interrupt	180	—
INTFLXA0TIM0	Timer 0 interrupt	181	—
INTFLXA0TIM1	Timer 1 interrupt	182	—
INTFLXA0TIM2	Timer 2 interrupt	183	—
INTFLXA0FDA	FIFO transfer interrupt	173	—
INTFLXA0FW	FIFO transfer warning interrupt	174	—
INTFLXA0OW	Output transfer warning interrupt	178	—
INTFLXA0OT	Output transfer end interrupt	177	—
INTFLXA0IQF	Input queue full interrupt	176	—
INTFLXA0IQE	Input queue empty interrupt	175	—

25.1.5 Reset Sources

FLXAn reset sources are listed in the following table. FLXAn is initialized by the following reset signal:

Table 25.15 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
FLXAn	All reset sources (ISORES)

Table 25.16 Reset Sources (RH850/F1KM-S4)

Unit Name	Reset Source
FLXAn	All reset sources (ISORES)

25.1.6 External Input/Output Signals

External input/output signals of FLXAn are listed below.

Table 25.17 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal Name
FLXA0		
rxda_extfxr	Channel A receive data input	FLXA0RXDA
fxr_txda	Channel A transmit data output	FLXA0TXDA
fxr_txena_n	Channel A transmit data enable	FLXA0TXENA
rxdb_extfxr	Channel B receive data input	FLXA0RXDB
fxr_txdb	Channel B transmit data output	FLXA0TXDB
fxr_txenb_n	Channel B transmit data enable	FLXA0TXENB
stpwt_extfxr	Stop watch trigger input	FLXA0STPWT

Table 25.18 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal Name
FLXA0		
rxda_extfxr	Channel A receive data input	FLXA0RXDA
fxr_txda	Channel A transmit data output	FLXA0TXDA
fxr_txena_n	Channel A transmit data enable	FLXA0TXENA
rxdb_extfxr	Channel B receive data input	FLXA0RXDB
fxr_txdb	Channel B transmit data output	FLXA0TXDB
fxr_txenb_n	Channel B transmit data enable	FLXA0TXENB
stpwt_extfxr	Stop watch trigger input	FLXA0STPWT

25.1.7 Functions

For communication on a FlexRay network, individual message buffers that can hold up to 254 bytes of data can be configured. Up to 128 message buffers can be configured in the message RAM. All functions concerning the handling of messages are implemented in the Message Handler. These functions include acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule, and providing message status information.

The register set of the FlexRay IP-module can be accessed directly by an external Host via the module’s Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, to access the Message RAM via Input/Output Buffer; and to control the data transfer between the Message RAM and the local RAM/global RAM.

The FlexRay IP-module supports the following features:

Item	Specification
Communication	Conformance with FlexRay protocol specification v2.1
Data transfer rate	Up to 10 Mbit/s on each channel
Input/Output pins per channel	TxD, RxD, TxEN
FlexRay channels	2 (channels A and B)
Message buffers	Up to 128 message buffers configurable Configuration of message buffers with different payload lengths possible Each message buffer can be configured as a receive buffer, transmit buffer, part of a receive FIFO buffer. Filtering for slot counter, cycle counter, and channel
Message RAM	8 Kbyte of Message RAM for storage can be configured 128 message buffers with a data section of up to 48 bytes or up to 30 message buffers with a data section of 254 bytes
FIFO	One configurable receive FIFO
Message buffer access	By host CPU via input and output buffers Input Buffer: Holds message to be transferred to the Message RAM Output Buffer: Holds message read from the Message RAM
	By data transfer function Input transfer: Message buffer content is transferred from local RAM/global RAM to Message RAM on CPU request Output transfer: Message buffer content is transferred from Message RAM to local RAM/global RAM automatically
Network management	Supported
Interrupts	Maskable module interrupts
Timer	Two absolute timers One relative timer One stop watch timer

25.1.8 Block Diagram

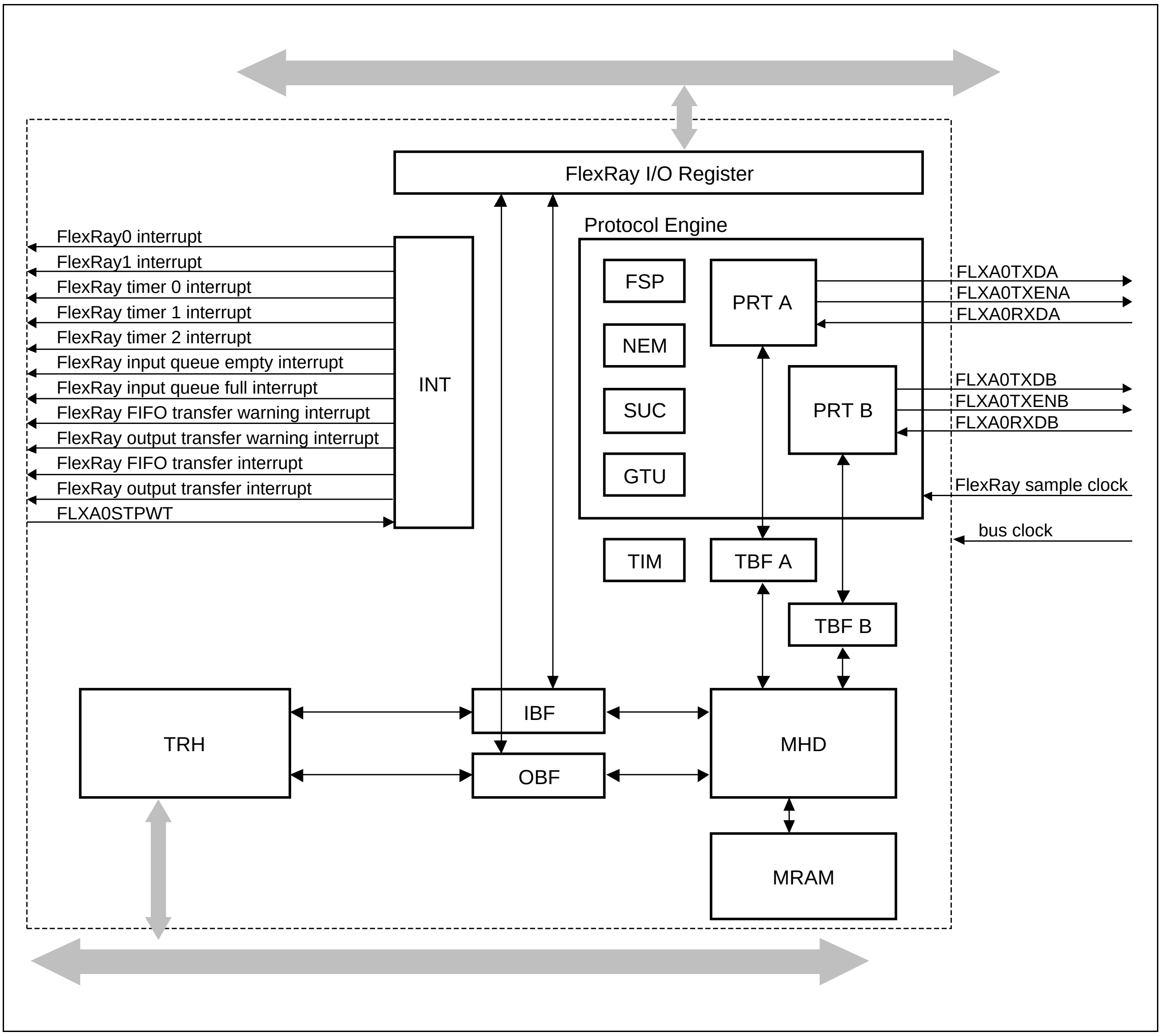


Figure 25.1 FlexRay IP Block Diagram

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The FlexRay Message Handler controls data transfers between the following components:

- Input/Output Buffer and Message RAM

- Temporary buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

Temporary buffer RAM (TBF A/B)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Temporary buffer RAMs for intermediate message storage and to the physical layer via bus driver (BD).

This controller has the following functionality:

- Control and check of bit timing
- Reception/transmission of FlexRay frames and symbols
- Check of header CRC
- Generation/check of frame CRC
- Interfacing with bus driver

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick (µT)
- Generation of macrotick (MT)
- Fault tolerant clock synchronization, by FTM algorithm, for:
  - rate correction
  - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Controller has the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

Frame and Symbol Processing (FSP)

The Frame and Symbol Processing has the following functions:

- Checks whether the timing of frames and symbols is correct or not
- Checks whether the syntax and semantics of the received frame are correct or not
- Sets the slot status flags

Network Management (NEM)

Handles the network management vector.

Interrupt Control (INT)

The Interrupt Controller has the following functions:

- Provides error and status interrupt flags
- Enables and disables interrupt sources
- Assignment of interrupt sources to one of the two general module interrupt lines
- Enables and disables module interrupt lines

Timer (TIM)

The Timer module includes the following macrotick timer:

- Two absolute timers
- One relative timer
- One stop watch timer

Transfer Handler (TRH)

Handles the data transfer between local RAM/global RAM and FlexRay module. The Transfer Handler supports the following transfer types:

- Transfer of buffer configuration data from the local RAM/global RAM to the Message RAM
- Transfer of payload data for temporary buffers from the local RAM/global RAM to the Message RAM
- Transfer of buffer configuration data and payload data for temporary buffer from the local RAM/global RAM to the Message RAM
- Automatic transfer of payload data from receive buffer to the local RAM/global RAM upon frame reception
- Automatic transfer of payload data, buffer configuration data and message buffer status data from receive buffer to the local RAM/global RAM upon frame reception
- Automatic transfer of buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the local RAM/global RAM in response to slot status update
- Manual transfer of payload data, buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the local RAM/global RAM

25.2 Registers

25.2.1 List of Registers

The FLXAn registers are listed in the following table.

For details on <FLXAn_base>, see **Section 25.1.2, Register Base Address**.

Table 25.19 List of Registers

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay operation control register	FLXAnFROC	<FLXAn_base> + 0004_H
	FlexRay operation status register	FLXAnFROS	<FLXAn_base> + $000C_H$
	FlexRay test register 1	FLXAnFRTEST1	<FLXAn_base> + 0010_H
	FlexRay test register 2	FLXAnFRTEST2	<FLXAn_base> + 0014_H
	FlexRay lock register	FLXAnFRLCK	<FLXAn_base> + $001C_H$
	FlexRay error interrupt register	FLXAnFREIR	<FLXAn_base> + 0020_H
	FlexRay status interrupt register	FLXAnFRSIR	<FLXAn_base> + 0024_H
	FlexRay error interrupt line select register	FLXAnFREILS	<FLXAn_base> + 0028_H
	FlexRay status interrupt line select register	FLXAnFRSILS	<FLXAn_base> + $002C_H$
	FlexRay error interrupt enable set register	FLXAnFREIES	<FLXAn_base> + 0030_H
	FlexRay error interrupt enable reset register	FLXAnFREIER	<FLXAn_base> + 0034_H
	FlexRay status interrupt enable set register	FLXAnFRSIES	<FLXAn_base> + 0038_H
	FlexRay status interrupt disable register	FLXAnFRSIER	<FLXAn_base> + $003C_H$
	FlexRay interrupt line enable register	FLXAnFRILE	<FLXAn_base> + 0040_H
	FlexRay timer 0 configuration register	FLXAnFRT0C	<FLXAn_base> + 0044_H
	FlexRay timer 1 configuration register	FLXAnFRT1C	<FLXAn_base> + 0048_H
	FlexRay stop watch register 1	FLXAnFRSTPW1	<FLXAn_base> + $004C_H$
	FlexRay stop watch register 2	FLXAnFRSTPW2	<FLXAn_base> + 0050_H
	FlexRay SUC configuration register 1	FLXAnFRSUCC1	<FLXAn_base> + 0080_H
	FlexRay SUC configuration register 2	FLXAnFRSUCC2	<FLXAn_base> + 0084_H
	FlexRay SUC configuration register 3	FLXAnFRSUCC3	<FLXAn_base> + 0088_H
	FlexRay NEM configuration register	FLXAnFRNEMC	<FLXAn_base> + $008C_H$
	FlexRay PRT configuration register 1	FLXAnFRPRTC1	<FLXAn_base> + 0090_H
	FlexRay PRT configuration register 2	FLXAnFRPRTC2	<FLXAn_base> + 0094_H
	FlexRay MHD configuration register	FLXAnFRMHDC	<FLXAn_base> + 0098_H
	FlexRay GTU configuration register 1	FLXAnFRGTUC1	<FLXAn_base> + $00A0_H$
	FlexRay GTU configuration register 2	FLXAnFRGTUC2	<FLXAn_base> + $00A4_H$
	FlexRay GTU configuration register 3	FLXAnFRGTUC3	<FLXAn_base> + $00A8_H$
	FlexRay GTU configuration register 4	FLXAnFRGTUC4	<FLXAn_base> + $00AC_H$
	FlexRay GTU configuration register 5	FLXAnFRGTUC5	<FLXAn_base> + $00B0_H$
	FlexRay GTU configuration register 6	FLXAnFRGTUC6	<FLXAn_base> + $00B4_H$
	FlexRay GTU configuration register 7	FLXAnFRGTUC7	<FLXAn_base> + $00B8_H$
	FlexRay GTU configuration register 8	FLXAnFRGTUC8	<FLXAn_base> + $00BC_H$
	FlexRay GTU configuration register 9	FLXAnFRGTUC9	<FLXAn_base> + $00C0_H$
	FlexRay GTU configuration register 10	FLXAnFRGTUC10	<FLXAn_base> + $00C4_H$
	FlexRay GTU configuration register 11	FLXAnFRGTUC11	<FLXAn_base> + $00C8_H$
	FlexRay CC status vector register	FLXAnFRCCSV	<FLXAn_base> + 0100_H
	FlexRay CC error vector register	FLXAnFRCCEV	<FLXAn_base> + 0104_H

Table 25.19 List of Registers

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay slot counter value register	FLXAnFRSCV	<FLXAn_base> + 0110_H
	FlexRay macrotick and cycle counter value register	FLXAnFRMTCCV	<FLXAn_base> + 0114_H
	FlexRay rate correction value register	FLXAnFRRCV	<FLXAn_base> + 0118_H
	FlexRay offset correction value register	FLXAnFROCV	<FLXAn_base> + $011C_H$
	FlexRay sync frame status register	FLXAnFRSFS	<FLXAn_base> + 0120_H
	FlexRay symbol window and NIT status register	FLXAnFRSWNIT	<FLXAn_base> + 0124_H
	FlexRay aggregated channel status register	FLXAnFRACS	<FLXAn_base> + 0128_H
	FlexRay even sync ID register m (m = 1 to 15)	FLXAnFRESIDm (m = 1 to 15)	<FLXAn_base> + 0130_H to <FLXAn_base> + 0168_H (<FLXAn_base> + 0130_H + (m – 1) × 4_H)
	FlexRay odd sync ID register m (m = 1 to 15)	FLXAnFROSIDm (m = 1 to 15)	<FLXAn_base> + 0170_H to <FLXAn_base> + $01A8_H$ (<FLXAn_base> + 0170_H + (m – 1) × 4_H)
	FlexRay network management vector register m (m = 1 to 3)	FLXAnFRNMVm (m = 1 to 3)	<FLXAn_base> + $01B0_H$ to <FLXAn_base> + $01B8_H$ (<FLXAn_base> + $01B0_H$ + (m – 1) × 4_H)
	FlexRay message RAM configuration register	FLXAnFRMRC	<FLXAn_base> + 0300_H
	FlexRay FIFO rejection filter register	FLXAnFRFRF	<FLXAn_base> + 0304_H
	FlexRay FIFO rejection filter mask register	FLXAnFRFRFM	<FLXAn_base> + 0308_H
	FlexRay FIFO critical level register	FLXAnFRFCL	<FLXAn_base> + $030C_H$
	FlexRay message handler status register	FLXAnFRMHDS	<FLXAn_base> + 0310_H
	FlexRay last dynamic transmit slot register	FLXAnFRLDTS	<FLXAn_base> + 0314_H
	FlexRay FIFO status register	FLXAnFRFSR	<FLXAn_base> + 0318_H
	FlexRay message handler constraints flags register	FLXAnFRMHDF	<FLXAn_base> + $031C_H$
	FlexRay transmission request register i (i = 1 to 4)	FLXAnFRTXRQi (i = 1 to 4)	<FLXAn_base> + 0320_H to <FLXAn_base> + $032C_H$ (<FLXAn_base> + 0320_H + (i – 1) × 4_H)
	FlexRay new data register i (i = 1 to 4)	FLXAnFRNDATi (i = 1 to 4)	<FLXAn_base> + 0330_H to <FLXAn_base> + $033C_H$ (<FLXAn_base> + 0330_H + (i – 1) × 4_H)
	FlexRay message buffer status changed register i (i = 1 to 4)	FLXAnFRMBSCi (i = 1 to 4)	<FLXAn_base> + 0340_H to <FLXAn_base> + $034C_H$ (<FLXAn_base> + 0340_H + (i – 1) × 4_H)
	FlexRay write data section register x (x = 1 to 64)	FLXAnFRWRDSx (x = 1 to 64)	<FLXAn_base> + 0400_H to <FLXAn_base> + $04FC_H$ (<FLXAn_base> + 0400_H + (x – 1) × 4_H)
	FlexRay write header section register 1	FLXAnFRWRHS1	<FLXAn_base> + 0500_H
	FlexRay write header section register 2	FLXAnFRWRHS2	<FLXAn_base> + 0504_H
	FlexRay write header section register 3	FLXAnFRWRHS3	<FLXAn_base> + 0508_H
	FlexRay input buffer command mask register	FLXAnFRIBCM	<FLXAn_base> + 0510_H
	FlexRay input buffer command request register	FLXAnFRIBCR	<FLXAn_base> + 0514_H
	FlexRay read data section register x (x = 1 to 64)	FLXAnFRRDDSx (x = 1 to 64)	<FLXAn_base> + 0600_H to <FLXAn_base> + $06FC_H$ (<FLXAn_base> + 0600_H + (x – 1) × 4_H)
	FlexRay read header section register 1	FLXAnFRRDHS1	<FLXAn_base> + 0700_H
	FlexRay read header section register 2	FLXAnFRRDHS2	<FLXAn_base> + 0704_H
	FlexRay read header section register 3	FLXAnFRRDHS3	<FLXAn_base> + 0708_H
	FlexRay message buffer status register	FLXAnFRMBS	<FLXAn_base> + $070C_H$
	FlexRay output buffer command mask register	FLXAnFROBCM	<FLXAn_base> + 0710_H

Table 25.19 List of Registers

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay output buffer command request register	FLXAnFROBCR	<FLXAn_base> + 0714_H
	FlexRay input transfer configuration register	FLXAnFRITC	<FLXAn_base> + 0800_H
	FlexRay output transfer configuration register	FLXAnFROTC	<FLXAn_base> + 0804_H
	FlexRay input pointer table base address register	FLXAnFRIBA	<FLXAn_base> + 0808_H
	FlexRay FIFO pointer table base address register	FLXAnFRFBA	<FLXAn_base> + $080C_H$
	FlexRay output pointer table base address register	FLXAnFROBA	<FLXAn_base> + 0810_H
	FlexRay input queue control register	FLXAnFRIQC	<FLXAn_base> + 0814_H
	FlexRay user input transfer request register	FLXAnFRUIR	<FLXAn_base> + 0818_H
	FlexRay user output transfer request register	FLXAnFRUOR	<FLXAn_base> + $081C_H$
	FlexRay Input Transfer Status Register	FLXAnFRITS	<FLXAn_base> + 0820_H
	FlexRay output transfer status register	FLXAnFROTS	<FLXAn_base> + 0824_H
	FlexRay access error status register	FLXAnFRAES	<FLXAn_base> + 0828_H
	FlexRay access error address register	FLXAnFRAEA	<FLXAn_base> + $082C_H$
	FlexRay message data available register i (i = 0 to 3)	FLXAnFRDAi (i = 0 to 3)	<FLXAn_base> + 0830_H to <FLXAn_base> + $083C_H$ (<FLXAn_base> + 0830_H + i × 4_H)
	FlexRay H-Bus configuration register	FLXAnFRAHBC	<FLXAn_base> + 0840_H
	FlexRay timer 2 configuration register	FLXAnFRT2C	<FLXAn_base> + 0844_H

25.2.2 FlexRay Operation Register

25.2.2.1 FLXAnFROC — FlexRay Operation Control Register

Access:	FLXAnFROC can be read or written in 32-bit units. FLXAnFROCL and FLXAnFROCH can be read or written in 16-bit units. FLXAnFROCLL and FLXAnFROCHL can be read or written in 8-bit units.
Address:	FLXAnFROC: <FLXAn_base> + 0004$_H$ FLXAnFROCL: <FLXAn_base> + 0004$_H$, FLXAnFROCH: <FLXAn_base> + 0006$_H$ FLXAnFROCLL: <FLXAn_base> + 0004$_H$, FLXAnFROCHL: <FLXAn_base> + 0006$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IE	T1IE	T0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OEP	—	—	—	—	—	BEC	OE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 25.20 FLXAnFROC Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IE	Timer 2 interrupt enable Bit 0: Disabled 1: Enabled
17	T1IE	Timer 1 interrupt enable Bit 0: Disabled 1: Enabled
16	T0IE	Timer 0 interrupt enable Bit 0: Disabled 1: Enabled
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	OEP	Operation Enable bit Protection Bit 0: OE is unprotected 1: OE is protected
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	BEC	Byte Endian Control Bit 0: Little endian 1: Big endian
0	OE	Operation Enable Bit 0: Operation disabled, SW reset 1: Operation Enabled

(1) FLXAnFROC.T2IE

Timer 2 interrupt enable bit

This bit controls the timer 2 interrupt.

0: Disabled
No interrupt will be requested and the timer 2 interrupt line will be released if pending.

1: Enabled
Timer 2 interrupt will be asserted when FLXAnFROTS.T2IS is 1.

(2) FLXAnFROC.T1IE

Timer 1 interrupt enable bit

The user can only set this bit to 1 when timer 1 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.TI1E is 0).
This bit controls the timer 1 interrupt.

0: Disabled
No interrupt will be requested and the timer 1 interrupt line will be released if pending.

1: Enabled
Timer 1 interrupt will be asserted when FLXAnFROTS.T1IS is 1.

(3) FLXAnFROC.T0IE

Timer 0 interrupt enable bit

The user can only set this bit to 1 when timer 0 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.TI0E is 0).
This bit controls the timer 0 interrupt.

0: Disabled
No interrupt will be requested and the timer 0 interrupt line will be released if pending.

1: Enabled
Timer 0 interrupt will be asserted when FLXAnFROTS.T0IS is 1.

(4) FLXAnFROC.OEP

Operation Enable bit Protection bit

This bit protects against unintended write access to the OE bit.

0: OE is unprotected
Write access to the OE bit is enabled

1: OE is protected
Write access to the OE bit is disabled

(5) FLXAnFROC.BEC

Byte Endian Control bit

The user can only change this bit when FLXAnFROS.OS is 1.
This bit controls the byte order on reading and writing the FlexRay Network Management Vector register (FLXAnFRNMVm), FlexRay Write Data Section (FLXAnFRWRDSx) and FlexRay Read Data Section (FLXAnFRRDDSx). This bit also controls the byte order when reading or writing FlexRay payload data using the data transfer function.
For details about the byte alignment please refer to **Section 25.3.17, Byte Alignment**.

0: Little endian
Byte alignment in FLXAnFRNMVm, FLXAnFRWRDSx and FLXAnFRRDDSx is in little endian style.

1: Big endian
Byte alignment in FLXAnFRNMVm, FLXAnFRWRDSx and FLXAnFRRDDSx is in big endian style.

(6) FLXAnFROC.OE

Operation Enable bit

The user can only write to this bit when FLXAnFROC.OEP is 0.
The user should only write this bit with 0 when FLXAnFROS.OS is 1.
The user should only write this bit with 1 when FLXAnFROS.OS is 0 and the FlexRay sample clock is enabled.
This bit controls the operation state and executes the software reset of the FlexRay module. The operation status bit (FLXAnFROS.OS) indicates whether the FlexRay module is in reset state or not.

0: Operation disabled, SW reset
Forcibly moves the FlexRay module to its reset state, whatever the state of the FlexRay module is.

1: Operation Enabled
Reset state of the FlexRay module is released.

25.2.2.2 FLXAnFROS — FlexRay Operation Status Register

Access:	FLXAnFROS can be read or written in 32-bit units. FLXAnFROSL is a read-only register that can be read in 16-bit units. FLXAnFROSH can be read or written in 16-bit units. FLXAnFROSLL is a read-only register that can be read in 8-bit units. FLXAnFROSHL can be read or written in 8-bit units.
Address:	FLXAnFROS: <FLXAn_base> + 000C$_H$ FLXAnFROSL: <FLXAn_base> + 000C$_H$, FLXAnFROSH: <FLXAn_base> + 000E$_H$, FLXAnFROSLL: <FLXAn_base> + 000C$_H$, FLXAnFROSHL: <FLXAn_base> + 000E$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IS	T1IS	T0IS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.21 FLXAnFROS Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IS	Timer 2 Interrupt Status Bit 0: Timer 2 has not matched the conditions configured in the FLXAnFRT2C register 1: Timer 2 matched the conditions configured in the FLXAnFRT2C register
17	T1IS	Timer 1 Interrupt Status Bit 0: Timer 1 has not matched the conditions configured in the FLXAnFRT1C register 1: Timer 1 matched the conditions configured in the FLXAnFRT1C register
16	T0IS	Timer 0 Interrupt Status Bit 0: Timer 0 has not matched the conditions configured in the FLXAnFRT0C register 1: Timer 0 matched the conditions configured in the FLXAnFRT0C register
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OS	Operation Status Bit 0: Operation disabled, reset state 1: Operation enabled

(1) FLXAnFROS.T2IS

Timer 2 Interrupt Status Bit

Writing 0 has no effect on the bit value.
This bit indicates that the expiration criteria configured in FLXAnFRT2C register has matched the FlexRay local time.
If enabled in FLXAnFROC.T2IE the timer 2 interrupt is generated when FLXAnFROS.T2IS is 1.

[Clearing condition]
This bit is cleared by writing 1 to FLXAnFROS.T2IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]
This bit is set when the expiration criteria configured in FLXAnFRT2C matches the FlexRay local time.

(2) FLXAnFROS.T1IS

Timer 1 Interrupt Status Bit

Writing 0 has no effect on the bit value.
This bit indicates that the expiration criteria configured in FLXAnFRT1C register has matched the FlexRay local time.
If enabled in FLXAnFROC.T1IE the timer 1 interrupt is generated when FLXAnFROS.T1IS is 1.

[Clearing condition]
This bit is cleared by writing 1 to FLXAnFROS.T1IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]
This bit is set when the expiration criteria configured in FLXAnFRT1C matches the FlexRay local time.

(3) FLXAnFROS.T0IS

Timer 0 Interrupt Status Bit

Writing 0 has no effect on the bit value.
This bit indicates that the expiration criteria configured in FLXAnFRT0C register has matched the FlexRay local time.
If enabled in FLXAnFROC.T0IE the timer 0 interrupt is generated when FLXAnFROS.T0IS is 1.

[Clearing condition]
This bit is cleared by writing 1 to FLXAnFROS.T0IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]
This bit is set when the expiration criteria configured in FLXAnFRT0C matches the FlexRay local time.

(4) FLXAnFROS.OS

Operation Status Bit

This bit indicates if the FlexRay module is in the reset or the operation state.
When FLXAnFROS.OS is 0 the FlexRay module gets initialized and registers mapped to the address area <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ cannot be accessed; read access from these registers will return undefined data.
When FLXAnFROS.OS is 1 it is possible to access to the address area <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ and to perform FlexRay communication.
When FLXAnFROS.OS changes from 0 to 1 all registers in the address range <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ are set to the “Values after reset”.

[Clearing condition]
When FLXAnFROC.OE is set to 0. It takes up to two peripheral bus clock cycles until FLXAnFROS.OS is set to 0.

[Setting condition]
When FLXAnFROC.OE is set to 1 it takes up to four peripheral clock cycles of the clock with the lower frequency out of the FlexRay sample clock and peripheral bus clock until FLXAnFROS.OS is set to 1.

25.2.3 Special Registers

25.2.3.1 FLXAnFRTEST1 — FlexRay Test Register 1

The FlexRay Test Register 1 holds the control bits to configure the test modes of the FlexRay module.

When the FlexRay is operated in one of its test modes that requires FLXAnFRTEST1.WRTEN to be set (RAM Test Mode, I/O Test Mode, Asynchronous Transmit Mode, and Loop Back Mode) only the selected test mode functionality is available.

To return from test mode operation to regular FlexRay operation we strongly recommend to apply a reset using FLXAnFROC.OE to reset all FlexRay internal state machines to their initial state.

The test functions are not available in addition to the normal operational mode functions, they change the functions of parts of the FlexRay module. Therefore normal operation as specified outside this chapter and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions.

The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications.

Access: FLXAnFRTEST1 can be read or written in 32-bit units.

Address: FLXAnFRTEST1: <FLXAn_base> + 0010$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CERB[3:0]				CERA[3:0]				—	—	TXENB	TXENA	TXB	TXA	RXB	RXA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	AOB	AOA	—	—	TMC[1:0]		—	—	ELBE	WRTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 25.22 FLXAnFRTEST1 Register Contents

Bit Position	Bit Name	Function
31 to 28	CERB[3:0]	Coding Error Report Channel B 0000$_B$: No coding error detected 0001$_B$: Header CRC error detected 0010$_B$: Frame CRC error detected 0011$_B$: Frame Start Sequence FSS too long 0100$_B$: First bit of Byte Start Sequence BSS seen low-level 0101$_B$: Second bit of Byte Start Sequence BSS seen high-level 0110$_B$: First bit of Frame End Sequence FES seen high-level 0111$_B$: Second bit of Frame End Sequence FES seen low-level 1000$_B$: CAS/MTS symbol seen too short 1001$_B$: CAS/MTS symbol seen too long 1010$_B$ to 1111$_B$: reserved

Table 25.22 FLXAnFRTEST1 Register Contents

Bit Position	Bit Name	Function
27 to 24	CERA[3:0]	Coding Error Report Channel A 0000_B: No coding error detected 0001_B: Header CRC error detected 0010_B: Frame CRC error detected 0011_B: Frame Start Sequence FSS too long 0100_B: First bit of Byte Start Sequence BSS seen low-level 0101_B: Second bit of Byte Start Sequence BSS seen high-level 0110_B: First bit of Frame End Sequence FES seen high-level 0111_B: Second bit of Frame End Sequence FES seen low-level 1000_B: CAS/MTS symbol seen too short 1001_B: CAS/MTS symbol seen too long 1010_B to 1111_B: reserved
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	TXENB	Control of Channel B Transmit Enable Pin 0: FLXAnTENB pin drives a 0 1: FLXAnTENB pin drives a 1
20	TXENA	Control of Channel A Transmit Enable Pin 0: FLXAnTENA pin drives a 0 1: FLXAnTENA pin drives a 1
19	TXB	Control of Channel B Transmit Pin 0: FLXAnTXDB pin drives a 0 1: FLXAnTXDB pin drives a 1
18	TXA	Control of Channel A Transmit Pin 0: FLXAnTXDA pin drives a 0 1: FLXAnTXDA pin drives a 1
17	RXB	Monitor Channel B Receive Pin 0: FLXAnRXDB is 0 1: FLXAnRXDB is 1
16	RXA	Monitor Channel A Receive Pin 0: FLXAnRXDA is 0 1: FLXAnRXDA is 1
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	AOB	Activity on B 0: No activity detected, channel B idle 1: Activity detected, channel B not idle
8	AOA	Activity on A 0: No activity detected, channel A idle 1: Activity detected, channel A not idle
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	TMC[1:0]	Test Multiplexer Control 00_B: Normal signal path 01_B: RAM Test Mode 10_B: I/O Test Mode 11_B: Normal signal path
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ELBE	External Loop Back Enable 0: Internal Loop Back 1: External Loop Back
0	WRTEN	Write Test Register Enable 0: Write access to test registers disabled 1: Write access to test registers enabled

(1) FLXAnFRTEST1.CERB

Coding Error Report Channel B

Set when a coding error is detected on channel B.
Reset to zero when register FLXAnFRTEST1 is read or written. Once the FLXAnFRTEST1.CERB is set it will remain unchanged until FLXAnFRTEST1 register is accessed.

NOTES

1. Coding errors are signalled in all states where frame decoding is possible.
   FLXAnFRTEST1.CERB should be ignored in all other states.
2. The error codes regarding CAS/MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

(2) FLXAnFRTEST1.CERA

Coding Error Report Channel A

Set when a coding error is detected on channel A.
Reset to zero when register FLXAnFRTEST1 is read or written. Once the FLXAnFRTEST1.CERA is set it will remain unchanged until FLXAnFRTEST1 register is accessed.

NOTES

1. Coding errors are signalled in all states where frame decoding is possible.
   FLXAnFRTEST1.CERA should be ignored in all other states.
2. The error codes regarding CAS/MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

(3) FLXAnFRTEST1.TXENB

Control of Channel B Transmit Enable Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTENB pin.

(4) FLXAnFRTEST1.TXENA

Control of Channel A Transmit Enable Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTENA pin.

(5) FLXAnFRTEST1.TXB

Control of Channel B Transmit Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTXDB pin.

(6) FLXAnFRTEST1.TXA

Control of Channel A Transmit Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTXDA pin.

(7) FLXAnFRTEST1.RXB

Monitor Channel B Receive Pin

This bit is used to test the interface to the physical layer (connectivity test) by reading the FLXAnRXDB pin.

(8) FLXAnFRTEST1.RXA

Monitor Channel A Receive Pin

This bit is used to test the interface to the physical layer (connectivity test) by reading the FLXAnRXDA pin.

(9) FLXAnFRTEST1.AOB

Activity on B

FLXAnFRTEST1.AOB is set when there is activity on channel B or if the POC state is in DEFAULT_CONFIG or CONFIG state.
During STARTUP, NORMAL_ACTIVE or NORMAL_PASSIVE the function of FLXAnFRTEST1.AOB is inverse of zChannelIdle as specified in *the FlexRay protocol specification V2.1, Section 3, BITSTRB process*.
FLXAnFRTEST1.AOB should be ignored in all other POC states.

(10) FLXAnFRTEST1.AOA

Activity on A

FLXAnFRTEST1.AOA is set when there is activity on channel A or if the POC state is in DEFAULT_CONFIG or CONFIG state.
During STARTUP, NORMAL_ACTIVE or NORMAL_PASSIVE the function of FLXAnFRTEST1.AOA is inverse of zChannelIdle as specified in *the FlexRay protocol specification V2.1, Section 3, BITSTRB process*.
FLXAnFRTEST1.AOA should be ignored in all other POC states.

(11) FLXAnFRTEST1.TMC

Test Multiplexer Control

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
00_B = Normal signal path
01_B = RAM Test Mode

Internal busses are multiplexed to make the Message RAM, Transient buffer RAM A and Transient buffer RAM B of the FlexRay module directly accessible by the Host. This mode is intended to enable testing of the FlexRay RAM during product testing.
10_B = I/O Test Mode

Output pins FLXAnTXDA, FLXAnTXDB, FLXAnTENA, FLXAnTENB, are driven to the values defined by bits FLXAnFRTEST1.TXA, FLXAnFRTEST1.TXB, FLXAnFRTEST1.TXENA, FLXAnFRTEST1.TXENB. The values applied to the input pins FLXAnRXDA, FLXAnRXDB can be read from register bits FLXAnFRTEST1.RXA, FLXAnFRTEST1.RXB.
11_B = Normal signal path

(12) FLXAnFRTEST1.ELBE

External Loop Back Enable

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.
There are two possibilities to perform a Loop Back test. External Loop Back via physical layer or internal Loop Back for in-system self-test.
In case of an internal Loop Back the FlexRay IP module pins FLXAnTENA, FLXAnTENB are in their inactive state, pins FLXAnTXDA, FLXAnTXDB are set to HIGH, pins FLXAnRXDA, FLXAnRXDB are not evaluated. Bit FLXAnFRTEST1.ELBE is evaluated only when POC is in Loop Back Mode and test multiplexer control is in non-multiplexing mode FLXAnFRTEST1.TMC = 00_B.

(13) FLXAnFRTEST1.WRTEN

Write Test Register Enable

To set the bit from 0 to 1 the test mode key has to be written as defined in **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. Enables write access to the test registers. The unlock sequence is not required when FLXAnFRTEST1.WRTEN is kept at 1 while other bits of the register are changed.
The bit can be reset to 0 at any time.

25.2.3.2 FLXAnFRTEST2 — FlexRay Test Register 2

The FlexRay Test Register 2 holds all bits required for the RAM test of the Message RAM, Transient buffer RAM A and Transient buffer RAM B of the FlexRay module.

Access: FLXAnFRTEST2 can be read or written in 32-bit units.

Address: FLXAnFRTEST2: <FLXAn_base> + 0014$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SSEL[2:0]			—	RS[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 25.23 FLXAnFRTEST2 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	These bits are always read as 0. The write value should be always 0.
6 to 4	SSEL[2:0]	Segment Select 000$_B$: Access to RAM bytes 0000$_H$ to 03FF$_H$ enabled 001$_B$: Access to RAM bytes 0400$_H$ to 07FF$_H$ enabled 010$_B$: Access to RAM bytes 0800$_H$ to 0BFF$_H$ enabled 011$_B$: Access to RAM bytes 0C00$_H$ to 0FFF$_H$ enabled 100$_B$: Access to RAM bytes 1000$_H$ to 13FF$_H$ enabled 101$_B$: Access to RAM bytes 1400$_H$ to 17FF$_H$ enabled 110$_B$: Access to RAM bytes 1800$_H$ to 1BFF$_H$ enabled 111$_B$: Access to RAM bytes 1C00$_H$ to 1FFF$_H$ enabled
3	Reserved	This bit is always read as 0. The write value should be always 0.
2 to 0	RS[2:0]	RAM Select 000$_B$ = Unused 001$_B$ = Unused 010$_B$ = Unused 011$_B$ = Unused 100$_B$ = Transient buffer RAM A (TBF1) 101$_B$ = Transient buffer RAM B (TBF2) 110$_B$ = Message RAM (MBF) 111$_B$ = Unused

(1) FLXAnFRTEST2.SSEL

Segment Select

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to “1”.
To enable access to the complete Message RAM (8192 byte addresses) the Message RAM is segmented into portions of 1024 bytes.

(2) FLXAnFRTEST2.RS

RAM Select

Write access to this register is only possible when FLXAnFRTEST1.WRTEN is set to “1”.
In RAM Test mode the RAM blocks selected by FLXAnFRTEST2.RS are mapped to module address <FLXAn_base> + 0400_H to ERAY + $07FF_H$ (1024 byte addresses).

25.2.3.3 FLXAnFRLCK — FlexRay Lock Register

Access: FLXAnFRLCK can be read or written in 32-bit units.
FLXAnFRLCKL can be read or written in 16-bit units.
FLXAnFRLCKLL can be read or written in 8-bit units.

Address: FLXAnFRLCK: <FLXAn_base> + $001C_H$,
FLXAnFRLCKL: <FLXAn_base> + $001C_H$,
FLXAnFRLCKLL: <FLXAn_base> + $001C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.24 FLXAnFRLCK Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CLK[7:0]	Configuration Lock Key Bit

(1) FLXAnFRLCK.CLK

Configuration Lock Key Bit

The Lock Register is write-only. Reading the register will return $0000\ 0000_H$.
To leave CONFIG state by writing FLXAnFRSUCC1.CMD (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the FLXAnFRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: FLXAnFRLCK.CLK = “$1100\ 1110_B$” (CE_H)
Second write: FLXAnFRLCK.CLK = “$0011\ 0001_B$” (31_H)
Third write: FLXAnFRSUCC1.CMD

CAUTION

If the Host uses 8/16-bit accesses to write the listed bit fields, the user has to ensure that no dummy accesses to the remaining register bytes/words are inserted by the compiler.

25.2.4 Interrupt Registers

25.2.4.1 FLXAnFREIR — FlexRay Error Interrupt Register

The flags are set when the CC detects one of the listed error conditions. The flags remain set until cleared.

Access: FLXAnFREIR can be read or written in 32-bit units.
FLXAnFREIRL and FLXAnFREIRH can be read or written in 16-bit units.
FLXAnFREIRLL, FLXAnFREIRLH, FLXAnFREIRHL, and FLXAnFREIRHH can be read or written in 8-bit units.

Address: FLXAnFREIR: <FLXAn_base> + 0020_H,
FLXAnFREIRL: <FLXAn_base> + 0020_H, FLXAnFREIRH: <FLXAn_base> + 0022_H,
FLXAnFREIRLL: <FLXAn_base> + 0020_H, FLXAnFREIRLH: <FLXAn_base> + 0021_H,
FLXAnFREIRHL: <FLXAn_base> + 0022_H, FLXAnFREIRHH: <FLXAn_base> + 0023_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABB	LTVB	EDB	—	—	—	—	—	TABA	LTVA	EDA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHF	IOBA	IIBA	EFA	RFO	AERR	CCL	CCF	SFO	SFBM	CNA	PEMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.25 FLXAnFREIR Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABB	Transmission Across Boundary Channel B Flag 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B
25	LTVB	Latest Transmit Violation Channel B Flag 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B
24	EDB	Error Detected on Channel B Flag 0: No error detected on channel B 1: Error detected on channel B
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABA	Transmission Across Boundary Channel A Flag 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A
17	LTVA	Latest Transmit Violation Channel A Flag 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A
16	EDA	Error Detected on Channel A Flag 0: No error detected on channel A 1: Error detected on channel A
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 25.25 FLXAnFREIR Register Contents

Bit Position	Bit Name	Function
11	MHF	Message Handler Constraints Flag 0: No message handler constraint violation detected 1: Message handler constraint violation detected
10	IOBA	Illegal Output buffer Access Flag 0: No illegal Host access to Output Buffer occurred 1: Illegal Host access to Output Buffer occurred
9	IIBA	Illegal Input Buffer Access Flag 0: No illegal Host access to Input Buffer occurred 1: Illegal Host access to Input Buffer occurred
8	EFA	Empty FIFO Access Flag 0: No access to empty FIFO occurred 1: Access to empty FIFO occurred
7	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
6	AERR	Access error Flag 0: Access error is not detected. 1: Access error is detected.
5	CCL	CHI Command Locked Flag 0: CHI command not accepted 1: CHI command accepted
4	CCF	Clock Correction Failure Flag 0: No clock correction error 1: Clock correction failed
3	SFO	Sync Frame Overflow Flag 0: Number of received sync frames ≤ FLXAnFRGTUC2 1: More sync frames received than configured by FLXAnFRGTUC2
2	SFBM	Sync Frames Below Minimum Flag 0: Sync node: 1 or more sync frames received, non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received
1	CNA	Command Not Accepted Flag 0: CHI command accepted 1: CHI command not accepted
0	PEMC	POC Error Mode Changed Flag 0: Error mode has not changed 1: Error mode has changed

(1) FLXAnFREIR.TABB

Transmission Across Boundary Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates that a transmission across a slot boundary occurred for channel B.

(2) FLXAnFREIR.LTVB

Latest Transmit Violation Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates to the CPU that a latest transmission violation occurred on channel B.

(3) FLXAnFREIR.EDB

Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This bit is set whenever one of the flags FLXAnFRACS.SEDB, FLXAnFRACS.CEDB, FLXAnFRACS.CIB, FLXAnFRACS.SBVB changes from 0 to 1.

(4) FLXAnFREIR.TABA

Transmission Across Boundary Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates that a transmission across a slot boundary occurred for channel A.

(5) FLXAnFREIR.LTVA

Latest Transmit Violation Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates to the CPU that a latest transmit violation occurred on channel A.

(6) FLXAnFREIR.EDA

Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This bit is set whenever one of the flags FLXAnFRACS.SEDA, FLXAnFRACS.CEDA, FLXAnFRACS.CIA, FLXAnFRACS.SBVA changes from 0 to 1.

(7) FLXAnFREIR.MHF

Message Handler Constraints Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates a Message Handler constraint violation condition. It is set whenever one of the flags FLXAnFRMHDF.SNUA, FLXAnFRMHDF.SNUB, FLXAnFRMHDF.FNFA, FLXAnFRMHDF.FNFB, FLXAnFRMHDF.TBFA, FLXAnFRMHDF.TBFB, FLXAnFRMHDF.WAHP changes from 0 to 1.

(8) FLXAnFREIR.IOBA

Illegal Output buffer Access Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while FLXAnFROBCR.OBSYS is set to 1.

(9) FLXAnFREIR.IIBA

Illegal Input Buffer Access Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies:

1. The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the
    - Header section of message buffer 0 or 1 configured as a key slot
    - Header section of static message buffers with buffer number < FLXAnFRMRC.FDB while FLXAnFRMRC.SEC = “01_B”
    - Header section of any static or dynamic message buffer while FLXAnFRMRC.SEC = “$1x_B$”
    - Header and/or data section of any message buffer belonging to the receive FIFO
2. The Host writes to any register of the Input Buffer while FLXAnFRIBCR.IBSYH is set to 1.

(10) FLXAnFREIR.EFA

Empty FIFO Access Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty.

(11) FLXAnFREIR.RFO

Receive FIFO Overrun Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with a new message. The current state of the FIFO is monitored in register FLXAnFRFSR.

(12) FLXAnFREIR.AERR

Access error flag Flag

Writing 0 in this bit has no effect.
This bit is cleared when writing 1 to it.
Notifies of an access error.
When the AMR, ATBF1, or ATBF2 bit in the FLXAnFRMHDS register changes from 0 to 1, this bit is set to 1.

(13) FLXAnFREIR.CCL

CHI Command Locked Flag

Writing 0 has no effect on the bit value. This bit is cleared by writing 1 to it.
The flag indicates that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because the execution of the previous CHI command has not yet completed. In this case bit FLXAnFREIR.CNA is also set to 1.

(14) FLXAnFREIR.CCF

Clock Correction Failure Flag

Writing 0 has no effect on the bit value. This bit is cleared by writing 1 to it.
This flag is set at the end of the communication cycle whenever one of the following errors occurs:

- Offset and/or rate correction incomplete
- Clock correction limit exceeded

The clock correction status is monitored in registers FLXAnFRCCEV and FLXAnFRSFS. A failure may occur during startup, therefore bit FLXAnFREIR.CCF should be set to 0 after the CC entered NORMAL_ACTIVE state.

(15) FLXAnFREIR.SFO

Sync Frame Overflow Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
It is set to 1 when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last two cycles exceeds the maximum number of sync frames as defined by FLXAnFRGTUC2.SNM.

(16) FLXAnFREIR.SFBM

Sync Frames Below Minimum Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 at the end of a cycle if the number of sync frames received during the last communication cycle was below the limit required for rate or offset correction term calculation (i.e. offset and/or rate correction incomplete). The clock correction status is monitored in FLXAnFRCCEV and FLXAnFRSFS.
This flag may be set to 1 during startup. Therefore this flag should be set to 0 by the Host after the CC entered NORMAL_ACTIVE state.

(17) FLXAnFREIR.CNA

Command Not Accepted Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
The flag indicates that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (FLXAnFREIR.CCL = 1).

(18) FLXAnFREIR.PEMC

POC Error Mode Changed Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 whenever the error mode signaled by FLXAnFRCCEV.ERRM changes.

25.2.4.2 FLXAnFRSIR — FlexRay Status Interrupt Register

The flags are set when the CC detects one of the listed events. The flags remain set until cleared.

Access:	FLXAnFRSIR can be read or written in 32-bit units. FLXAnFRSIRL and FLXAnFRSIRH can be read or written in 16-bit units. FLXAnFRSIRLL, FLXAnFRSIRLH, FLXAnFRSIRHL, and FLXAnFRSIRHH can be read or written in 8-bit units.
Address:	FLXAnFRSIR: <FLXAn_base> + 0024_H, FLXAnFRSIRL: <FLXAn_base> + 0024_H, FLXAnFRSIRH: <FLXAn_base> + 0026_H, FLXAnFRSIRLL: <FLXAn_base> + 0024_H, FLXAnFRSIRLH: <FLXAn_base> + 0025_H, FLXAnFRSIRHL: <FLXAn_base> + 0026_H, FLXAnFRSIRHH: <FLXAn_base> + 0027_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSB	WUPB	—	—	—	—	—	—	MTSA	WUPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.26 FLXAnFRSIR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSB	MTS Received on Channel B Flag (vSS!ValidMTSB) 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
24	WUPB	Wakeup Pattern Channel B Flag 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSA	MTS Received on Channel A Flag (vSS!ValidMTSA) 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
16	WUPA	Wakeup Pattern Channel A Flag 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A
15	SDS	Start of Dynamic Segment Flag 0: Dynamic segment not yet started 1: Dynamic segment started
14	MBSI	Message Buffer Status Interrupt Flag 0: No message buffer status change of message buffer with MBI = 1 1: Message buffer status of at least one message buffer with MBI = 1 has changed
13	SUCS	Startup Completed Successfully Flag 0: No startup completed successfully 1: Startup completed successfully
12	SWE	Stop Watch Event Flag 0: No Stop Watch Event 1: Stop Watch Event occurred

Table 25.26 FLXAnFRSIR Register Contents

Bit Position	Bit Name	Function
11	TOBC	OBF Transfer Complete Flag 0: No transfer completed 1: Transfer between Message RAM and Output Buffer completed
10	TIBC	IBF Transfer Complete Flag 0: No transfer completed 1: Transfer between Input Buffer and Message RAM completed
9	TI1	Timer Interrupt 1 Flag 0: No timer interrupt 1 1: Timer interrupt 1 occurred
8	TI0	Timer Interrupt 0 Flag 0: No timer interrupt 0 1: Timer interrupt 0 occurred
7	NMVC	Network Management Vector Changed Flag 0: No change in the network management vector 1: Network management vector changed
6	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
5	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty
4	RXI	Receive Interrupt Flag 0: No ND flag of a receive buffer with MBI = 1 has been set to 1 1: At least one ND flag of a receive buffer with MBI = 1 has been set to 1
3	TXI	Transmit Interrupt Flag 0: No frame transmitted from a transmit buffer with MBI = 1 1: At least one frame was transmitted from a transmit buffer with MBI = 1
2	CYCS	Cycle Start Interrupt Flag 0: No communication cycle started 1: Communication cycle started
1	CAS	Collision Avoidance Symbol Flag 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received
0	WST	Wakeup Status Flag 0: Wakeup status unchanged 1: Wakeup status changed

(1) FLXAnFRSIR.MTSB

MTS Received on Channel B Flag (vSS!ValidMTSB)

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Indicates that a Media Access Test Symbol was received on channel B during the preceding symbol window.
Updated by the CC for each channel at the end of the symbol window.

(2) FLXAnFRSIR.WUPB

Wakeup Pattern Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 when a wakeup pattern was received on channel B in either of the following states:

- WAKEUP
- READY
- STARTUP state

(3) FLXAnFRSIR.MTSA

MTS Received on Channel A Flag (vSS!ValidMTSA)

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Indicates that a Media Access Test Symbol was received on channel A during the preceding symbol window.
Updated by the CC for each channel at the end of the symbol window.

(4) FLXAnFRSIR.WUPA

Wakeup Pattern Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 when a wakeup pattern was received on channel A in either of the following states:

- WAKEUP
- READY
- STARTUP state

(5) FLXAnFRSIR.SDS

Start of Dynamic Segment Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when the dynamic segment starts.

(6) FLXAnFRSIR.MBSI

Message Buffer Status Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when the message buffer status FLXAnFRMBS has changed and if bit MBI of that message buffer is 1 (see **Table 25.119, Header Section of a Message Buffer in the Message RAM**).

(7) FLXAnFRSIR.SUCS

Startup Completed Successfully Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

(8) FLXAnFRSIR.SWE

Stop Watch Event Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set after a stop watch activation when the current cycle counter and macrotick value are stored in the Stop Watch register (see **Section 25.2.5.4, FLXAnFRSTPW1 — FlexRay Stop Watch Register 1**).

(9) FLXAnFRSIR.TOBC

Transfer Output Buffer Completed Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and FLXAnFROBCR.OBSYS has been cleared by the Message Handler.

(10) FLXAnFRSIR.TIBC

Transfer Input Buffer Completed Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set whenever a transfer from the message RAM to the input buffer has completed and FLXAnFRIBCR.IBSYS has been cleared by the Message Handler.

(11) FLXAnFRSIR.TI1

Timer 1 Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set whenever timer 1 matches the conditions configured in register FLXAnFRT1C. FlexRay timer 1 interrupt is generated when the T1IE bit in the FLXAnFROC register is effective.

(12) FLXAnFRSIR.TI0

Timer 0 Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set whenever timer 0 matches the conditions configured in register FLXAnFRT0C. FlexRay timer 0 interrupt is generated when the T0IE bit in the FLXAnFROC register is effective.

(13) FLXAnFRSIR.NMVC

Network Management Vector Changed Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This is set when a change in the Network Management Vector occurs.

(14) FLXAnFRSIR.RFCL

Receive FIFO Critical Level Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set when the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level as configured by FLXAnFRFCL.CL.

(15) FLXAnFRSIR.RFNE

Receive FIFO Not Empty Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The current state of the receive FIFO is monitored in register FLXAnFRFSR.

(16) FLXAnFRSIR.RXI

Receive Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see **Section 25.2.9.6, FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4)**), and if bit MBI of that message buffer is set to 1 (see **Table 25.119, Header Section of a Message Buffer in the Message RAM**)

(17) FLXAnFRSIR.TXI

Transmit Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC at the end of frame transmission if bit MBI in the respective message buffer is set to 1 (see **Table 25.119, Header Section of a Message Buffer in the Message RAM**).

(18) FLXAnFRSIR.CYCS

Cycle Start Interrupt Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when a communication cycle starts.

(19) FLXAnFRSIR.CAS

Collision Avoidance Symbol Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received.

(20) FLXAnFRSIR.WST

Wakeup Status Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set when FLXAnFRCCSV.WSV changes to a value other than UNDEFINED.

25.2.4.3 FLXAnFREILS — FlexRay Error Interrupt Line Select Register

The FlexRay Error Interrupt Line Select register assigns an interrupt generated by a specific error interrupt flag from register FLXAnFREIR to one of the two modules interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access:	FLXAnFREILS can be read or written in 32-bit units. FLXAnFREILSL and FLXAnFREILSH can be read or written in 16-bit units. FLXAnFREILSLL, FLXAnFREILSLH, FLXAnFREILSHL, and FLXAnFREILSHH can be read or written in 8-bit units.
Address:	FLXAnFREILS: <FLXAn_base> + 0028$_H$, FLXAnFREILSL: <FLXAn_base> + 0028$_H$, FLXAnFREILSH: <FLXAn_base> + 002A$_H$, FLXAnFREILSLL: <FLXAn_base> + 0028$_H$, FLXAnFREILSLH: <FLXAn_base> + 0029$_H$, FLXAnFREILSHL: <FLXAn_base> + 002A$_H$, FLXAnFREILSHH: <FLXAn_base> + 002B$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBL	LTVBL	EDBL	—	—	—	—	—	TABAL	LTVAL	EDAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFL	IOBAL	IIBAL	EFAL	RFOL	AERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.27 FLXAnFREILS Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBL	Transmission Across Boundary Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
25	LTVBL	Latest Transmit Violation Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	EDBL	Error Detected on Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAL	Transmission Across Boundary Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
17	LTVAL	Latest Transmit Violation Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	EDAL	Error Detected on Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFL	Message Handler Constraints Flag Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 25.27 FLXAnFREILS Register Contents

Bit Position	Bit Name	Function
10	IOBAL	Illegal Output Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	IIBAL	Illegal Input Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
8	EFAL	Empty FIFO Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	RFOL	Receive FIFO Overrun Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	AERRL	Access Error Interrupt Output Select Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	CCLL	CHI Command Locked Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	CCFL	Clock Correction Failure Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	SFOL	Sync Frame Overflow Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	SFBML	Sync Frames Below Minimum Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CNAL	Command Not Accepted Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	PEMCL	POC Error Mode Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

25.2.4.4 FLXAnFRSILS — FlexRay Status Interrupt Line Select Register

The FlexRay Status Interrupt Line Select register assign an interrupt generated by a specific status interrupt flag from register FLXAnFRSIR to one of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access:	FLXAnFRSILS can be read or written in 32-bit units. FLXAnFRSILSL and FLXAnFRSILSH can be read or written in 16-bit units. FLXAnFRSILSLL, FLXAnFRSILSLH, FLXAnFRSILSHL, and FLXAnFRSILSHH can be read or written in 8-bit units.
Address:	FLXAnFRSILS: <FLXAn_base> + $002C_H$, FLXAnFRSILSL: <FLXAn_base> + $002C_H$, FLXAnFRSILSH: <FLXAn_base> + $002E_H$, FLXAnFRSILSLL: <FLXAn_base> + $002C_H$, FLXAnFRSILSLH: <FLXAn_base> + $002D_H$, FLXAnFRSILSHL: <FLXAn_base> + $002E_H$, FLXAnFRSILSHH: <FLXAn_base> + $002F_H$
Value after reset:	0303 $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBL	WUPBL	—	—	—	—	—	—	MTSAL	WUPAL
Value after reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.28 FLXAnFRSILS Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBL	Media Access Test Symbol Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	WUPBL	Wakeup Pattern Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAL	Media Access Test Symbol Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	WUPAL	Wakeup Pattern Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15	SDSL	Start of Dynamic Segment Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
14	MBSIL	Message Buffer Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
13	SUCSL	Startup Completed Successfully Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 25.28 FLXAnFRSILS Register Contents

Bit Position	Bit Name	Function
12	SWEL	Stop Watch Event Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
11	TOBCL	Transfer Output Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
10	TIBCL	Transfer Input Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	TI1L	Timer 1 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
8	TI0L	Timer 0 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	NMVCL	Network Management Vector Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	RFCLL	Receive FIFO Critical Level Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	RFNEL	Receive FIFO Not Empty Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	RXIL	Receive Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	TXIL	Transmit Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	CYCSL	Cycle Start Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CASL	Collision Avoidance Symbol Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	WSTL	Wakeup Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

25.2.4.5 FLXAnFREIES — FlexRay Error Interrupt Enable Set Register

The settings in the FlexRay Error Interrupt Enable Set register (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value. Writing a 1 sets the interrupt enable bit.

Access: FLXAnFREIES can be read or written in 32-bit units.
FLXAnFREIESL and FLXAnFREIESH can be read or written in 16-bit units.
FLXAnFREIESLL, FLXAnFREIESLH, FLXAnFREIESHL, and FLXAnFREIESHH can be read or written in 8-bit units.

Address: FLXAnFREIES: <FLXAn_base> + 0030$_H$,
FLXAnFREIESL: <FLXAn_base> + 0030$_H$, FLXAnFREIESH: <FLXAn_base> + 0032$_H$,
FLXAnFREIESLL: <FLXAn_base> + 0030$_H$, FLXAnFREIESLH: <FLXAn_base> + 0031$_H$,
FLXAnFREIESHL: <FLXAn_base> + 0032$_H$, FLXAnFREIESHH: <FLXAn_base> + 0033$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBE	LTVBE	EDBE	—	—	—	—	—	TABAE	LTVAE	EDAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFE	IOBAE	IIBAE	EFAE	RFOE	AERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.29 FLXAnFREIES Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBE	Transmission Across Boundary Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBE	Latest Transmit Violation Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBE	Error Detected on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAE	Transmission Across Boundary Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAE	Latest Transmit Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 25.29 FLXAnFREIES Register Contents

Bit Position	Bit Name	Function
16	EDAE	Error Detected on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFE	Message Handler Constraints Flag Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAE	Illegal Output Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAE	Illegal Input Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAE	Empty FIFO Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOE	Receive FIFO Overrun Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRE	Access Error Interrupt Enable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLE	CHI Command Locked Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFE	Clock Correction Failure Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOE	Sync Frame Overflow Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBME	Sync Frames Below Minimum Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAE	Command Not Accepted Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCE	POC Error Mode Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

25.2.4.6 FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register

The settings in the FlexRay Error Interrupt Enable Set register (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value. Writing a 1 clears the interrupt enable bit.

Access: FLXAnFREIER can be read or written in 32-bit units.
FLXAnFREIERL and FLXAnFREIERH can be read or written in 16-bit units.
FLXAnFREIERLL, FLXAnFREIERLH, FLXAnFREIERHL, and FLXAnFREIERHH can be read or written in 8- bit units.

Address: FLXAnFREIER: <FLXAn_base> + 0034$_H$,
FLXAnFREIERL: <FLXAn_base> + 0034$_H$, FLXAnFREIERH: <FLXAn_base> + 0036$_H$,
FLXAnFREIERLL: <FLXAn_base> + 0034$_H$, FLXAnFREIERLH: <FLXAn_base> + 0035$_H$,
FLXAnFREIERHL: <FLXAn_base> + 0036$_H$, FLXAnFREIERHH: <FLXAn_base> + 0037$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBD	LTVBD	EDBD	—	—	—	—	—	TABAD	LTVAD	EDAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFD	IOBAD	IIBAD	EFAD	RFOD	AERRD	CCLD	CCFD	SFOD	SFBMD	CNAD	PEMCD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.30 FLXAnFREIER Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBD	Transmission Across Boundary Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBD	Latest Transmit Violation Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBD	Error Detected on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAD	Transmission Across Boundary Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAD	Latest Transmit Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	EDAD	Error Detected on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 25.30 FLXAnFREIER Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFD	Message Handler Constraints Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAD	Illegal Output Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAD	Illegal Input Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAD	Empty FIFO Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOD	Receive FIFO Overrun Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRD	Access Error Interrupt Disable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLD	CHI Command Locked Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFD	Clock Correction Failure Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOD	Sync Frame Overflow Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBMD	Sync Frames Below Minimum Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAD	Command Not Accepted Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCD	POC Error Mode Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

25.2.4.7 FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register

The settings in the FlexRay Status Interrupt Enable Set register (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value. Writing a 1 sets the interrupt enable bit.

Access: FLXAnFRSIES can be read or written in 32-bit units.
FLXAnFRSIESL and FLXAnFRSIESH and can be read or written in 16-bit units.
FLXAnFRSIESLL, FLXAnFRSIESLH, FLXAnFRSIESHL, and FLXAnFRSIESHH can be read or written in 8-bit units.

Address: FLXAnFRSIES: <FLXAn_base> + 0038_H,
FLXAnFRSIESL: <FLXAn_base> + 0038_H, FLXAnFRSIESH: <FLXAn_base> + $003A_H$,
FLXAnFRSIESLL: <FLXAn_base> + 0038_H, FLXAnFRSIESLH: <FLXAn_base> + 0039_H,
FLXAnFRSIESHL: <FLXAn_base> + $003A_H$, FLXAnFRSIESHH: <FLXAn_base> + $003B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBE	WUPBE	—	—	—	—	—	—	MTSAE	WUPAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.31 FLXAnFRSIES Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBE	MTS Received on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBE	Wakeup Pattern Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAE	MTS Received on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAE	Wakeup Pattern Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSE	Start of Dynamic Segment Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
14	MBSIE	Message Buffer Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 25.31 FLXAnFRSIES Register Contents

Bit Position	Bit Name	Function
13	SUCSE	Startup Completed Successfully Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWEE	Stop Watch Event Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCE	Transfer Output Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCE	Transfer Input Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1E	Timer 1 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0E	Timer 0 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCE	Network Management Vector Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLE	Receive FIFO Critical Level Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNEE	Receive FIFO Not Empty Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXIE	Receive Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXIE	Transmit Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSE	Cycle Start Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASE	Collision Avoidance Symbol Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTE	Wakeup Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

25.2.4.8 FLXAnFRSIER — FlexRay Status Interrupt Disable Register

The settings in the FlexRay Status Interrupt Enable Set register (FLXAnFRSIES) and FlexRay Status Interrupt Disable (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value. Writing a 1 clears the interrupt enable bit.

Access:	FLXAnFRSIER can be read or written in 32-bit units. FLXAnFRSIERL and FLXAnFRSIERH can be read or written in 16-bit units. FLXAnFRSIERLL, FLXAnFRSIERLH, FLXAnFRSIERHL, and FLXAnFRSIERHH can be read or written in 8- bit units.
Address:	FLXAnFRSIER: <FLXAn_base> + 003C$_{H}$, FLXAnFRSIERL: <FLXAn_base> + 003C$_{H}$, FLXAnFRSIERH: <FLXAn_base> + 003E$_{H}$, FLXAnFRSIERLL: <FLXAn_base> + 003C$_{H}$, FLXAnFRSIERLH: <FLXAn_base> + 003D$_{H}$, FLXAnFRSIERHL: <FLXAn_base> + 003E$_{H}$, FLXAnFRSIERHH: <FLXAn_base> + 003F$_{H}$
Value after reset:	0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBD	WUPBD	—	—	—	—	—	—	MTSAD	WUPAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSD	MBSID	SUCSD	SWED	TOBCD	TIBCD	TI1D	TI0D	NMVCD	RFCLD	RFNED	RXID	TXID	CYCSD	CASD	WSTD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.32 FLXAnFRSIER Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBD	MTS Received on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBD	Wakeup Pattern Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAD	MTS Received on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAD	Wakeup Pattern Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSD	Start of Dynamic Segment Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
14	MBSID	Message Buffer Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 25.32 FLXAnFRSIES Register Contents

Bit Position	Bit Name	Function
13	SUCSD	Startup Completed Successfully Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWED	Stop Watch Event Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCD	Transfer Output Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCD	Transfer Input Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1D	Timer Interrupt 1 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0D	Timer Interrupt 0 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCD	Network Management Vector Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLD	Receive FIFO Critical Level Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNED	Receive FIFO Not Empty Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXID	Receive Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXID	Transmit Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSD	Cycle Start Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASD	Collision Avoidance Symbol Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTD	Wakeup Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

25.2.4.9 FLXAnFRILE — FlexRay Interrupt Line Enable Register

Each of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt) can be enabled/disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

Access:	FLXAnFRILE can be read or written in 32-bit units. FLXAnFRILEL can be read or written in 16-bit units. FLXAnFRILELL can be read or written in 8-bit units.
Address:	FLXAnFRILE: <FLXAn_base> + 0040$_H$, FLXAnFRILEL: <FLXAn_base> + 0040$_H$, FLXAnFRILELL: <FLXAn_base> + 0040$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EINT1	EINT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.33 FLXAnFRILE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	EINT1	Enable FlexRay 1 Interrupt Line Bit 0: FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt enabled
0	EINT0	Enable FlexRay 0 Interrupt Line Bit 0: FlexRay 0 interrupt disabled 1: FlexRay 0 interrupt enabled

25.2.5 FlexRay Timer Registers

25.2.5.1 FLXAnFRT0C — FlexRay Timer 0 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 0 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 0 passes, FLXAnFRSIR.TI0 and FLXAnFROS.T0IS are set to 1. A timer 0 interrupt then occurs while the FLXAnFROC.T0IE bit is effective.

CAUTION

The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0.

Access: FLXAnFRT0C can be read or written in 32-bit units.
FLXAnFRT0CL and FLXAnFRT0CH can be read or written in 16-bit units.
FLXAnFRT0CLL, FLXAnFRT0CLH, FLXAnFRT0CHL, and FLXAnFRT0CHH can be read or written in 8-bit units.

Address: FLXAnFRT0C: <FLXAn_base> + 0044$_H$,
FLXAnFRT0CL: <FLXAn_base> + 0044$_H$, FLXAnFRT0CH: <FLXAn_base> + 0046$_H$,
FLXAnFRT0CLL: <FLXAn_base> + 0044$_H$, FLXAnFRT0CLH: <FLXAn_base> + 0045$_H$,
FLXAnFRT0CHL: <FLXAn_base> + 0046$_H$, FLXAnFRT0CHH: <FLXAn_base> + 0047$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T0CC[6:0]							—	—	—	—	—	—	T0MS	T0RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.34 FLXAnFRT0C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T0MO[13:0]	Timer 0 Macrotick Offset Bit Timer 0 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T0CC[6:0]	Timer 0 Cycle Code Bit Timer 0 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T0MS	Timer 0 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T0RC	Timer 0 Run Control Bit 0: Timer 0 halted 1: Timer 0 running

(1) FLXAnFRT0C.T0MO

Timer 0 Macrotick Offset Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.
Configures the macrotick offset from the beginning of the communication cycle where the interrupt is to occur. The FlexRay timer 0 interrupt occurs at this offset for each cycle of the cycle set.

(2) FLXAnFRT0C.T0CC

Timer 0 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.
The 7-bit timer 0 cycle code determines the cycle set used for generation of the FlexRay timer 0 interrupt. For details about the configuration of the cycle code see **Section 25.3.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT0C.T0MS

Timer 0 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.
Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT0C.T0RC

Timer 0 Run Control Bit

Timer 0 can be activated (set FLXAnFRT0C.T0RC to 1) when the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.
Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

25.2.5.2 FLXAnFRT1C — FlexRay Timer 1 Configuration Register

This register is a relative timer. After the specified number of macroticks (MT) has expired, a FlexRay timer 1 interrupt is asserted. When the FlexRay timer 1 passes, FLXAnFRSIR.TI1 and FLXAnFROS.T1IS are set to 1. A timer 1 interrupt then occurs while the FLXAnFROC.T1IE bit is effective.

Access: FLXAnFRT1C can be read or written in 32-bit units.
FLXAnFRT1CL and FLXAnFRT1CH can be read or written in 16-bit units.
FLXAnFRT1CLL, FLXAnFRT1CHL, and FLXAnFRT1CHH can be read or written in 8-bit units.

Address: FLXAnFRT1C: <FLXAn_base> + 0048$_H$,
FLXAnFRT1CL: <FLXAn_base> + 0048$_H$, FLXAnFRT1CH: <FLXAn_base> + 004A$_H$,
FLXAnFRT1CLL: <FLXAn_base> + 0048$_H$, FLXAnFRT1CHL: <FLXAn_base> + 004A$_H$,
FLXAnFRT1CHH: <FLXAn_base> + 004B$_H$

Value after reset: 0002 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T1MC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	T1MS	T1RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.35 FLXAnFRT1C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T1MC[13:0]	Timer 1 Macrotick Count Bit
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T1MS	Timer 1 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T1RC	Timer 1 Run Control Bit 0: Timer 1 halted 1: Timer 1 running

(1) FLXAnFRT1C.T1MC

Timer 1 Macrotick Count Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to FLXAnFRT1C.T1RC.

- Valid values are 2 to 16383 MT in continuous mode
- Valid values are 1 to 16383 MT in single-shot mode

When the configured macrotick count is reached the FlexRay timer 1 interrupt is generated.

(2) FLXAnFRT1C.T1MS

Timer 1 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to FLXAnFRT1C.T1RC.
Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(3) FLXAnFRT1C.T1RC

Timer 1 Run Control Bit

Timer 1 can be activated (set FLXAnFRT1C.T1RC to 1) as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.
Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

25.2.5.3 FLXAnFRT2C — FlexRay Timer 2 Configuration Register

This register is an absolute timer. Timer 2 has the same absolute timer features as timer 0.

Access: FLXAnFRT2C can be read or written in 32-bit units.
FLXAnFRT2CL and FLXAnFRT2CH can be read or written in 16-bit units.
FLXAnFRT2CLL, FLXAnFRT2CLH, FLXAnFRT2CHL, and FLXAnFRT2CHH can be read or written in 8-bit units.

Address: FLXAnFRT2C: <FLXAn_base> + 0844$_H$
FLXAnFRT2CL: <FLXAn_base> + 0844$_H$, FLXAnFRT2CH: <FLXAn_base> + 0846$_H$,
FLXAnFRT2CLL: <FLXAn_base> + 0844$_H$, FLXAnFRT2CLH: <FLXAn_base> + 0845$_H$,
FLXAnFRT2CHL: <FLXAn_base> + 0846$_H$, FLXAnFRT2CHH: <FLXAn_base> + 0847$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T2MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T2CC[6:0]							—	—	—	—	—	—	T2MS	T2RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.36 FLXAnFRT2C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T2MO[13:0]	Timer 2 Macrotick Offset Bit Timer 2 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T2CC[6:0]	Timer 2 Cycle Code Bit Timer 2 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T2MS	Timer 2 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T2RC	Timer 2 Run Control Bit 0: Timer 2 halted 1: Timer 2 running

(1) FLXAnFRT2C.T2MO

Timer 2 Macrotick Offset Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.
Allowed range is 0 to FLXAnFRGTUC2.MPC.
These bits indicate the timer macrotick value defining the timer expiration condition.

(2) FLXAnFRT2C.T2CC

Timer 2 Cycle Code Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.
This bit indicates the cycle counter filter code defining the timer expiration condition.
See **Section 25.3.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT2C.T2MS

Timer 2 Mode Select Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.
This bit indicates the operation mode of the Timer 2.

0: Single-shot mode
The timer is operating in the non-repetitive (single shot) mode. Once the configured expiration criteria are matching the timer will be automatically halted.

1: Continuous mode
The timer is operating in the repetitive (continuous) mode. The timer will expire every time the configured expiration criteria are matching. The timer is not halted.

(4) FLXAnFRT2C.T2RC

Timer 2 Run Control Bit

The user can only set this bit to 1 when the POC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.
This bit indicates the activation state of the Timer 2.
When the expiration criteria are matching for a single shot timer, then this bit is cleared automatically and the Timer 2 is halted.

[Setting condition]
This bit is set by writing 1 to it when the POC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

[Clearing condition]
This bit is cleared when the POC state leaves the NORMAL_ACTIVE or NORMAL_PASSIVE state except for transitions between the two states.
This bit is cleared when the timer is operating in the non-repetitive (single shot) mode (FLXAnFRT2C.T2MS is 0) and the expiration criteria are matching.
This bit is cleared by writing 0 to it.

25.2.5.4 FLXAnFRSTPW1 — FlexRay Stop Watch Register 1

The stop watch is activated by the following trigger events.

- FlexRay 0 interrupt or FlexRay 1 interrupt
- Writing bit FLXAnFRSTPW1.SSWT to 1

At the first MT counter increment after the stop watch starts, the actual cycle counter and macrotick values are captured in register FLXAnFRSTPW1 while the slot counter values for channel A and B are captured in register FLXAnFRSTPW2.

Access: FLXAnFRSTPW1 can be read or written in 32-bit units.
FLXAnFRSTPW1L can be read or written in 16-bit units.
FLXAnFRSTPW1H is a read-only register that can be read in 16-bit units.
FLXAnFRSTPW1LL can be read or written in 8-bit units.
FLXAnFRSTPW1LH, FLXAnFRSTPW1HL, and FLXAnFRSTPW1HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSTPW1: <FLXAn_base> + 004C$_H$,
FLXAnFRSTPW1L: <FLXAn_base> + 004C$_H$, FLXAnFRSTPW1H: <FLXAn_base> + 004E$_H$,
FLXAnFRSTPW1LL: <FLXAn_base> + 004C$_H$, FLXAnFRSTPW1LH: <FLXAn_base> + 004D$_H$,
FLXAnFRSTPW1HL: <FLXAn_base> + 004E$_H$, FLXAnFRSTPW1HH: <FLXAn_base> + 004F$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SMTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SCCV[5:0]						—	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.37 FLXAnFRSTPW1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	SMTV[13:0]	Stop Watch Captured Macrotick Value
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	SCCV[5:0]	Stop Watch Captured Cycle Counter Value
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	EINT1	Enable FlexRay Interrupt 1 Trigger Bit 0: Stop watch trigger by FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt event triggers stop watch
5	EINT0	Enable FlexRay 0 Interrupt Trigger Bit 0: Stop watch trigger by FlexRay 0 interrupt disabled 1: FlexRay interrupt 0 event triggers stop watch
4	EETP	Enable External Trigger Pin 0: Stop watch trigger via MCU input pin (FLXA0STPWT) disabled 1: A rising or falling edge on MCU input pin (FLXA0STPWT) trigger stop watch

Table 25.37 FLXAnFRSTPW1 Register Contents

Bit Position	Bit Name	Function
3	SSWT	Software Stop Watch Trigger Bit 0: Software trigger reset 1: Stop watch activated by software trigger
2	EDGE	Stop Watch Trigger Edge Select Bit 0: Falling edge 1: Rising edge
1	SWMS	Stop Watch Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	ESWT	Enable Hardware Stop Watch Trigger Bit 0: Stop watch trigger disabled 1: Stop watch trigger enabled

(1) FLXAnFRSTPW1.SMTV

Stop Watch Captured Macrotick Value

Indicates the state of the macrotick counter when the stop watch event occurred.

(2) FLXAnFRSTPW1.SCCV

Stop Watch Captured Cycle Counter Value

Indicates the state of the cycle counter when the stop watch event occurred.

(3) FLXAnFRSTPW1.EINT1

Enable FlexRay 1 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 1 interrupt when FLXAnFRSTPW1.ESWT = 1.

(4) FLXAnFRSTPW1.EINT0

Enable FlexRay 0 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 0 interrupt when FLXAnFRSTPW1.ESWT = 1.

(5) FLXAnFRSTPW1.EETP

Enable External Trigger Pin

Enables stop watch trigger event via MCU pin FLXAnSTPWT if FLXAnFRSTPW1.ESWT = ‘1’.

(6) FLXAnFRSTPW1.SSWT

Software Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.
Writing 1 in this bit activates the stop watch. This bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

(7) FLXAnFRSTPW1.EDGE

Stop Watch Trigger Edge Select Bit

(8) FLXAnFRSTPW1.SWMS

Stop Watch Mode Select Bit

(9) FLXAnFRSTPW1.ESWT

Enable Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.
If enabled, a FlexRay 0 interrupt event or a FlexRay 1 interrupt event activates the stop watch.
In single-shot mode, this bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

25.2.5.5 FLXAnFRSTPW2 — FlexRay Stop Watch Register 2

Access: FLXAnFRSTPW2 is a read-only register that can be read in 32-bit units.
FLXAnFRSTPW2L and FLXAnFRSTPW2H are the read-only registers that can be read in 16-bit units.
FLXAnFRSTPW2LL, FLXAnFRSTPW2LH, FLXAnFRSTPW2HL, and FLXAnFRSTPW2HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSTPW2: <FLXAn_base> + 0050_H,
FLXAnFRSTPW2L: <FLXAn_base> + 0050_H, FLXAnFRSTPW2H: <FLXAn_base> + 0052_H,
FLXAnFRSTPW2LL: <FLXAn_base> + 0050_H, FLXAnFRSTPW2LH: <FLXAn_base> + 0051_H,
FLXAnFRSTPW2HL: <FLXAn_base> + 0052_H, FLXAnFRSTPW2HH: <FLXAn_base> + 0053_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SSCVB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SSCVA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.38 FLXAnFRSTPW2 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SSCVB[10:0]	Stop Watch Captured Slot Counter Value Channel B
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SSCVA[10:0]	Stop Watch Captured Slot Counter Value Channel A

(1) FLXAnFRSTPW2.SSCVB

Stop Watch Captured Slot Counter Value Channel B

Indicates the state of the slot counter value for channel B when the stop watch event occurred.

(2) FLXAnFRSTPW2.SSCVA

Stop Watch Captured Slot Counter Value Channel A

Indicates the state of the slot counter value for channel A when the stop watch event occurred.

25.2.6 CC Control Registers

This section describes the registers provided by the CC (Communication Controller) to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered after reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to execute the lock release sequence as described in **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**.

25.2.6.1 FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1

Access: FLXAnFRSUCC1 can be read or written in 32-bit units.
FLXAnFRSUCC1L and FLXAnFRSUCC1H can be read or written in 16-bit units.
FLXAnFRSUCC1LL, FLXAnFRSUCC1LH, FLXAnFRSUCC1HL, and FLXAnFRSUCC1HH can be read or written in 8-bit units.

Address: FLXAnFRSUCC1: <FLXAn_base> + 0080$_H$,
FLXAnFRSUCC1L: <FLXAn_base> + 0080$_H$, FLXAnFRSUCC1H: <FLXAn_base> + 0082$_H$,
FLXAnFRSUCC1LL: <FLXAn_base> + 0080$_H$, FLXAnFRSUCC1LH: <FLXAn_base> + 0081$_H$,
FLXAnFRSUCC1HL: <FLXAn_base> + 0082$_H$, FLXAnFRSUCC1HH: <FLXAn_base> + 0083$_H$

Value after reset: 0C40 1080$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CCHB	CCHA	MTSB	MTSA	HCSE	TSM	WUCS	PTA[4:0]				
Value after reset	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[4:0]					—	TXSY	TXST	PBSY	—	—	—	CMD[3:0]			
Value after reset	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.39 FLXAnFRSUCC1 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	CCHB	Connected to Channel B Bit Configures pChannels 0: Not connected to channel B 1: Node connected to channel B (default after reset)
26	CCHA	Connected to Channel A Bit Configures pChannels 0: Not connected to channel A 1: Node connected to channel A (default after reset)
25	MTSB	Select Channel B for MTS Transmission Bit 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission
24	MTSA	Select Channel A for MTS Transmission Bit 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission

Table 25.39 FLXAnFRSUCC1 Register Contents

Bit Position	Bit Name	Function
23	HCSE	Halt due to Clock Sync Error Bit Configures pAllowHaltDueToClock 0: CC will enter/remain in NORMAL_PASSIVE 1: CC will enter HALT state
22	TSM	Transmission Slot Mode Bit Configures pSingleSlotEnabled 0: ALL Slot Mode 1: SINGLE Slot Mode (value after hard reset)
21	WUCS	Wakeup Channel Select Bit Configures pWakeupChannel 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B
20 to 16	PTA[4:0]	Passive to Active Bit Configures pAllowPassiveToActive
15 to 11	CSA[4:0]	Cold Start Attempts Bit Configures gColdStartAttempts
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TXSY	Transmit Sync Frame in Key Slot Bit Configures pKeySlotUsedForSync 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node
8	TXST	Transmit Startup Frame in Key Slot Bit Configures pKeySlotUsedForStartup 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter
7	PBSY	POC Busy Flag 0: POC not busy, FLXAnFRSUCC1.CMD writeable 1: POC is busy, FLXAnFRSUCC1.CMD locked
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	CMD[3:0]	CHI Command Vector Bit 0000_B: command_not_accepted 0001_B: CONFIG 0010_B: READY 0011_B: WAKEUP 0100_B: RUN 0101_B: ALL_SLOTS 0110_B: HALT 0111_B: FREEZE 1000_B: SEND_MTS 1001_B: ALLOW_COLDSTART 1010_B: RESET_STATUS_INDICATORS 1011_B: MONITOR_MODE 1100_B: CLEAR_RAMS others: reserved

(1) FLXAnFRSUCC1.CCHB

Connected to Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Configures whether the node is connected to channel B (pChannels).

(2) FLXAnFRSUCC1.CCHA

Connected to Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Configures whether the node is connected to channel A (pChannels).

(3) FLXAnFRSUCC1.MTSB

Select Channel B for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
FLXAnFRSUCC1.MTSB may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1, an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = “1000_B”.
The bit selects channel B for MTS symbol transmission.

(4) FLXAnFRSUCC1.MTSA

Select Channel A for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
FLXAnFRSUCC1.MTSA may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1, an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = “1000_B”.
The bit selects channel A for MTS symbol transmission.

(5) FLXAnFRSUCC1.HCSE

Halt due to Clock Sync Error Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Controls the transition to HALT state due to a clock synchronization error (pAllowHaltDueToClock).

(6) FLXAnFRSUCC1.TSM

Transmission Slot Mode Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Sets the value after transmission slot mode reset (pSingleSlotEnabled).
In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 or message buffers 0 and 1 depending on bit FLXAnFRMRC.SPLM.
If FLXAnFRSUCC1.TSM = 1, message buffer 0 or message buffers 0 and 1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots.
FLXAnFRSUCC1.TSM is a configuration bit which can only be set/reset by the Host.

The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing FLXAnFRSUCC1.CMD = “0101_B” in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by FLXAnFRCCSV.SLM.

(7) FLXAnFRSUCC1.WUCS

Wakeup Channel Select Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
With this bit the Host selects the channel on which the CC sends the Wakeup pattern (pWakeupChannel).

(8) FLXAnFRSUCC1.PTA

Passive to Active Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 31 even/odd cycle pairs.
Defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state (pAllowPassiveToActive).
If set to “00000_B” the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(9) FLXAnFRSUCC1.CSA

Cold Start Attempts Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Must be identical in all nodes of a cluster. Valid values are 2 to 31.
Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node (gColdStartAttempts).

(10) FLXAnFRSUCC1.TXSY

Transmit Sync Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Defines whether the key slot is used to transmit sync frames (pKeySlotUsedForSync).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

---

(11) FLXAnFRSUCC1.TXST

Transmit Startup Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Defines whether the key slot is used to transmit startup frames (pKeySlotUsedForStartup).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

---

(12) FLXAnFRSUCC1.PBSY

POC Busy Flag

Signals that the POC is busy and cannot accept a command from the Host. FLXAnFRSUCC1.CMD is locked against write accesses.
Set to 1 after reset during initialization of internal RAM blocks.

(13) FLXAnFRSUCC1.CMD

CHI Command Vector Bit

The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted, and flag FLXAnFREIR.CNA will be set to 1.
In general the Host must check FLXAnFRSUCC1.PBSY before writing a new CHI command.
In case the previous CHI command has not yet completed, FLXAnFREIR.CCL is set to 1 together with FLXAnFREIR.CNA; the CHI command needs to be repeated.
Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will FLXAnFREIR.CNA be set.
Reading FLXAnFRSUCC1.CMD shows whether the last CHI command was accepted. The actual POC state is monitored by FLXAnFRCCSV.POCS.

- command_not_accepted
  FLXAnFRSUCC1.CMD is reset to "0000_B" due to one of the following conditions:
  - Illegal command applied by the Host
  - Host applied command to leave CONFIG state without preceding config lock key
  - Host applied new command while execution of the previous Host command has not completed
  - Host writes command_not_accepted

When FLXAnFRSUCC1.CMD is cleared to "0000_B", FLXAnFREIR.CNA is set to 1, and if enabled an interrupt is generated.
Commands which are not accepted are not executed.

CONFIG Command

Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, or READY. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

READY Command

Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

WAKEUP Command

Go to POC state WAKEUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

RUN Command

Go to POC state STARTUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to “0000_B” = command_not_accepted.

ALL_SLOTS Command

Leave SINGLE slot mode and go to ALL-SLOTS mode after successful startup/integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to “0000_B” = command_not_accepted.

HALT Command

Set halt request FLXAnFRCCSV.HRQ to 1 and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to “0000_B” = command_not_accepted.

FREEZE Command

Set the freeze status indicator FLXAnFRCCSV.FSI to 1 and go to POC state HALT immediately. Can be called from any state.

SEND_MTS Command

Send single MTS symbol during the next following symbol window on the channel configured by FLXAnFRSUCC1.MTSA, FLXAnFRSUCC1.MTSB, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (FLXAnFRCCSV.SLM = “11_B”). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

ALLOW_COLDSTART Command

The command clears FLXAnFRCCSV.CSI to enable the coldstart of the node. When called in states DEFAULT_CONFIG, CONFIG, or HALT, FLXAnFRSUCC1.CMD will be cleared to “0000_B” = command_not_accepted. To enable the coldstart it is also required that both FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set.

RESET_STATUS_INDICATORS Command

Clears status flags FLXAnFRCCSV.CSNI, FLXAnFRCCSV.CSAI, and FLXAnFRCCSV.WSV to their values after reset. May be called in POC states READY and STARTUP. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted.

CLEAR_RAMS Command

Sets FLXAnFRMHDS.CRAM to 1 when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted. FLXAnFRMHDS.CRAM is also set to 1 after reset. By setting FLXAnFRMHDS.CRAM all internal RAM blocks are initialized to zero. During the initialization of the RAMs, FLXAnFRSUCC1.PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS.

The initialization of the internal message RAM requires 2048 peripheral bus clock cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after reset or after assertion of CHI command CLEAR_RAMS.

Before asserting CHI command CLEAR_RAMS, make sure that no transfer between Message RAM and IBF/OBF or Message RAM and Temporary Buffer RAM is ongoing and that the data transfer handler has no effect (FLXAnFRITS.ITS = 0 and FLXAnFROTS.OTS = 0). This command also clears the Message Buffer Status registers FLXAnFRMHDS, FLXAnFRLDTS, FLXAnFRFSR, FLXAnFRMHDF, FLXAnFRTXRQ1/2/3/4, FLXAnFRNDAT1/2/3/4, and FLXAnFRMBSC1/2/3/4.

CAUTIONS

1. All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the FlexRay sample clock domain after at most 8 cycles of the slower of the two clocks “bus clock” and “FlexRay sample clock”, assuming that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXAnFRCCSV will show data that is additionally delayed by synchronization from the FlexRay sample clock domain to the bus clock domain. The maximum additional delay is 12 cycles of the slower of the two clocks “bus clock” and “FlexRay sample clock”.
2. When the communication is stopped by the FREEZE or READY command and the communication is restarted as Leading Coldstart node, the startup frame may not be transmitted on cycle 0 depending on the internal condition of the FlexRay module. This case occurs when the Startup frame is set in one of slot 1 to slot 7.
   This does not occur in ColdStart after a hardware reset.
   Even if this occurs, the second trial of ColdStart will succeed. ColdStart time becomes longer, but ColdStart will not be affected by the occurrence.
   To avoid this, allocate the Startup/Sync frame in the static slot 8 or higher.

Table 25.40, Reference to CHI Host Command Summary from FlexRay Protocol Specification below references the CHI commands from *the FlexRay Protocol Specification (Section 2.1.1.1, Table 2.2)* to the FlexRay CHI command vector FLXAnFRSUCC1.CMD.

Table 25.40 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI Command	Where Processed (POC States)	CHI Command Vector CMD
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active, POC: normal passive	HALT
READY	All except POC: default config, POC: config, POC: ready, POC: halt	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

25.2.6.2 FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2

Access: FLXAnFRSUCC2 can be read or written in 32-bit units.
FLXAnFRSUCC2L and FLXAnFRSUCC2H can be read or written in 16-bit units.
FLXAnFRSUCC2LL, FLXAnFRSUCC2LH, FLXAnFRSUCC2HL, and FLXAnFRSUCC2HH can be read or written in 8-bit units.

Address: FLXAnFRSUCC2: <FLXAn_base> + 0084_H
FLXAnFRSUCC2L: <FLXAn_base> + 0084_H, FLXAnFRSUCC2H: <FLXAn_base> + 0086_H,
FLXAnFRSUCC2LL: <FLXAn_base> + 0084_H, FLXAnFRSUCC2LH: <FLXAn_base> + 0085_H,
FLXAnFRSUCC2HL: <FLXAn_base> + 0086_H, FLXAnFRSUCC2HH: <FLXAn_base> + 0087_H

Value after reset: 0100 0504_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LTN[3:0]				—	—	—	LT[20:16]				
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LT[15:0]															
Value after reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.41 FLXAnFRSUCC2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	LTN[3:0]	Listen Timeout Noise Bit Configures (gListenNoise – 1)
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	LT[20:0]	Listen Timeout Bit Configures pdListenTimeout

(1) FLXAnFRSUCC2.LTN

Listen Timeout Noise Bit
The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
The range for gListenNoise is 2 to 16.
FLXAnFRSUCC2.LTN must be configured identical in all nodes of a cluster.
Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout.

CAUTION

The wakeup/startup noise timeout is calculated as follows:
pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT × (FLXAnFRSUCC2.LTN + 1)

(2) FLXAnFRSUCC2.LT

Listen Timeout Bit
The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
The range for pdListenTimeout is 1284 to 1283846 µT. Configures wakeup/startup listen time out in µT.

25.2.6.3 FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3

Access: FLXAnFRSUCC3 can be read or written in 32-bit units.
FLXAnFRSUCC3L can be read or written in 16-bit units.
FLXAnFRSUCC3LL can be read or written in 8-bit units.

Address: FLXAnFRSUCC3: <FLXAn_base> + 0088$_H$,
FLXAnFRSUCC3L: <FLXAn_base> + 0088$_H$,
FLXAnFRSUCC3LL: <FLXAn_base> + 0088$_H$

Value after reset: 0000 0011$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WCF[3:0]				WCP[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.42 FLXAnFRSUCC3 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	WCF[3:0]	Maximum Without Clock Correction Fatal Bit (transition to HALT state) Configures gMaxWithoutClockCorrectionFatal
3 to 0	WCP[3:0]	Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state) Configures gMaxWithoutClockCorrectionPassive

(1) FLXAnFRSUCC3.WCF

Maximum Without Clock Correction Fatal Bit (transition to HALT state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 1 to 15.
Must be identical in all nodes of a cluster.
Defines the number of consecutive even/odd cycle pairs with missing clock correction conditions that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set.

(2) FLXAnFRSUCC3.WCP

Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 1 to 15.
Must be identical in all nodes of a cluster.
Defines the number of consecutive even/odd cycle pairs with missing clock correction conditions that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state.

25.2.6.4 FLXAnFRNEMC — FlexRay NEM Configuration Register

Access:	FLXAnFRNEMC can be read or written in 32-bit units. FLXAnFRNEMCL can be read or written in 16-bit units. FLXAnFRNEMCLL can be read or written in 8-bit units.
Address:	FLXAnFRNEMC: <FLXAn_base> + 008C$_H$, FLXAnFRNEMCL: <FLXAn_base> + 008C$_H$, FLXAnFRNEMCLL: <FLXAn_base> + 008C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	NML[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.43 FLXAnFRNEMC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	NML[3:0]	Network Management Vector Length Bit Configures gNetworkManagementVectorLength

(1) FLXAnFRNEMC.NML

Network Management Vector Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 12 bytes.
The configured length must be identical in all nodes of a cluster. These bits configure the length of the NM vector in bytes.

25.2.6.5 FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1

Access:	FLXAnFRPRTC1 can be read or written in 32-bit units. FLXAnFRPRTC1L and FLXAnFRPRTC1H can be read or written in 16-bit units. FLXAnFRPRTC1LL, FLXAnFRPRTC1LH, FLXAnFRPRTC1HL, and FLXAnFRPRTC1HH can be read or written in 8-bit units.
Address:	FLXAnFRPRTC1: <FLXAn_base> + 0090_H, FLXAnFRPRTC1L: <FLXAn_base> + 0090_H, FLXAnFRPRTC1H: <FLXAn_base> + 0092_H, FLXAnFRPRTC1LL: <FLXAn_base> + 0090_H, FLXAnFRPRTC1LH: <FLXAn_base> + 0091_H, FLXAnFRPRTC1HL: <FLXAn_base> + 0092_H, FLXAnFRPRTC1HH: <FLXAn_base> + 0093_H
Value after reset:	084C 0633_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RWP[5:0]						—	RXW[8:0]								
Value after reset	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[1:0]		SPP[1:0]		—	CASM[6:0]							TSST[3:0]			
Value after reset	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.44 FLXAnFRPRTC1 Register Contents

Bit Position	Bit Name	Function
31 to 26	RWP[5:0]	Repetitions of Tx Wakeup Pattern Bit Configures pWakeupPattern
25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 16	RXW[8:0]	Wakeup Symbol Receive Window Length Bit Configures gdWakeupSymbolRxWindow
15, 14	BRP[1:0]	Baud Rate Prescaler Bit Configures gdSampleClockPeriod and pSamplesPerMicrotick 00_B = 10 Mbps 01_B = 5 Mbps 10_B = 2.5 Mbps 11_B = 2.5 Mbps
13, 12	SPP[1:0]	Strobe Point Position Bit Configures Strobe point position 00_B = Sample 5 01_B = Sample 4 10_B = Sample 6 11_B = Sample 5
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 4	CASM[6:0]	Collision Avoidance Symbol Max Bit Configures gdCASRxLowMax
3 to 0	TSST[3:0]	Transmission Start Sequence Transmitter Bit Configures gdTSSTransmitter

(1) FLXAnFRPRTC1.RWP

Repetitions of Tx Wakeup Pattern Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 63.
Configures the number of repetitions (sequences) of the Tx wakeup symbol.

(2) FLXAnFRPRTC1.RXW

Wakeup Symbol Receive Window Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 76 to 301.
Must be identical in all nodes of a cluster.
Configures the number of bit times used by the node to test the duration of the received wakeup pattern.

(3) FLXAnFRPRTC1.BRP

Baud Rate Prescaler Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock set to 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

- 00_B = 10 MBit/s
  gdSampleClockPeriod = 12.5 ns = 1 × “sample clock”
  pSamplesPerMicrotick = 2 (1 µT = 25 ns)
- 01_B = 5 MBit/s
  gdSampleClockPeriod = 25 ns = 2 × “sample clock”
  pSamplesPerMicrotick = 1 (1 µT = 25 ns)
- 10_B, 11_B = 2.5 MBit/s
  gdSampleClockPeriod = 50 ns = 4 × “sample clock”
  pSamplesPerMicrotick = 1 (1 µT = 50 ns)

(4) FLXAnFRPRTC1.SPP

Strobe Point Position Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by FLXAnFRPRTC1.SPP.

CAUTION

The current revision 2.1 of the FlexRay protocol requires that FLXAnFRPRTC1.SPP = “00_B”.
The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

(5) FLXAnFRPRTC1.CASM

Collision Avoidance Symbol Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
CASM[6] is fixed to 1. Valid values are 67 to 99.
Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS).

(6) FLXAnFRPRTC1.TSST

Transmission Start Sequence Transmitter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 3 to 15.
Must be identical in all nodes of a cluster.
Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 µT = 100ns @ 10 Mbps).

25.2.6.6 FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2

Access: FLXAnFRPRTC2 can be read or written in 32-bit units.
FLXAnFRPRTC2L and FLXAnFRPRTC2H can be read or written in 16-bit units.
FLXAnFRPRTC2LL, FLXAnFRPRTC2LH, FLXAnFRPRTC2HL, and FLXAnFRPRTC2HH can be read or written in 8-bit units.

Address: FLXAnFRPRTC2:<FLXAn_base> + 0094_H,
FLXAnFRPRTC2L: <FLXAn_base> + 0094_H, FLXAnFRPRTC2H: <FLXAn_base> + 0096_H,
FLXAnFRPRTC2LL: <FLXAn_base> + 0094_H, FLXAnFRPRTC2LH: <FLXAn_base> + 0095_H,
FLXAnFRPRTC2HL: <FLXAn_base> + 0096_H, FLXAnFRPRTC2HH: <FLXAn_base> + 0097_H

Value after reset: $0F2D\ 0A0E_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TXL[5:0]						TXI[7:0]							
Value after reset	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXL[5:0]						—	—	RXI[5:0]					
Value after reset	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.45 FLXAnFRPRTC2 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	TXL[5:0]	Wakeup Symbol Transmit Low Bit Configures gdWakeupSymbolTxLow
23 to 16	TXI[7:0]	Wakeup Symbol Transmit Idle Bit Configures gdWakeupSymbolTxIdle
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RXL[5:0]	Wakeup Symbol Receive Low Bit Configures gdWakeupSymbolRxLow
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RXI[5:0]	Wakeup Symbol Rx Idle Bit Configures gdWakeupSymbolRxIdle

(1) FLXAnFRPRTC2.TXL

Wakeup Symbol Transmit Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 15 to 60.
Must be identical in all nodes of a cluster.
Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol.

(2) FLXAnFRPRTC2.TXI

Wakeup Symbol Transmit Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 45 to 180.
Must be identical in all nodes of a cluster.
Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol.

(3) FLXAnFRPRTC2.RXL

Wakeup Symbol Receive Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 10 to 55.
Must be identical in all nodes of a cluster.
Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol.

(4) FLXAnFRPRTC2.RXI

Wakeup Symbol Rx Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 14 to 59.
Must be identical in all nodes of a cluster.
Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol.

25.2.6.7 FLXAnFRMHDC — FlexRay MHD Configuration Register

Access: FLXAnFRMHDC can be read or written in 32-bit units.
FLXAnFRMHDCL and FLXAnFRMHDCH can be read or written in 16-bit units.
FLXAnFRMHDCLL, FLXAnFRMHDCHL, and FLXAnFRMHDCHH can be read or written in 8-bit units.

Address: FLXAnFRMHDC: <FLXAn_base> + 0098$_H$,
FLXAnFRMHDCL: <FLXAn_base> + 0098$_H$, FLXAnFRMHDCH: <FLXAn_base> + 009A$_H$,
FLXAnFRMHDCLL: <FLXAn_base> + 0098$_H$, FLXAnFRMHDCHL: <FLXAn_base> + 009A$_H$,
FLXAnFRMHDCHH: <FLXAn_base> + 009B$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SLT[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SFDL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.46 FLXAnFRMHDC Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	SLT[12:0]	Start of Latest Transmit Bit Configures pLatestTx
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	SFDL[6:0]	Static Frame Data Length Bit Configures gPayloadLengthStatic

(1) FLXAnFRMHDC.SLT

Start of Latest Transmit Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 7981.
Configures the latest maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXAnFRMHDC.SLT is set to zero.

(2) FLXAnFRMHDC.SFDL

Static Frame Data Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 127.
The payload length must be identical in all nodes of a cluster.
Configures the cluster-wide payload length for all frames sent in the static segment in double bytes.

25.2.6.8 FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1

Access:	FLXAnFRGTUC1 can be read or written in 32-bit units. FLXAnFRGTUC1L and FLXAnFRGTUC1H can be read or written in 16-bit units. FLXAnFRGTUC1LL, FLXAnFRGTUC1LH, and FLXAnFRGTUC1HL can be read or written in 8-bit units.
Address:	FLXAnFRGTUC1: <FLXAn_base> + $00A0_H$, FLXAnFRGTUC1L: <FLXAn_base> + $00A0_H$, FLXAnFRGTUC1H: <FLXAn_base> + $00A2_H$, FLXAnFRGTUC1LL: <FLXAn_base> + $00A0_H$, FLXAnFRGTUC1LH: <FLXAn_base> + $00A1_H$, FLXAnFRGTUC1HL: <FLXAn_base> + $00A2_H$
Value after reset:	$0000\ 0280_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UT[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UT[15:0]															
Value after reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.47 FLXAnFRGTUC1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	UT[19:0]	Setting of Communication Cycle in Microticks Bit Configures pMicroPerCycle

(1) FLXAnFRGTUC1.UT

Setting of Communication Cycle in Microticks Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 640 to 640000 µT.
Configures the duration of the communication cycle in microticks.

25.2.6.9 FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2

Access: FLXAnFRGTUC2 can be read or written in 32-bit units.
FLXAnFRGTUC2L and FLXAnFRGTUC2H can be read or written in 16-bit units.
FLXAnFRGTUC2LL, FLXAnFRGTUC2LH, and FLXAnFRGTUC2HL can be read or written in 8-bit units.

Address: FLXAnFRGTUC2: <FLXAn_base> + 00A4$_H$,
FLXAnFRGTUC2L: <FLXAn_base> + 00A4$_H$, FLXAnFRGTUC2H: <FLXAn_base> + 00A6$_H$,
FLXAnFRGTUC2LL: <FLXAn_base> + 00A4$_H$, FLXAnFRGTUC2LH: <FLXAn_base> + 00A5$_H$,
FLXAnFRGTUC2HL: <FLXAn_base> + 00A6$_H$

Value after reset: 0002 000A$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SNM[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MPC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.48 FLXAnFRGTUC2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	SNM[3:0]	Sync Node Max Bit
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MPC[13:0]	Setting of Communication Cycle in Macrotick Bit Configures gMacroPerCycle

(1) FLXAnFRGTUC2.SNM

Sync Node Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 15.
Must be identical in all nodes of a cluster.
Indicates the maximum number of frames with sync frame indicator bit SYN set to 1.

(2) FLXAnFRGTUC2.MPC

Setting of Communication Cycle in Macrotick Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 10 to 16000 MT.
The cycle length must be identical in all nodes of a cluster. Configures the duration of one communication cycle in macroticks.

25.2.6.10 FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3

Access: FLXAnFRGTUC3 can be read or written in 32-bit units.
FLXAnFRGTUC3L and FLXAnFRGTUC3H can be read or written in 16-bit units.
FLXAnFRGTUC3LL, FLXAnFRGTUC3LH, FLXAnFRGTUC3HL, and FLXAnFRGTUC3HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC3: <FLXAn_base> + 00A8$_H$,
FLXAnFRGTUC3L: <FLXAn_base> + 00A8$_H$, FLXAnFRGTUC3H: <FLXAn_base> + 00AA$_H$,
FLXAnFRGTUC3LL: <FLXAn_base> + 00A8$_H$, FLXAnFRGTUC3LH: <FLXAn_base> + 00A9$_H$,
FLXAnFRGTUC3HL: <FLXAn_base> + 00AA$_H$, FLXAnFRGTUC3HH: <FLXAn_base> + 00AB$_H$

Value after reset: 0202 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MIOB[6:0]							—	MIOA[6:0]						
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UIOB[7:0]								UIOA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.49 FLXAnFRGTUC3 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MIOB[6:0]	Macrotick Initial Offset Channel B Bit Configures pMacroInitialOffset[B]
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MIOA[6:0]	Macrotick Initial Offset Channel A Bit Configures pMacroInitialOffset[A]
15 to 8	UIOB[7:0]	Microtick Initial Offset Channel B Bit Configures pMicroInitialOffset[B]
7 to 0	UIOA[7:0]	Microtick Initial Offset Channel A Bit Configures pMicroInitialOffset[A]

(1) FLXAnFRGTUC3.MIOB

Macrotick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 72 MT.
Must be identical in all nodes of a cluster.
Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(2) FLXAnFRGTUC3.MIOA

Macrotick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 72 MT.
Must be identical in all nodes of a cluster.
Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(3) FLXAnFRGTUC3.UIOB

Microtick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 240 µT.
Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [B] and therefore has to be set for each channel independently.

(4) FLXAnFRGTUC3.UIOA

Microtick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 240 µT.
Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [A] and therefore has to be set for each channel independently.

25.2.6.11 FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4

For details about configuration of FLXAnFRGTUC4.NIT and FLXAnFRGTUC4.OCS see **Section 25.3.2.5, Configuration of NIT Start and Offset Correction Start**.

Access:	FLXAnFRGTUC4 can be read or written in 32-bit units. FLXAnFRGTUC4L and FLXAnFRGTUC4H can be read or written in 16-bit units. FLXAnFRGTUC4LL, FLXAnFRGTUC4LH, FLXAnFRGTUC4HL, and FLXAnFRGTUC4HH can be read or written in 8-bit units.
Address:	FLXAnFRGTUC4: <FLXAn_base> + 00AC$_H$, FLXAnFRGTUC4L: <FLXAn_base> + 00AC$_H$, FLXAnFRGTUC4H: <FLXAn_base> + 00AE$_H$, FLXAnFRGTUC4LL: <FLXAn_base> + 00AC$_H$, FLXAnFRGTUC4LH: <FLXAn_base> + 00AD$_H$, FLXAnFRGTUC4HL: <FLXAn_base> + 00AE$_H$, FLXAnFRGTUC4HH: <FLXAn_base> + 00AF$_H$
Value after reset:	0008 0007$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OCS[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	NIT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.50 FLXAnFRGTUC4 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	OCS[13:0]	Offset Correction Start Bit Configures (gOffsetCorrectionStart – 1)
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	NIT[13:0]	Network Idle Time Start Bit Configures (gMacroPerCycle -gdNIT – 1)

(1) FLXAnFRGTUC4.OCS

Offset Correction Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 8 to 15998 MT.
For cluster consisting of E-Ray implementations only, it is sufficient to program FLXAnFRGTUC4.OCS = FLXAnFRGTUC4.NIT + 1.
Must be identical in all nodes of a cluster.
Determines the start position of the offset correction within the NIT phase, calculated from start of cycle.

(2) FLXAnFRGTUC4.NIT

Network Idle Time Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 7 to 15997 MT.
Must be identical in all nodes of a cluster.
Configures the starting point of the Network Idle Time NIT from the beginning of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if Macrotick = gMacroPerCycle –gdNIT –1 and the increment pulse of Macrotick is set.

25.2.6.12 FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5

Access: FLXAnFRGTUC5 can be read or written in 32-bit units.
FLXAnFRGTUC5L and FLXAnFRGTUC5H can be read or written in 16-bit units.
FLXAnFRGTUC5LL, FLXAnFRGTUC5LH, FLXAnFRGTUC5HL, and FLXAnFRGTUC5HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC5: <FLXAn_base> + 00B0$_H$,
FLXAnFRGTUC5L: <FLXAn_base> + 00B0$_H$, FLXAnFRGTUC5H: <FLXAn_base> + 00B2$_H$,
FLXAnFRGTUC5LL: <FLXAn_base> + 00B0$_H$, FLXAnFRGTUC5LH: <FLXAn_base> + 00B1$_H$,
FLXAnFRGTUC5HL: <FLXAn_base> + 00B2$_H$, FLXAnFRGTUC5HH: <FLXAn_base> + 00B3$_H$

Value after reset: 0E00 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEC[7:0]								—	—	—	CDD[4:0]				
Value after reset	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCB[7:0]								DCA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.51 FLXAnFRGTUC5 Register Contents

Bit Position	Bit Name	Function
31 to 24	DEC[7:0]	Decoding Correction Bit Configures pDecodingCorrection
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	CDD[4:0]	Cluster Drift Damping Bit Configures pClusterDriftDamping
15 to 8	DCB[7:0]	Delay Compensation Channel B Bit Configures pDelayCompensation[B]
7 to 0	DCA[7:0]	Delay Compensation Channel A Bit Configures pDelayCompensation[A]

(1) FLXAnFRGTUC5.DEC

Decoding Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 14 to 143 µT.
Configures the decoding correction value in microticks used to determine the primary time reference point.

(2) FLXAnFRGTUC5.CDD

Cluster Drift Damping Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 20 µT.
Configures the cluster drift damping value in microticks used in clock synchronization to minimize accumulation of rounding errors.

(3) FLXAnFRGTUC5.DCB

Delay Compensation Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 200 µT.
Used to compensate for reception delays on channel B. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

(4) FLXAnFRGTUC5.DCA

Delay Compensation Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 200 µT.
Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

25.2.6.13 FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6

Access: FLXAnFRGTUC6 can be read or written in 32-bit units.
FLXAnFRGTUC6L and FLXAnFRGTUC6H can be read or written in 16-bit units.
FLXAnFRGTUC6LL, FLXAnFRGTUC6LH, FLXAnFRGTUC6HL, and FLXAnFRGTUC6HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC6: <FLXAn_base> + 00B4$_H$,
FLXAnFRGTUC6L: <FLXAn_base> + 00B4$_H$, FLXAnFRGTUC6H: <FLXAn_base> + 00B6$_H$,
FLXAnFRGTUC6LL: <FLXAn_base> + 00B4$_H$, FLXAnFRGTUC6LH: <FLXAn_base> + 00B5$_H$,
FLXAnFRGTUC6HL: <FLXAn_base> + 00B6$_H$, FLXAnFRGTUC6HH: <FLXAn_base> + 00B7$_H$

Value after reset: 0002 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MOD[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASR[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.52 FLXAnFRGTUC6 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MOD[10:0]	Maximum Oscillator Drift Bit Configures pdMaxDrift
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	ASR[10:0]	Accepted Startup Range Bit Configures pdAcceptedStartupRange

(1) FLXAnFRGTUC6.MOD

Maximum Oscillator Drift Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 1923 µT.
Configures the maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT.

(2) FLXAnFRGTUC6.ASR

Accepted Startup Range Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 1875 µT.
Configures the number of microticks constituting the expanded range of measured deviation for startup frames during integration.

25.2.6.14 FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7

Access: FLXAnFRGTUC7 can be read or written in 32-bit units.
FLXAnFRGTUC7L and FLXAnFRGTUC7H can be read or written in 16-bit units.
FLXAnFRGTUC7LL, FLXAnFRGTUC7LH, FLXAnFRGTUC7HL, and FLXAnFRGTUC7HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC7: <FLXAn_base> + 00B8$_{H}$,
FLXAnFRGTUC7L: <FLXAn_base> + 00B8$_{H}$, FLXAnFRGTUC7H: <FLXAn_base> + 00BA$_{H}$,
FLXAnFRGTUC7LL: <FLXAn_base> + 00B8$_{H}$, FLXAnFRGTUC7LH: <FLXAn_base> + 00B9$_{H}$,
FLXAnFRGTUC7HL: <FLXAn_base> + 00BA$_{H}$, FLXAnFRGTUC7HH: <FLXAn_base> + 00BB$_{H}$

Value after reset: 0002 0004$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NSS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SSL[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.53 FLXAnFRGTUC7 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	NSS[9:0]	Number of Static Slots Bit Configures gNumberOfStaticSlots
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	SSL[9:0]	Static Slot Length Bit Configures gdStaticSlot

(1) FLXAnFRGTUC7.NSS

Number of Static Slots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 1023.
The number of static slots must be identical in all nodes of a cluster. Configures the number of static slots in a cycle.

(2) FLXAnFRGTUC7.SSL

Static Slot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 4 to 659 MT.
The static slot length must be identical in all nodes of a cluster. Configures the length of a static slot in macroticks.

25.2.6.15 FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8

Access: FLXAnFRGTUC8 can be read or written in 32-bit units.
FLXAnFRGTUC8L and FLXAnFRGTUC8H can be read or written in 16-bit units.
FLXAnFRGTUC8LL, FLXAnFRGTUC8HL, and FLXAnFRGTUC8HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC8: <FLXAn_base> + 00BC$_H$,
FLXAnFRGTUC8L: <FLXAn_base> + 00BC$_H$, FLXAnFRGTUC8H: <FLXAn_base> + 00BE$_H$,
FLXAnFRGTUC8LL: <FLXAn_base> + 00BC$_H$, FLXAnFRGTUC8HL: <FLXAn_base> + 00BE$_H$,
FLXAnFRGTUC8HH: <FLXAn_base> + 00BF$_H$

Value after reset: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NMS[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.54 FLXAnFRGTUC8 Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	NMS[12:0]	Number of Minislots Bit Configures gNumberOfMinislots
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	MSL[5:0]	Minislot Length Bit Configures gdMinislot

(1) FLXAnFRGTUC8.NMS

Number of Minislots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 7986.
The number of minislots must be identical in all nodes of a cluster. Configures the number of minislots within the dynamic segment of a cycle.

(2) FLXAnFRGTUC8.MSL

Minislot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 63 MT.
The minislot length must be identical in all nodes of a cluster. Configures the length of a minislot in macroticks.

25.2.6.16 FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9

Access:	FLXAnFRGTUC9 can be read or written in 32-bit units. FLXAnFRGTUC9L and FLXAnFRGTUC9H can be read or written in 16-bit units. FLXAnFRGTUC9LL, FLXAnFRGTUC9LH, and FLXAnFRGTUC9HL can be read or written in 8-bit units.
Address:	FLXAnFRGTUC9: <FLXAn_base> + 00C0$_H$, FLXAnFRGTUC9L: <FLXAn_base> + 00C0$_H$, FLXAnFRGTUC9H: <FLXAn_base> + 00C2$_H$, FLXAnFRGTUC9LL: <FLXAn_base> + 00C0$_H$, FLXAnFRGTUC9LH: <FLXAn_base> + 00C1$_H$, FLXAnFRGTUC9HL: <FLXAn_base> + 00C2$_H$
Value after reset:	0000 0101$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSI[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MAPO[4:0]					—	—	APO[5:0]					
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.55 FLXAnFRGTUC9 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DSI[1:0]	Dynamic Slot Idle Phase Bit Configures gdDynamicSlotIdlePhase
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	MAPO[4:0]	Minislot Action Point Offset Bit Configures gdMinislotActionPointOffset
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	APO[5:0]	Action Point Offset Bit Configures gdActionPointOffset

(1) FLXAnFRGTUC9.DSI

Dynamic Slot Idle Phase Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 2.
Must be identical in all nodes of a cluster.
Configures the duration of the dynamic slot idle phase in the number of minislots. The duration has to be greater or equal than the idle detection time.

(2) FLXAnFRGTUC9.MAPO

Minislot Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 1 to 31 MT.
Must be identical in all nodes of a cluster.
Configures the action point offset in macroticks within the minislots of the dynamic segment.

(3) FLXAnFRGTUC9.APO

Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 1 to 63 MT.
Must be identical in all nodes of a cluster.
Configures the action point offset in macroticks within static slots and symbol window.

25.2.6.17 FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10

Access: FLXAnFRGTUC10 can be read or written in 32-bit units.
FLXAnFRGTUC10L and FLXAnFRGTUC10H can be read or written in 16-bit units.
FLXAnFRGTUC10LL, FLXAnFRGTUC10LH, FLXAnFRGTUC10HL, and FLXAnFRGTUC10HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC10: <FLXAn_base> + 00C4$_{\text{H}}$,
FLXAnFRGTUC10L: <FLXAn_base> + 00C4$_{\text{H}}$, FLXAnFRGTUC10$_{\text{H}}$: <FLXAn_base> + 00C6$_{\text{H}}$,
FLXAnFRGTUC10LL: <FLXAn_base> + 00C4$_{\text{H}}$, FLXAnFRGTUC10L$_{\text{H}}$: <FLXAn_base> + 00C5$_{\text{H}}$,
FLXAnFRGTUC10HL: <FLXAn_base> + 00C6$_{\text{H}}$, FLXAnFRGTUC10H$_{\text{H}}$: <FLXAn_base> + 00C7$_{\text{H}}$

Value after reset: 0002 0005$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MOC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.56 FLXAnFRGTUC10 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MRC[10:0]	Maximum Rate Correction Bit Configures pRateCorrectionOut
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MOC[13:0]	Maximum Offset Correction Bit Configures pOffsetCorrectionOut

(1) FLXAnFRGTUC10.MRC

Maximum Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 2 to 1923 µT.
Configures the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks the internal rate correction value against the maximum rate correction value (absolute value).

(2) FLXAnFRGTUC10.MOC

Maximum Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 5 to 15266 µT.
Configures the maximum permitted offset correction value (absolute value) to be applied by the internal clock synchronization algorithm (absolute value). The CC checks the internal offset correction value against the maximum offset correction value.

25.2.6.18 FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11

Access: FLXAnFRGTUC11 can be read or written in 32-bit units.
FLXAnFRGTUC11L and FLXAnFRGTUC11H can be read or written in 16-bit units.
FLXAnFRGTUC11LL, FLXAnFRGTUC11LH, FLXAnFRGTUC11HL, and FLXAnFRGTUC11HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC11: <FLXAn_base> + 00C8$_H$,
FLXAnFRGTUC11L: <FLXAn_base> + 00C8$_H$, FLXAnFRGTUC11H: <FLXAn_base> + 00CA$_H$,
FLXAnFRGTUC11LL: <FLXAn_base> + 00C8$_H$, FLXAnFRGTUC11LH: <FLXAn_base> + 00C9$_H$,
FLXAnFRGTUC11HL: <FLXAn_base> + 00CA$_H$, FLXAnFRGTUC11HH: <FLXAn_base> + 00CB$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ERC[2:0]			—	—	—	—	—	EOC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ERCC[1:0]		—	—	—	—	—	—	EOCC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.57 FLXAnFRGTUC11 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 24	ERC[2:0]	External Rate Correction Bit Configures pExternRateCorrection
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	EOC[2:0]	External Offset Correction Bit Configures pExternOffsetCorrection
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	ERCC[1:0]	External Rate Correction Control Bit Configures vExternRateControl 00$_B$: External rate correction is prohibited. 01$_B$: External rate correction is prohibited. 10$_B$: Subtract 11$_B$: Add
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	EOCC[1:0]	External Offset Correction Control Bit Configures vExternOffsetControl 00$_B$: External offset correction is prohibited. 01$_B$: External offset correction is prohibited. 10$_B$: Subtract 11$_B$: Add

(1) FLXAnFRGTUC11.ERC

External Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 7 µT.
Configures the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated rate correction value. The value is applied during NIT.

(2) FLXAnFRGTUC11.EOC

External Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Valid values are 0 to 7 µT.
Configures the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated offset correction value. The value is applied during NIT.

(3) FLXAnFRGTUC11.ERCC

External Rate Correction Control Bit

Should be modified only outside NIT (Network Idle Time).
By writing to FLXAnFRGTUC11.ERCC the external rate correction is enabled as specified below.

- 00_B = External rate correction is prohibited.
- 01_B = External rate correction is prohibited.
- 10_B = Subtract
  External rate correction value subtracted from calculated rate correction value
- 11_B = Add
  External rate correction value added to calculated rate correction value

(4) FLXAnFRGTUC11.EOCC

External Offset Correction Control Bit

Should be modified only outside NIT (Network Idle Time).
By writing to FLXAnFRGTUC11.EOCC the external offset correction is enabled as specified below.

- 00_B = External offset correction is prohibited.
- 01_B = External offset correction is prohibited.
- 10_B = Subtract
  External offset correction value subtracted from calculated offset correction value
- 11_B = Add
  External offset correction value added to calculated offset correction value

25.2.7 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses).

25.2.7.1 FLXAnFRCCSV — FlexRay CC Status Vector Register

Access: FLXAnFRCCSV is a read-only register that can be read in 32-bit units.
FLXAnFRCCSVL and FLXAnFRCCSVH are the read-only registers that can be read in 16-bit units.
FLXAnFRCCSVLL, FLXAnFRCCSVLH, FLXAnFRCCSVHL, and FLXAnFRCCSVHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRCCSV: <FLXAn_base> + 0100$_H$,
FLXAnFRCCSVL: <FLXAn_base> + 0100$_H$, FLXAnFRCCSVH: <FLXAn_base> + 0102$_H$,
FLXAnFRCCSVLL: <FLXAn_base> + 0100$_H$, FLXAnFRCCSVLH: <FLXAn_base> + 0101$_H$,
FLXAnFRCCSVHL: <FLXAn_base> + 0102$_H$, FLXAnFRCCSVHH: <FLXAn_base> + 0103$_H$

Value after reset: 0010 4000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PSL[5:0]						RCA[4:0]					WSV[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CSI	CSAI	CSNI	—	—	SLM[1:0]		HRQ	FSI	POCS[5:0]					
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.58 FLXAnFRCCSV Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 24	PSL[5:0]	POC Status Log Flag Status of FLXAnFRCCSV.POCS immediately before entering HALT state.
23 to 19	RCA[4:0]	Remaining Coldstart Attempts Flag Indicates vRemainingColdstartAttempts
18 to 16	WSV[2:0]	Wakeup Status Flag Indicates vPOC!WakeupStatus 000$_B$: UNDEFINED 001$_B$: RECEIVED_HEADER 010$_B$: RECEIVED_WUP 011$_B$: COLLISION_HEADER 100$_B$: COLLISION_WUP 101$_B$: COLLISION_UNKNOWN 110$_B$: TRANSMITTED 111$_B$: Reserved
15	Reserved	When read, the value after reset is returned.
14	CSI	Cold Start Inhibit Flag Indicates vColdStartInhibit 0: Cold starting of node enabled 1: Cold starting of node disabled
13	CSAI	Coldstart Abort Indicator Flag

Table 25.58 FLXAnFRCCSV Register Contents

Bit Position	Bit Name	Function
12	CSNI	Coldstart Noise Indicator Flag Indicates vPOC!ColdstartNoise
11, 10	Reserved	When read, the value after reset is returned.
9, 8	SLM[1:0]	Slot Mode Flag Indicates vPOC!SlotMode 00_B: SINGLE 01_B: reserved 10_B: ALL_PENDING 11_B: ALL
7	HRQ	Halt Request Flag Indicates vPOC!CHIHaltRequest
6	FSI	Freeze Status Indicator Flag Indicates vPOC!Freeze
5 to 0	POCS[5:0]	Protocol Operation Control Status Flag

(1) FLXAnFRCCSV.PSL

POC Status Log Flag

Set the value of FLXAnFRCCSV.POCS immediately before entering HALT state.
Set to HALT when FREEZE command is applied during HALT state and FLXAnFRCCSV.FSI is not already set i.e. the HALT state was not reached by FREEZE command.
Reset to “000000_B” when leaving HALT state.

(2) FLXAnFRCCSV.RCA

Remaining Coldstart Attempts Flag

Indicates the number of remaining coldstart attempts (vRemainingColdstartAttempts).
The value after a reset of FLXAnFRCCSV.RCA during CONFIG and DEFAULT_CONFIG state is also FLXAnFRSUCC1.CSA.
The RUN command resets this counter to the maximum number of coldstart attempts as configured by FLXAnFRSUCC1.CSA.

(3) FLXAnFRCCSV.WSV

Wakeup Status Flag

Indicates the status of the current wakeup attempt (vPOC!WakeupStatus).
Reset to 0 when entering Wakeup state, by CHI command RESET_STATUS_INDICATORS, or by transition from DEFAULT_CONFIG to CONFIG state.

- 000_B = UNDEFINED
  Wakeup not yet executed by the CC.
- 001_B = RECEIVED_HEADER
  Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.
- 010_B = RECEIVED_WUP
  Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

- 011_B = COLLISION_HEADER
  Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.
- 100_B = COLLISION_WUP
  Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.
- 101_B = COLLISION_UNKNOWN
  Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.
- 110_B = TRANSMITTED
  Set when the CC has successfully completed the transmission of the wakeup pattern.
- 111_B = reserved

(4) FLXAnFRCCSV.CSI

Cold Start Inhibit Flag

Indicates whether the node is disabled from cold starting (vColdStartInhibit).
The flag is set to 1 whenever the POC enters READY state due to CHI command READY.
The flag has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = “1001_B”).

(5) FLXAnFRCCSV.CSAI

Coldstart Abort Indicator Flag

Indicates that coldstart was aborted.
Cleared by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(6) FLXAnFRCCSV.CSNI

Coldstart Noise Indicator Flag

Indicates that the cold start procedure occurred under noisy conditions (vPOC!ColdstartNoise).
Cleared by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(7) FLXAnFRCCSV.SLM

Slot Mode Flag

Indicates the actual slot mode of the POC (vPOC!SlotMode) in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE.
Default value is SINGLE. Changes to ALL, depending on FLXAnFRSUCC1.TSM.
In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL.
Set FLXAnFRSUCC1.TSM to SINGLE except for NORMAL_ACTIVE or NORMAL_PASSIVE.

(8) FLXAnFRCCSV.HRQ

Halt Request Flag

Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle (vPOC!CHIHaltRequest).
Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(9) FLXAnFRCCSV.FSI

Freeze Status Indicator Flag

Indicates that the POC has entered the HALT state due to CHI command FREEZE (CMD = 0111_B) or due to an error condition requiring an immediate POC halt (vPOC!Freeze).
Reset by transition from HALT to DEFAULT_CONFIG state.

(10) FLXAnFRCCSV.POCS

Protocol Operation Control Status Flag

Indicates the current state of operation of the CC Protocol Operation Control
00 0000_B = DEFAULT_CONFIG state
00 0001_B = READY state
00 0010_B = NORMAL_ACTIVE state
00 0011_B = NORMAL_PASSIVE state
00 0100_B = HALT state
00 1111_B = CONFIG state

Indicates the current state of operation of the POC in the wakeup path
01 0000_B = WAKEUP_STANDBY state
01 0001_B = WAKEUP_LISTEN state
01 0010_B = WAKEUP_SEND state
01 0011_B = WAKEUP_DETECT state

Indicates the current state of operation of the POC in the startup path
10 0000_B = STARTUP_PREPARE state
10 0001_B = COLDSTART_LISTEN state
10 0010_B = COLDSTART_COLLISION_RESOLUTION state
10 0011_B = COLDSTART_CONSISTENCY_CHECK state
10 0100_B = COLDSTART_GAP state
10 0101_B = COLDSTART_JOIN State
10 0110_B = INTEGRATION_COLDSTART_CHECK state
10 0111_B = INTEGRATION_LISTEN state
10 1000_B = INTEGRATION_CONSISTENCY_CHECK state
10 1001_B = INITIALIZE_SCHEDULE state
10 1010_B = ABORT_STARTUP state
10 1011_B = STARTUP_SUCCESS state

Others = reserved

25.2.7.2 FLXAnFRCCEV — FlexRay CC Error Vector Register

Access:	FLXAnFRCCEV is a read-only register that can be read in 32-bit units. FLXAnFRCCEVL is a read-only register that can be read in 16-bit units. FLXAnFRCCEVLL and FLXAnFRCCEVLH are the read-only registers that can be read in 8-bit units.
Address:	FLXAnFRCCEV: <FLXAn_base> + 0104_H, FLXAnFRCCEVL: <FLXAn_base> + 0104_H, FLXAnFRCCEVLL: <FLXAn_base> + 0104_H, FLXAnFRCCEVLH: <FLXAn_base> + 0105_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PTAC[4:0]					ERRM[1:0]		—	—	CCFC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.59 FLXAnFRCCEV Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12 to 8	PTAC[4:0]	Passive to Active Count Flag Indicates vAllowPassiveToActive
7, 6	ERRM[1:0]	Error Mode Flag Indicates vPOC!ErrorMode 00_B: ACTIVE 01_B: PASSIVE 10_B: COMM_HALT 11_B: reserved
5, 4	Reserved	When read, the value after reset is returned.
3 to 0	CCFC[3:0]	Clock Correction Failed Counter Indicates vClockCorrectionFailed

(1) FLXAnFRCCEV.PTAC

Passive to Active Count Flag

Indicates the number of consecutive even/odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA -1.
Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(2) FLXAnFRCCEV.ERRM

Error Mode Flag

Indicates the current error mode of the POC (vPOC!ErrorMode).
Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(3) FLXAnFRCCEV.CCFC

Clock Correction Failed Counter

Indicates the clock correction failed counter value of the POC (vClockCorrectionFailed).
The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active.
The Clock Correction Failed Counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active.
The Clock Correction Failed Counter stops at 15.
Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

25.2.7.3 FLXAnFRSCV — FlexRay Slot Counter Value Register

Access: FLXAnFRSCV is a read-only register that can be read in 32-bit units.
FLXAnFRSCVL and FLXAnFRSCVH are the read-only registers that can be read in 16-bit units.
FLXAnFRSCVLL, FLXAnFRSCVLH, FLXAnFRSCVHL, and FLXAnFRSCVHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSCV: <FLXAn_base> + 0110$_H$,
FLXAnFRSCVL: <FLXAn_base> + 0110$_H$, FLXAnFRSCVH: <FLXAn_base> + 0112$_H$,
FLXAnFRSCVLL: <FLXAn_base> + 0110$_H$, FLXAnFRSCVLH: <FLXAn_base> + 0111$_H$,
FLXAnFRSCVHL: <FLXAn_base> + 0112$_H$, FLXAnFRSCVHH: <FLXAn_base> + 0113$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SCCB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCCA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.60 FLXAnFRSCV Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SCCB[10:0]	Slot Counter Channel B Indicates vSlotCounter[B]
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SCCA[10:0]	Slot Counter Channel A Indicates vSlotCounter[A]

(1) FLXAnFRSCV.SCCB

Slot Counter Channel B

Indicates the current slot counter value on channel B (vSlotCounter[B]). The value is incremented by the CC and initialized at the start of a communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSCV.SCCA

Slot Counter Channel A

Indicates the current slot counter value on channel A (vSlotCounter[A]). The value is incremented by the CC and initialized at the start of a communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.4 FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register

Access: FLXAnFRMTCCV is a read-only register that can be read in 32-bit units.
FLXAnFRMTCCVL and FLXAnFRMTCCVH are the read-only registers that can be read in 16-bit units.
FLXAnFRMTCCVLL, FLXAnFRMTCCVLH, and FLXAnFRMTCCVHL are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMTCCV: <FLXAn_base> + 0114$_H$,
FLXAnFRMTCCVL: <FLXAn_base> + 0114$_H$, FLXAnFRMTCCVH: <FLXAn_base> + 0116$_H$,
FLXAnFRMTCCVLL: <FLXAn_base> + 0114$_H$,FLXAnFRMTCCVLH: <FLXAn_base> + 0115$_H$,
FLXAnFRMTCCVHL: <FLXAn_base> + 0116$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	CCV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.61 FLXAnFRMTCCV Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	CCV[5:0]	Cycle Counter Value Indicates vCycleCounter
15, 14	Reserved	When read, the value after reset is returned.
13 to 0	MTV[13:0]	Macrotick Value Indicates vMacrotick

(1) FLXAnFRMTCCV.CCV

Cycle Counter Value

Indicates the current cycle counter value (vCycleCounter). The value is incremented by the CC at the start of a communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRMTCCV.MTV

Macrotick Value

Indicates the current macrotick value (vMacrotick). The value is incremented by the CC and reset at the start of a communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.5 FLXAnFRRCV — FlexRay Rate Correction Value Register

Access: FLXAnFRRCV is a read-only register that can be read in 32-bit units.
FLXAnFRRCVL is a read-only register that can be read in 16-bit units.
FLXAnFRRCVLL and FLXAnFRRCVLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRCV: <FLXAn_base> + 0118_H,
FLXAnFRRCVL: <FLXAn_base> + 0118_H,
FLXAnFRRCVLL: <FLXAn_base> + 0118_H, FLXAnFRRCVLH: <FLXAn_base> + 0119_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RCV[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.62 FLXAnFRRCV Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11 to 0	RCV[11:0]	Rate Correction Value Flag Indicates vRateCorrection

(1) FLXAnFRRCV.RCV

Rate Correction Value Flag

Indicates internal rate correction value (vRateCorrection/ two’s complement) before limitation. If the FLXAnFRRCV.RCV value exceeds the limits defined by FLXAnFRGTUC10.MRC, flag FLXAnFRSFS.RCLR is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

CAUTION

The amount by which this value exceeded the limits is added to the external rate correction value.

25.2.7.6 FLXAnFROCV — FlexRay Offset Correction Value Register

Access: FLXAnFROCV is a read-only register that can be read in 32-bit units.
FLXAnFROCVL and FLXAnFROCVH are the read-only registers that can be read in 16-bit units.
FLXAnFROCVLL, FLXAnFROCVLH, and FLXAnFROCVHL are the read-only registers that can be read in 8- bit units.

Address: FLXAnFROCV: <FLXAn_base> + $011C_H$,
FLXAnFROCVL: <FLXAn_base> + $011C_H$, FLXAnFROCVH: <FLXAn_base> + $011E_H$,
FLXAnFROCVLL: <FLXAn_base> + $011C_H$, FLXAnFROCVLH: <FLXAn_base> + $011D_H$,
FLXAnFROCVHL: <FLXAn_base> + $011E_H$,

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OCV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.63 FLXAnFROCV Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 0	OCV[18:0]	Offset Correction Value Flag Indicates vOffsetCorrection

(1) FLXAnFROCV.OCV

Offset Correction Value Flag

Indicates the internal offset correction value (vOffsetCorrection/ two’s complement) before limitation. If the FLXAnFROCV.OCV value exceeds the limits defined by FLXAnFRGTUC10.MOC, flag FLXAnFRSFS.OCLR is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

CAUTION

The amount by which this value exceeded the limits is added to the external offset correction value.

25.2.7.7 FLXAnFRSFS — FlexRay Sync Frame Status Register

Access: FLXAnFRSFS is a read-only register that can be read in 32-bit units.
FLXAnFRSFSL and FLXAnFRSFSH are the read-only registers that can be read in 16-bit units.
FLXAnFRSFSLL, FLXAnFRSFSLH, and FLXAnFRSFSHL are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSFS: <FLXAn_base> + 0120$_H$
FLXAnFRSFSL: <FLXAn_base> + 0120$_H$, FLXAnFRSFSH: <FLXAn_base> + 0122$_H$
FLXAnFRSFSLL: <FLXAn_base> + 0120$_H$, FLXAnFRSFSLH: <FLXAn_base> + 0121$_H$,
FLXAnFRSFSHL: <FLXAn_base> + 0122$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RCLR	MRCS	OCLR	MOCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSBO[3:0]				VSBE[3:0]				VSAO[3:0]				VSAE[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.64 FLXAnFRSFS Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19	RCLR	Rate Correction Limit Reached Flag 0: Rate correction below limit 1: Rate correction limit reached
18	MRCS	Missing Rate Correction Signal Flag 0: Rate correction signal valid 1: Missing rate correction signal
17	OCLR	Offset Correction Limit Reached Flag 0: Offset correction below limit 1: Offset correction limit reached
16	MOCS	Missing Offset Correction Signal Flag 0: Offset correction signal valid 1: Missing offset correction signal
15 to 12	VSBO[3:0]	Valid Sync Frames Channel B, odd communication cycle
11 to 8	VSBE[3:0]	Valid Sync Frames Channel B, even communication cycle
7 to 4	VSAO[3:0]	Valid Sync Frames Channel A, odd communication cycle
3 to 0	VSAE[3:0]	Valid Sync Frames Channel A, even communication cycle

(1) FLXAnFRSFS.RCLR

Rate Correction Limit Reached Flag

The Rate Correction Limit Reached flag indicates that the rate correction value has exceeded its limit as defined by FLXAnFRGTUC10.MRC10 - MRC0. The flag is updated by the CC at start of offset correction phase.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(2) FLXAnFRSFS.MRCS

Missing Rate Correction Signal Flag

The Missing Rate Correction flag indicates that no rate correction calculation can be performed because no pairs of even/odd sync frames were received. The flag is updated by the CC at start of offset correction phase.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(3) FLXAnFRSFS.OCLR

Offset Correction Limit Reached Flag

The Offset Correction Limit Reached flag indicates that the offset correction value has exceeded its limit as defined by FLXAnFRGTUC10.MOC. The flag is updated by the CC at start of offset correction phase.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(4) FLXAnFRSFS.MOCS

Missing Offset Correction Signal Flag

The Missing Offset Correction flag indicates that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(5) FLXAnFRSFS.VSBO

Valid Sync Frames Channel B, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is 1.
Indicates the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(6) FLXAnFRSFS.VSBE

Valid Sync Frames Channel B, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is 1.
Indicates the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(7) FLXAnFRSFS.VSAO

Valid Sync Frames Channel A, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is 1.
Indicates the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(8) FLXAnFRSFS.VSAE

Valid Sync Frames Channel A, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is 1.
Indicates the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

25.2.7.8 FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register

Indicates the symbol windows related status. Symbol window related status information is updated by the CC at the end of the symbol window for each channel. NIT related status information is updated by the CC at the end of the NIT for each channel.

During startup the status data is not updated.

Access: FLXAnFRSWNIT is a read-only register that can be read in 32-bit units.
FLXAnFRSWNITL is a read-only register that can be read in 16-bit units.
FLXAnFRSWNITLL and FLXAnFRSWNITLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSWNIT: <FLXAn_base> + 0124_{H},
FLXAnFRSWNITL: <FLXAn_base> + 0124_{H},
FLXAnFRSWNITLL: <FLXAn_base> + 0124_{H}, FLXAnFRSWNITLH: <FLXAn_base> + 0125_{H}

Value after reset: 0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.65 FLXAnFRSWNIT Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	SBNB	Slot Boundary Violation during NIT Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B
10	SENB	Syntax Error during NIT Channel B Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel B
9	SBNA	Slot Boundary Violation during NIT Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A
8	SENA	Syntax Error during NIT Channel A Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel A
7	MTSB	MTS Received on Channel B Flag 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
6	MTSA	MTS Received on Channel A Flag 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
5	TCSB	Transmission Conflict in Symbol Window Channel B Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B

Table 25.65 FLXAnFRSWNIT Register Contents

Bit Position	Bit Name	Function
4	SBSB	Slot Boundary Violation in Symbol Window Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B
3	SESB	Syntax Error in Symbol Window Channel B Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel B
2	TCSA	Transmission Conflict in Symbol Window Channel A Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A
1	SBSA	Slot Boundary Violation in Symbol Window Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A
0	SESA	Syntax Error in Symbol Window Channel A Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel A

(1) FLXAnFRSWNIT.SBNB

Indicates a Slot Boundary Violation during NIT Channel B Flag (vSS!BViolationB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSWNIT.SENB

Indicates a Syntax Error during NIT Channel B Flag (vSS!SyntaxErrorB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRSWNIT.SBNA

Indicates a Slot Boundary Violation during NIT Channel A Flag (vSS!BViolationA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRSWNIT.SENA

Indicates a Syntax Error during NIT Channel A Flag (vSS!SyntaxErrorA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRSWNIT.MTSB

Indicates a MTS Received on Channel B Flag (vSS!ValidMTSB).

Media Access Test Symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.
When this bit is set to 1, also interrupt flag FLXAnFRSIR.MTSB is set to 1. Cleared when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRSWNIT.MTSA

Indicates a MTS Received on Channel A Flag (vSS!ValidMTSA).

Media Access Test Symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.
When this bit is set to 1, also interrupt flag FLXAnFRSIR.MTSA is set to 1. Cleared when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRSWNIT.TCSB

Indicates a Transmission Conflict in Symbol Window Channel B Flag (vSS!TxConflictB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRSWNIT.SBSB

Indicates a Slot Boundary Violation in Symbol Window Channel B Flag (vSS!BViolationB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRSWNIT.SESB

Indicates a Syntax Error in Symbol Window Channel B Flag (vSS!SyntaxErrorB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRSWNIT.TCSA

Indicates a Transmission Conflict in Symbol Window Channel A Flag (vSS!TxConflictA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(11) FLXAnFRSWNIT.SBSA

Indicates a Slot Boundary Violation in Symbol Window Channel A Flag (vSS!BViolationA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(12) FLXAnFRSWNIT.SESA

Indicates a Syntax Error in Symbol Window Channel A Flag (vSS!SyntaxErrorA).

Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.9 FLXAnFRACS — FlexRay Aggregated Channel Status Register

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception.

The aggregated channel status also includes status data from the symbol window and the network idle time.

The status data is updated (set) after each slot and aggregated until it is cleared by the Host. During startup the status data is not updated.

Access: FLXAnFRACS can be read or written in 32-bit units.
FLXAnFRACSL can be read or written in 16-bit units.
FLXAnFRACSLL and FLXAnFRACSLH can be read or written in 8-bit units.

Address: FLXAnFRACS: <FLXAn_base> + 0128_{H},
FLXAnFRACSL: <FLXAn_base> + 0128_{H},
FLXAnFRACSLL: <FLXAn_base> + 0128_{H}, FLXAnFRACSLH: <FLXAn_base> + 0129_{H}

Value after reset: 0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SBVB	CIB	CEDB	SEDB	VFRB	—	—	—	SBVA	CIA	CEDA	SEDA	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 25.66 FLXAnFRACS Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SBVB	Slot Boundary Violation on Channel B Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B
11	CIB	Communication Indicator Channel B Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication
10	CEDB	Content Error Detected on Channel B Flag 0: No frame with content error received 1: Frame(s) with content error received on channel B
9	SEDB	Syntax Error Detected on Channel B Flag 0: No syntax error observed 1: Syntax error(s) observed on channel B
8	VFRB	Valid Frame Received on Channel B Flag 0: No valid frame received 1: Valid frame(s) received on channel B
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SBVA	Slot Boundary Violation on Channel A Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A

Table 25.66 FLXAnFRACS Register Contents

Bit Position	Bit Name	Function
3	CIA	Communication Indicator Channel A Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication
2	CEDA	Content Error Detected on Channel A Flag 0: No frame with content error received 1: Frame(s) with content error received on channel A
1	SEDA	Syntax Error Detected on Channel A Flag 0: No syntax error observed 1: Syntax error(s) observed on channel A
0	VFRA	Valid Frame Received on Channel A Flag 0: No valid frame received 1: Valid frame(s) received on channel A

(1) FLXAnFRACS.SBVB

Slot Boundary Violation on Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRACS.CIB

Communication Indicator Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Indicates that one or more valid frames were received on channel B in slots that also contained additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

CAUTION

The set condition of the flag FLXAnFRACS.CIB is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRACS.CEDB

Content Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Indicates that one or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRACS.SEDB

Syntax Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRACS.VFRB

Valid Frame Received on Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more valid frames were received on channel B in any static or dynamic slot during the observation period.
Cleared when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRACS.SBVA

Slot Boundary Violation on Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRACS.CIA

Communication Indicator Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

CAUTION

The set condition of the flag FLXAnFRACS.CIA is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRACS.CEDA

Content Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRACS.SEDA

Syntax Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRACS.VFRA

Valid Frame Received on Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
One or more valid frames were received on channel A in any static or dynamic slot during the observation period.
Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.10 FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)

Registers FLXAnFRESID1 to FLXAnFRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXAnFRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFRESID1.RXEA, FLXAnFRESID1.RXEB are set. The value is updated during the NIT of each even communication cycle.

Access: FLXAnFRESIDm is a read-only register that can be read in 32-bit units.
FLXAnFRESIDmL is a read-only register that can be read in 16-bit units.
FLXAnFRESIDmLL and FLXAnFRESIDmLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRESIDm: <FLXAn_base> + 0130_H + (m – 1) × 4_H,
FLXAnFRESIDmL: <FLXAn_base> + 0130_H + (m – 1) × 4_H,
FLXAnFRESIDmLL: <FLXAn_base> + 0130_H + (m – 1) × 4_H,
FLXAnFRESIDmLH: <FLXAn_base> + 0130_H + (m – 1) × 4_H + 1_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXEB	RXEA	—	—	—	—	EID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.67 FLXAnFRESIDm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXEB	Received/Configured Even Sync ID on Channel B Flag 0: No sync frame received on channel B/node not configured to transmit sync frames 1: Sync frame received on channel B/node configured to transmit sync frames
14	RXEA	Received/Configured Even Sync ID on Channel A Flag 0: No sync frame received on channel A/node not configured to transmit sync frames 1: Sync frame received on channel A/node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	EID[9:0]	Even Sync ID (vsSyncIDListA,B even Flag)

(1) FLXAnFRESIDm.RXEB

Received/Configured Even Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXAnFRESID1.EID.
Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRESIDm.RXEA

Received/Configured Even Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXAnFRESID1.EID.
Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRESIDm.EID

Even Sync ID Flag (vsSyncIDListA,B even)

Indicates the sync frame ID received in an even communication cycle. Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.11 FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)

Registers FLXAnFROSID1 to FLXAnFROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXAnFROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFROSID1.RXOA, FLXAnFROSID1.RXOB are set. The value is updated during the NIT of each odd communication cycle.

Access: FLXAnFROSIDm is a read-only register that can be read in 32-bit units.
FLXAnFROSIDmL is a read-only register that can be read in 16-bit units.
FLXAnFROSIDmLL and FLXAnFROSIDmLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFROSIDm: <FLXAn_base> + 0170_H + (m – 1) × $4_{H,}$
FLXAnFROSIDmL: <FLXAn_base> + 0170_H + (m – 1) × $4_{H,}$
FLXAnFROSIDmLL: <FLXAn_base> + 0170_H + (m – 1) × $4_{H,}$
FLXAnFROSIDmLH: <FLXAn_base> + 0170_H + (m – 1) × 4_H + 1_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXOB	RXOA	—	—	—	—	OID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.68 FLXAnFROSIDm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXOB	Received/Configured Odd Sync ID on Channel B Flag 0: No sync frame received on channel B/node not configured to transmit sync frames 1: Sync frame received on channel B/node configured to transmit sync frames
14	RXOA	Received/Configured Odd Sync ID on Channel A Flag 0: No sync frame received on channel A/node not configured to transmit sync frames 1: Sync frame received on channel A/node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	OID[9:0]	Odd Sync ID Flag (vsSyncIDListA,B odd)

(1) FLXAnFROSIDm.RXOB

Received/Configured Odd Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXAnFROSID1.OID.
Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFROSIDm.RXOA

Received/Configured Odd Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXAnFROSID1.OID.
Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFROSIDm.OID

Odd Sync ID Flag (vsSyncIDListA,B odd)

Indicates the sync frame ID odd communication cycle.
Cleared when leaving CONFIG state or when entering STARTUP state.

25.2.7.12 FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)

The three network management registers hold the accrued NM vector (see **Section 25.3.7, Network Management**).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

NMVn-bytes exceeding the configured NM vector length are not valid.

For information about the byte alignment of the received NM vector in this register see **Section 25.3.17, Byte Alignment**.

Access: FLXAnFRNMVm is a read-only register that can be read in 32-bit units.
FLXAnFRNMVmL and FLXAnFRNMVmH are the read-only registers that can be read in 16-bit units.
FLXAnFRNMVmLL, FLXAnFRNMVmLH, FLXAnFRNMVmHL, and FLXAnFRNMVmHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRNMVm: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H,
FLXAnFRNMVmL: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H,
FLXAnFRNMVmH: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H + 2_H,
FLXAnFRNMVmLL: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H,
FLXAnFRNMVmLH: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H + 1_H,
FLXAnFRNMVmHL: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H + 2_H,
FLXAnFRNMVmHH: <FLXAn_base> + $01B0_H$ + (m – 1) × 4_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.69 FLXAnFRNMVm Register Contents

Bit Position	Bit Name	Function
31 to 0	NM[31:0]	NM Vector The three network management vector registers hold the accrued NM vector (configurable 0 to 12 bytes). The NM vector to be held is generated by bit-wise logic OR for each NM vector received on each channel (valid static frames with PPI = 1) (see **Section 25.3.7, Network Management**). For information about the byte alignment of the received NM vector in this register see **Section 25.3.17, Byte Alignment**. NMVn-bytes exceeding the configured NM vector length are not valid. The Register Contents are updated at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state. These bits are cleared when leaving CONFIG state or when entering STARTUP state.

25.2.8 Message Buffer Control Registers

25.2.8.1 FLXAnFRMRC — FlexRay Message RAM Configuration Register

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO.

The Message RAM can be divided into up three different areas; Static Buffer area, Static and Dynamic Buffer area, FIFO area. If present, the Static Buffer area is starting at Message Buffer 0.

The start of the Static and Dynamic Buffer area is configured by FLXAnFRMRC.FDB. FLXAnFRMRC.FDB defines the end of the Static Buffer area. If no Static Buffer area is present, the Static and Dynamic Buffer area starts at Message Buffer 0.

The start of the FIFO area is configured by FLXAnFRMRC.FFB. FLXAnFRMRC.FFB defines the end of the previous area, which can be either the Static Buffer area or the Static and Dynamic Buffer area. If no Static Buffer area and no Static and Dynamic Buffer area is present, the FIFO area starts at Message Buffer 0.

The end of the last configured area, which can be the Static Buffer area, the Static and Dynamic buffer area, or the FIFO area, is configured by FLXAnFRMRC.LCB.

Figure 25.2, Message RAM Organization shows an example configuration of the Message RAM where all there area are configured.

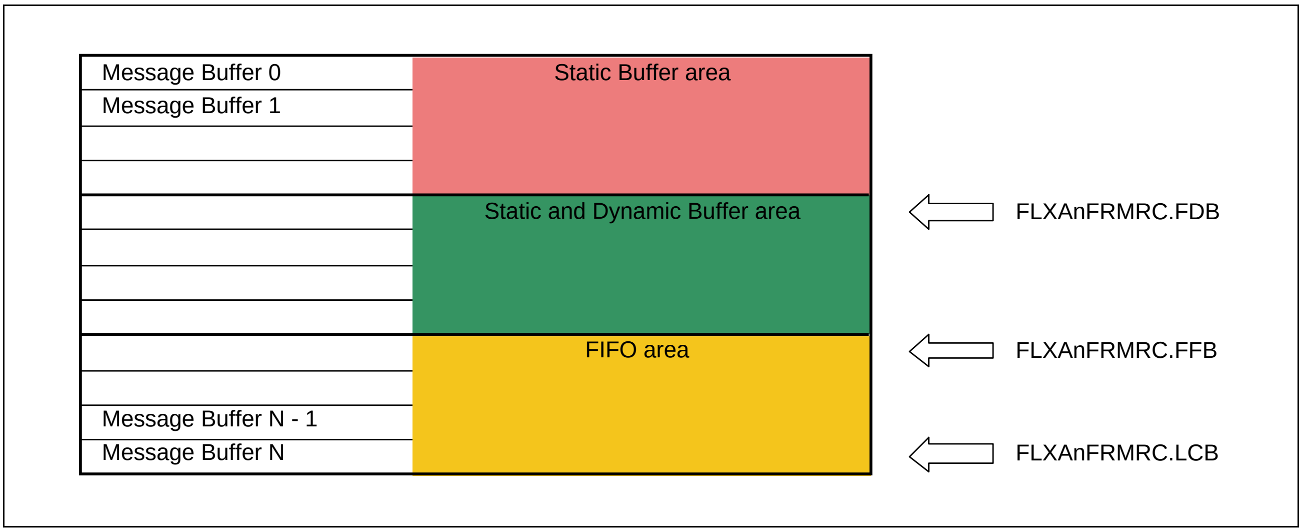


Figure 25.2 Message RAM Organization

CAUTIONS

1. If the node is configured as sync node (FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (FLXAnFRSUCC1.TSM = 1), message buffer 0 or 1 is reserved for sync frames or single slot frames and have to be configured with the node- specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 or 1 is treated like all other message buffers.
2. The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see **Section 25.3.13, Message RAM**.
3. If two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the “Static Buffers” or at the beginning of the “Static + Dynamic Buffers” section.

4. The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.
5. The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via FLXAnFRWRHS2.PL and FLXAnFRWRHS3.DP. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.

Access: FLXAnFRMRC can be read or written in 32-bit units.
FLXAnFRMRCL and FLXAnFRMRCH can be read or written in 16-bit units.
FLXAnFRMRCLL, FLXAnFRMRCLH, FLXAnFRMRCHL and FLXAnFRMRCHH can be read or written in 8-bit units.

Address: FLXAnFRMRC: <FLXAn_base> + 0300_H,
FLXAnFRMRCL: <FLXAn_base> + 0300_H, FLXAnFRMRCH: <FLXAn_base> + 0302_H,
FLXAnFRMRCLL: <FLXAn_base> + 0300_H, FLXAnFRMRCLH: <FLXAn_base> + 0301_H,
FLXAnFRMRCHL: <FLXAn_base> + 0302_H, FLXAnFRMRCHH: <FLXAn_base> + 0303_H

Value after reset: 0180 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPLM	SEC[1:0]		LCB[7:0]							
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FFB[7:0]								FDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.70 FLXAnFRMRC Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	SPLM	Sync Frame Payload Multiplex Bit 0: Only message buffer 0 locked against reconfiguration 1: Both message buffers 0 and 1 are locked against reconfiguration
25, 24	SEC[1:0]	Secure Buffers Bit 00_B: all buffers unlocked 01_B: static buffers locked, FIFO locked, limited transmission 10_B: all buffers locked 11_B: all buffers locked, limited transmission
23 to 16	LCB[7:0]	Last Configured Buffer Bit 0 to 127: Number of message buffers is FLXAnFRMRF.LCB + 1 128: No message buffer assigned to the FIFO
15 to 8	FFB[7:0]	First Buffer of FIFO Bit 0: All message buffers assigned to the FIFO 1 to 127: Message buffers from FLXAnFRMRC.FFB to FLXAnFRMRC.LCB assigned to the FIFO 128: No message buffer configured
7 to 0	FDB[7:0]	First Dynamic Buffer Bit 0: No group of message buffers exclusively for the static segment configured 1 to 127: Message buffers 0 to FLXAnFRMRC.FDB - 1 reserved for static segment 128: No dynamic message buffers configured

(1) FLXAnFRMRC.SPLM

Sync Frame Payload Multiplex Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
This bit is only evaluated if the node is configured as sync node (FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (FLXAnFRSUCC1.TSM = 1).
When this bit is set to 1 message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B.
When this bit is set to 0, sync frames are transmitted from message buffer 0 with the same payload data on all channels configured. Note that the channel filter configuration for message buffer 0 or message buffer 1 has to be chosen accordingly.

(2) FLXAnFRMRC.SEC

Secure Buffers Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state.
For temporary unlocking, see **Section 25.3.13.4, Host Handling of Access Errors**.

- 00_B = all buffers unlocked
  Reconfiguration of message buffers enabled with numbers < FLXAnFRMRC.FFB enabled
  - Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if FLXAnFRMRC.SPLM = 1, also message buffer 1) is always locked
- 01_B = static buffers locked, FIFO locked, limited transmission
  Reconfiguration of message buffers with numbers < FLXAnFRMRC.FDB and with numbers ≥ FLXAnFRMRC.FFB locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled
- 10_B = all buffers locked
  Reconfiguration of all message buffers locked
- 11_B = all buffers locked, limited transmission
  Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

(3) FLXAnFRMRC.LCB

Last Configured Buffer Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FDB.
When a FIFO area is configured (FLXAnFRMRC.FFB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FFB.

(4) FLXAnFRMRC.FFB

First Buffer of FIFO Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.FFB > FLXAnFRMRC.FDB.

(5) FLXAnFRMRC.FDB

First Dynamic Buffer Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

25.2.8.2 FLXAnFRFRF — FlexRay FIFO Rejection Filter Register

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO.

Access: FLXAnFRFRF can be read or written in 32-bit units.
FLXAnFRFRFL and FLXAnFRFRFH can be read or written in 16-bit units.
FLXAnFRFRFLL, FLXAnFRFRFLH, FLXAnFRFRFLH, FLXAnFRFRFHL, and FLXAnFRFRFHH can be read or written in 8-bit units.

Address: FLXAnFRFRF: <FLXAn_base> + 0304_H,
FLXAnFRFRFL: <FLXAn_base> + 0304_H, FLXAnFRFRFH: <FLXAn_base> + 0306_H,
FLXAnFRFRFLL: <FLXAn_base> + 0304_H, FLXAnFRFRFLH: <FLXAn_base> + 0305_H,
FLXAnFRFRFHL: <FLXAn_base> + 0306_H, FLXAnFRFRFHH: <FLXAn_base> + 0307_H

Value after reset: 0180 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNF	RSS	CYF[6:0]						
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FID[10:0]											CH[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.71 FLXAnFRFRF Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	RNF	Reject Null Frames Bit 0: Null frames are stored in the FIFO 1: Reject all null frames
23	RSS	Reject in Static Segment Bit 0: FIFO also used for static segment 1: Reject messages in static segment
22 to 16	CYF[6:0]	Cycle Counter Filter Bit Cycle Counter Filter
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	FID[10:0]	Frame ID Filter Bit 0 to 2047: Frame ID filter values
1, 0	CH[1:0]	Channel Filter Bit 00_B: receive on both channels 01_B: receive only on channel B 10_B: receive only on channel A 11_B: no reception

(1) FLXAnFRFRF.RNF

Reject Null Frames Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
If this bit is set to 1, received null frames are not stored in the FIFO.

(2) FLXAnFRFRF.RSS

Reject in Static Segment Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
If this bit is set to 1, the FIFO is used only for the dynamic segment.

(3) FLXAnFRFRF.CYF

Cycle Counter Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected. For details about the configuration of the cycle counter filter see **Section 25.3.8.2, Cycle Counter Filtering**.

(4) FLXAnFRFRF.FID

Frame ID Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXAnFRFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FLXAnFRFRFM.MFID is zero, a frame ID filter value of zero means that no frame ID is rejected.

(5) FLXAnFRFRF.CH

Channel Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

25.2.8.3 FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register

The FlexRay FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set to 1, it indicates that the corresponding bit in the FLXAnFRFRF register will not be considered for rejection filtering.

Access: FLXAnFRFRFM can be read or written in 32-bit units.
FLXAnFRFRFML can be read or written in 16-bit units.
FLXAnFRFRFMLL and FLXAnFRFRFMLH can be read or written in 8-bit units.

Address: FLXAnFRFRFM: <FLXAn_base> + 0308$_{\text{H}}$,
FLXAnFRFRFML: <FLXAn_base> + 0308$_{\text{H}}$,
FLXAnFRFRFMLL: <FLXAn_base> + 0308$_{\text{H}}$, FLXAnFRFRFMLH: <FLXAn_base> + 0309$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MFID[10:0]											—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.72 FLXAnFRFRFM Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	MFID[10:0]	Mask Frame ID Filter Bit 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRFRFM.MFID

Mask Frame ID Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

25.2.8.4 FLXAnFRFCL — FlexRay FIFO Critical Level Register

Access:	FLXAnFRFCL can be read or written in 32-bit units. FLXAnFRFCLL can be read or written in 16-bit units. FLXAnFRFCLLL can be read or written in 8-bit units.
Address:	FLXAnFRFCL: <FLXAn_base> + $030C_H$, FLXAnFRFCLL: <FLXAn_base> + $030C_H$, FLXAnFRFCLLL: <FLXAn_base> + $030C_H$
Value after reset:	$0000\ 0080_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.73 FLXAnFRFCL Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CL[7:0]	Critical Level Bit Critical Level

(1) FLXAnFRFCL.CL

Critical Level Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.
When the receive FIFO fill level (FLXAnFRFSR.RFFL) is equal or greater than the critical level configured by FLXAnFRFCL.CL, the receive FIFO critical level flag FLXAnFRFSR.RFCL is set to 1.
If FLXAnFRFCL.CL is programmed to values > 128, bit FLXAnFRFSR.RFCL is never set to 1.

25.2.9 Message Buffer Status Registers

25.2.9.1 FLXAnFRMHDS — FlexRay Message Handler Status Register

Access: FLXAnFRMHDS can be read or written in 32-bit units.
FLXAnFRMHDSL can be read or written in 16-bit units.
FLXAnFRMHDSH is a read-only register that can be read in 16-bit units.
FLXAnFRMHDSLL can be read or written in 8-bit units.
FLXAnFRMHDSLH, FLXAnFRMHDSHL, and FLXAnFRMHDSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMHDS: <FLXAn_base> + 0310$_H$,
FLXAnFRMHDSL: <FLXAn_base> + 0310$_H$, FLXAnFRMHDSH: <FLXAn_base> + 0312$_H$,
FLXAnFRMHDSLL: <FLXAn_base> + 0310$_H$, FLXAnFRMHDSLH: <FLXAn_base> + 0311$_H$,
FLXAnFRMHDSHL: <FLXAn_base> + 0312$_H$, FLXAnFRMHDSHH: <FLXAn_base> + 0313$_H$

Value after reset: 0000 0080$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MBU[6:0]							—	MBT[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	FMB[6:0]							CRAM	MFMB	FMBD	ATBF2	ATBF1	AMR	—	—
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.74 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MBU[6:0]	Message Buffer Updated Flag
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MBT[6:0]	Message Buffer Transmitted Flag
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	FMB[6:0]	Faulty Message Buffer Number Flag
7	CRAM	Clear all internal RAM's Flag 0: No execution of the CHI command CLEAR_RAMS 1: Execution of the CHI command CLEAR_RAMS ongoing
6	MFMB	Multiple Faulty Message Buffer Detection Flag 0: No additional faulty message buffer. 1: Additional faulty message buffer was detected while the FLXAnFRMHDS.FMBD flag is set to 1.
5	FMBD	Faulty Message Buffer Detection Flag 0: No faulty message buffer. 1: Message buffer referenced by FLXAnFRMHDS.FMBD holds faulty data with a parity error.
4	ATBF2	Nonresident Buffer RAM B Access Error Flag 0: No access error 1: Access error occurred when reading the RAM B.

Table 25.74 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
3	ATBF1	Nonresident Buffer RAM A Access Error Flag 0: No access error. 1: Access error occurred when reading the RAM A.
2	AMR	Message RAM Access Error Flag 0 = No access error 1 = Access error occurred when reading the Message RAM.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRMHDS.MBU

Message Buffer Updated Flag

Number of message buffer that was updated last by the CC. For this message buffer the respective ND and/or MBC flag in the FLXAnFRNDAT1/2/3/4 registers and the FLXAnFRMBSC1/2/3/4 registers are also set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDS.MBT

Message Buffer Transmitted Flag

Number of last successfully transmitted message buffer.
If the message buffer is configured for single-shot mode, the respective TXR flag in the FLXAnFRTXRQ1/2/3/4 registers was reset to 0.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDS.FMB

Faulty Message Buffer Number Flag

This flag indicates that an access error occurred when reading from the message buffer referenced by FLXAnFRMHDS.FMB.
The value of this flag is only valid when FLXAnFRMHDS.AMR and flag FLXAnFRMHDS.FMBD are set to 1.
This flag is not updated while the FLXAnFRMHDS.FMBD flag is 1.
This flag is cleared by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDS.CRAM

Internal RAM Clear Flag

This flag indicates that the CHI command CLEAR_RAMS is ongoing (all bits of the message RAM, input buffer, output buffer and nonresident buffer are written to 0).
This flag is set by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDS.MFMB

Multiple Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.
This bit is cleared when writing 1 to it.
This bit indicates that an additional faulty message buffer was detected while the FMBD flag is set.
This bit is cleared by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDS.FMBD

Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.
This bit is cleared when writing 1 to it.
This bit indicates that the message buffer holds faulty data due to an access error.
This bit is cleared by the CHI command CLEAR_RAMS.

(7) FLXAnFRMHD.ATBF2

Nonresident Buffer RAM B Access Error Flag

Writing 0 has no effect on the bit value.
This bit is cleared when writing 1 to it.
This flag indicates that an access error occurred when reading the RAM B.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1.
This flag can be reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHD.ATBF1

Nonresident Buffer RAM A Access Error Flag

Writing 0 has no effect on the bit value.
This bit is cleared when writing 1 to it.
This flag indicates that an access error occurred when reading the RAM A.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1.
This flag can be reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDS.AMR

Message RAM Access Error Flag

Writing 0 has no effect on the bit value.
This bit is cleared when writing 1 to it.
This flag indicates that an access error occurred when reading the Message RAM.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1.
This flag can be reset by the CHI command CLEAR_RAMS.

25.2.9.2 FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register

Access: FLXAnFRLDTS is a read-only register that can be read in 32-bit units.
FLXAnFRLDTSL and FLXAnFRLDTSH are the read-only registers that can be read in 16-bit units.
FLXAnFRLDTSLL, FLXAnFRLDTSLH, FLXAnFRLDTSHL, and FLXAnFRLDTSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRLDTS: <FLXAn_base> + 0314_H,
FLXAnFRLDTSL: <FLXAn_base> + 0314_H, FLXAnFRLDTSH: <FLXAn_base> + 0316_H,
FLXAnFRLDTSLL: <FLXAn_base> + 0314_H, FLXAnFRLDTSLH: <FLXAn_base> + 0315_H,
FLXAnFRLDTSHL: <FLXAn_base> + 0316_H, FLXAnFRLDTSHH: <FLXAn_base> + 0317_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	LDTB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LDTA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.75 FLXAnFRLDTS Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	LDTB[10:0]	Last Dynamic Transmission Channel B Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	LDTA[10:0]	Last Dynamic Transmission Channel A Flag

(1) FLXAnFRLDTS.LDTB

Last Dynamic Transmission Channel B Flag

Stores the value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node.
It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS of channel B.

(2) FLXAnFRLDTS.LDTA

Last Dynamic Transmission Channel A Flag

Stores the value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node.
It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS of channel A.

25.2.9.3 FLXAnFRFSR — FlexRay FIFO Status Register

Access: FLXAnFRFSR is a read-only register that can be read in 32-bit units.
FLXAnFRFSRL is a read-only register that can be read in 16-bit units.
FLXAnFRFSRLL and FLXAnFRFSRLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRFSR: <FLXAn_base> + 0318$_H$,
FLXAnFRFSRL: <FLXAn_base> + 0318$_H$,
FLXAnFRFSRLL: <FLXAn_base> + 0318$_H$, FLXAnFRFSRLH: <FLXAn_base> + 0319$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFFL[7:0]								—	—	—	—	—	RFO	RFCL	RFNE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.76 FLXAnFRFSR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	RFFL[7:0]	Receive FIFO Fill Level Flag
7 to 3	Reserved	When read, the value after reset is returned.
2	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
1	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
0	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty

(1) FLXAnFRFSR.RFFL

Receive FIFO Fill Level Flag

Indicates the number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRFSR.RFO

Receive FIFO Overrun Flag

The flag is set to 1 by the CC when a receive FIFO overrun is detected.
When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag FLXAnFREIR.RFO is set to 1.
The flag is cleared by the next FIFO read access issued by the Host.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRFSR.RFCL

Receive FIFO Critical Level Flag

This flag is set to 1 when the receive FIFO fill level FLXAnFRFSR.RFFL is equal or greater than the critical level as configured by FLXAnFRFCL.CL.
When FLXAnFRFSR.RFCL changes from 0 to 1 bit FLXAnFRSIR.RFCL is set to 1, and if enabled, an interrupt is generated.
The flag is cleared by the CC as soon as FLXAnFRFSR.RFFL drops below FLXAnFRFCL.CL.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRFSR.RFNE

Receive FIFO Not Empty Flag

This flag is set to 1 by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag FLXAnFRSIR.RFNE is set to 1.
The bit is reset to 0 after the Host has read all messages from the FIFO.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

25.2.9.4 FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register

Some constraints exist for the Message Handler regarding bus clock frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FLXAnFRMHDF.

Access:	FLXAnFRMHDF can be read or written in 32-bit units. FLXAnFRMHDFL can be read or written in 16-bit units. FLXAnFRMHDFLL and FLXAnFRMHDFLH can be read or written in 8-bit units.
Address:	FLXAnFRMHDF: <FLXAn_base> + $031C_H$, FLXAnFRMHDFL: <FLXAn_base> + $031C_H$, FLXAnFRMHDFLL: <FLXAn_base> + $031C_H$, FLXAnFRMHDFLH: <FLXAn_base> + $031D_H$
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.77 FLXAnFRMHDF Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	WAHP	Write Attempt to Header Partition Flag 0: No write attempt to header partition 1: Write attempt to header partition
7	TNSB	Transmission Not Started Channel B Flag 0: No transmission not started on channel B 1: Transmission not started on channel B
6	TNSA	Transmission Not Started Channel A Flag 0: No transmission not started on channel A 1: Transmission not started on channel A
5	TBFB	Temporary buffer Access Failure B Flag 0: No TBF B access failure 1: TBF B access failure
4	TBFA	Temporary buffer Access Failure A Flag 0: No TBF A access failure 1: TBF A access failure
3	FNFB	Find Sequence Not Finished Channel B Flag 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B
2	FNFA	Find Sequence Not Finished Channel A Flag 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A
1	SNUB	Status Not Updated Channel B Flag 0: No overload condition occurred when updating MBS for channel B 1: MBS for channel B not updated

Table 25.77 FLXAnFRMHDF Register Contents

Bit Position	Bit Name	Function
0	SNUA	Status Not Updated Channel A Flag 0: No overload condition occurred when updating MBS for channel A 1: MBS for channel A not updated

(1) FLXAnFRMHDF.WAHP

Write Attempt to Header Partition Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
Outside DEFAULT_CONFIG and CONFIG state this flag is set to 1 by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDF.TNSB

Transmission Not Started Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when the Message Handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDF.TNSA

Transmission Not Started Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDF.TBFB

Temporary buffer Access Failure B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when a read or write access to TBF B requested by PRT (Protocol controller) B could not complete within the available time.
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDF.TBFA

Temporary buffer Access Failure A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when a read or write access to TBF A requested by PRT A could not complete within the available time.
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDF.FNFB

Find Sequence Not Finished Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(7) FLXAnFRMHDF.FNFA

Find Sequence Not Finished Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHDF.SNUB

Status Not Updated Channel B Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer’s status (FLXAnFRMBS).
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDF.SNUA

Status Not Updated Channel A Flag

Writing 0 has no effect on the bit value.
This bit is cleared by writing 1 to it.
This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer’s status (FLXAnFRMBS).
When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.
Reset by the CHI command CLEAR_RAMS.

25.2.9.5 FLXAnFRTXRQi — FlexRay Transmission Request i (i = 1 to 4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are valid for transmit buffers only. If the number of configured message buffers is less than 128, the unused TXR flags are invalid.

Access: FLXAnFRTXRQi is a read-only register that can be read in 32-bit units.
FLXAnFRTXRQiL and FLXAnFRTXRQiH are the read-only registers that can be read in 16-bit units.
FLXAnFRTXRQiLL, FLXAnFRTXRQiLH, FLXAnFRTXRQiHL, and FLXAnFRTXRQiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRTXRQi: <FLXAn_base> + 0320_H + (i – 1) × 4_H,
FLXAnFRTXRQiL: <FLXAn_base> + 0320_H + (i – 1) × 4_H,
FLXAnFRTXRQiH: <FLXAn_base> + 0320_H + (i – 1) × 4_H + 2_H,
FLXAnFRTXRQiLL: <FLXAn_base> + 0320_H + (i – 1) × 4_H,
FLXAnFRTXRQiLH: <FLXAn_base> + 0320_H + (i – 1) × 4_H + 1_H,
FLXAnFRTXRQiHL: <FLXAn_base> + 0320_H + (i – 1) × 4_H + 2_H,
FLXAnFRTXRQiHH: <FLXAn_base> + 0320_H + (i – 1) × 4_H + 3_H,

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXRo (o = (i – 1) × 32 + 31 to (i – 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXRo (o = (i – 1) × 32 + 15 to (i – 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.78 FLXAnFRTXRQi Register Contents

Bit Position	Bit Name	Function
31 to 0	TXRo	Transmission Request Flag o

(1) FLXAnFRTXRQi.TXRo (o = (i – 1) × 32 to i × 32 – 1)

Transmission Request Flag o

If the flag is set to 1, the respective message buffer is ready for transmission or transmission of this message buffer is in progress.
In single-shot mode the flags are reset to 0 after transmission has completed.
This bit is cleared by the CHI command CLEAR_RAMS.

25.2.9.6 FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, unused ND flags are invalid.

Access: FLXAnFRNDATi is a read-only register that can be read in 32-bit units.
FLXAnFRNDATiL and FLXAnFRNDATiH are the read-only registers that can be read in 16-bit units.
FLXAnFRNDATiLL, FLXAnFRNDATiLH, FLXAnFRNDATiHL, and FLXAnFRNDATiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRNDATi: <FLXAn_base> + 0330_H + (i – 1) × 4_H,
FLXAnFRNDATiL: <FLXAn_base> + 0330_H + (i – 1) × 4_H,
FLXAnFRNDATiH: <FLXAn_base> + 0330_H + (i – 1) × 4_H + 2_H,
FLXAnFRNDATiLL: <FLXAn_base> + 0330_H + (i – 1) × 4_H,
FLXAnFRNDATiLH: <FLXAn_base> + 0330_H + (i – 1) × 4_H + 1_H,
FLXAnFRNDATiHL: <FLXAn_base> + 0330_H + (i – 1) × 4_H + 2_H,
FLXAnFRNDATiHH: <FLXAn_base> + 0330_H + (i – 1) × 4_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NDm (m = (i – 1) × 32 + 31 to (i – 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDm (m = (i – 1) × 32 + 15 to (i – 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.79 FLXAnFRNDATi Register Contents

Bit Position	Bit Name	Function
31 to 0	NDm	New Data Flag m

(1) FLXAnFRNDATi.NDm (m = (i – 1) × 32 to i × 32 – 1)

New Data Flag m

The flags are set to 1 when a valid received data frame matches the message buffer’s filter configuration, independent of the payload length received or the payload length configured for that message buffer.
The flags are not set to 1 after reception of null frames except for message buffers belonging to the receive FIFO.
An ND flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.
Cleared when leaving CONFIG state or when entering STARTUP state.
This bit is cleared by the CHI command CLEAR_RAMS.

25.2.9.7 FLXAnFRMBSCi — FlexRay Message Buffer Status Changed Register i (i = 1 to 4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, unused MBC flags are invalid.

Access: FLXAnFRMBSCi is a read-only register that can be read in 32-bit units.
FLXAnFRMBSCiL and FLXAnFRMBSCiH are the read-only registers that can be read in 16-bit units.
FLXAnFRMBSCiLL, FLXAnFRMBSCiLH, FLXAnFRMBSCiHL, and FLXAnFRMBSCiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMBSCi: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$,
FLXAnFRMBSCiL: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$,
FLXAnFRMBSCiH: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$ + 2$_H$,
FLXAnFRMBSCiLL: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$,
FLXAnFRMBSCiLH: <FLXAn_base> + 0340$_H$+ (i – 1) × 4$_H$ + 1$_H$,
FLXAnFRMBSCiHL: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$ + 2$_H$,
FLXAnFRMBSCiHH: <FLXAn_base> + 0340$_H$ + (i – 1) × 4$_H$ + 3$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBCm (m = (i – 1) × 32 + 31 to (i – 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBCm (m = (i – 1) × 32+ 15 to (i – 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.80 FLXAnFRMBSCi Register Contents

Bit Position	Bit Name	Function
31 to 0	MBCm	Message Buffer Status Changed Flag m

(1) FLXAnFRMBSCi.MBCm (m = (i – 1) × 32 to i × 32 – 1)

Message Buffer Status Changed Flag m

Indicates whether the Message Handler has changed one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, FTB in the header section (see **Section 25.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** and **Section 25.3.13.1, Header Partition**) of the respective message buffer.
An MBC flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.
Cleared when leaving CONFIG state or when entering STARTUP state.
This bit is cleared by the CHI command CLEAR_RAMS.

25.2.10 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in **Section 25.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via FLXAnFRWRHS2.PLC and FLXAnFRWRHS3.DP. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in **Section 25.3.12.1, Reconfiguration of Message Buffers**.

These registers cannot be written when the input data transfer function shown in **Section 25.3.16.1, Input Data Transfer** is used and the FLXAnFRITS.ITS bit is 1.

25.2.10.1 FLXAnFRWRDSx — FlexRay Write Data Section Register x (x = 1 to 64)

This register holds the data wards to be transferred to the data section of the specified message buffer. The number of data words written to the Message RAM is defined by the payload length configured in FLXAnFRWRHS2.PLC bit.

Access: FLXAnFRWRDSx can be read or written in 32-bit units.
FLXAnFRWRDSxL and FLXAnFRWRDSxH can be read or written in 16-bit units.
FLXAnFRWRDSxLL, FLXAnFRWRDSxLH, FLXAnFRWRDSxHL, and FLXAnFRWRDSxHH can be read or written in 8-bit units.

Address: FLXAnFRWRDSx: <FLXAn_base> + 0400_H + (x – 1) × 4_H,
FLXAnFRWRDSxL: <FLXAn_base> + 0400_H + (x – 1) × 4_H,
FLXAnFRWRDSxH: <FLXAn_base> + 0400_H + (x – 1) × 4_H + 2_H,
FLXAnFRWRDSxLL: <FLXAn_base> + 0400_H + (x – 1) × 4_H,
FLXAnFRWRDSxLH: <FLXAn_base> + 0400_H + (x – 1) × 4_H + 1_H,
FLXAnFRWRDSxHL: <FLXAn_base> + 0400_H + (x – 1) × 4_H + 2_H,
FLXAnFRWRDSxHH: <FLXAn_base> + 0400_H + (x – 1) × 4_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.81 FLXAnFRWRDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data Bit

(1) FLXAnFRWRDSx.MD

Message Data Bit

For information about the byte alignment of the message data in this register see **Section 25.3.17, Byte Alignment**.

CAUTIONS

1. In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.
2. When writing to FLXAnFRWRDSx, each 32-bit word has to be filled up by one 32-bit access OR two consecutive 16-bit accesses OR four consecutive 8- bit accesses before the transfer from the Input Buffer to the Message RAM is started. If not all bytes of a 32- bit word have been written by the Host (8/16-bit access only), FLXAnFRWRDSx holds partly undefined data.
Reset by the CHI command CLEAR_RAMS.

25.2.10.2 FLXAnFRWRHS1 — FlexRay Write Header Section Register 1

Access: FLXAnFRWRHS1 can be read or written in 32-bit units.
FLXAnFRWRHS1L and FLXAnFRWRHS1H can be read or written in 16-bit units.
FLXAnFRWRHS1LL, FLXAnFRWRHS1LH, FLXAnFRWRHS1HL, and FLXAnFRWRHS1HH can be read or written in 8-bit units.

Address: FLXAnFRWRHS1: <FLXAn_base> + 0500_H,
FLXAnFRWRHS1L: <FLXAn_base> + 0500_H, FLXAnFRWRHS1H: <FLXAn_base> + 0502_H,
FLXAnFRWRHS1LL: <FLXAn_base> + 0500_H, FLXAnFRWRHS1LH: <FLXAn_base> + 0501_H,
FLXAnFRWRHS1HL: <FLXAn_base> + 0502_H, FLXAnFRWRHS1HH: <FLXAn_base> + 0503_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.82 FLXAnFRWRHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MBI	Message Buffer Interrupt Bit 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled
28	TXM	Transmission Mode Setting Bit 0: Continuous mode 1: Single-shot mode
27	PPIT	Payload Preamble Indicator Transmit Bit 0: Payload Preamble Indicator is set to 0 1: Payload Preamble Indicator is set to 1
26	CFG	Message Buffer Direction Configuration Bit 0: The corresponding buffer is configured as Receive Buffer 1: The corresponding buffer is configured as Transmit Buffer
25, 24	CH[1:0]	Channel Filter Control Bit
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CYC[6:0]	Cycle Code Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	FID[10:0]	Frame ID Bit

(1) FLXAnFRWRHS1.MBI

Message Buffer Interrupt Enable Bit

This bit enables the message buffer interrupt.
After a dedicated receive buffer has been updated by the Message Handler, flag FLXAnFRSIR.RXI and /or FLXAnFRSIR.MBSI are set to 1. After a transmission has completed flag FLXAnFRSIR.TXI is set to 1.

(2) FLXAnFRWRHS1.TXM

Transmission Mode Setting Bit

This bit selects transmit mode of the corresponding message buffer. For transmit mode, see **Section 25.3.9.3, Transmit Buffers**.

(3) FLXAnFRWRHS1.PPIT

Payload Preamble Indicator Transmit Bit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames in transmit frames of the corresponding message buffer.
If the bit is set to 1 in a static message buffer, the respective message buffer holds network management information.
If the bit is set to 1 in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the Host.

(4) FLXAnFRWRHS1.CFG

Message Buffer Direction Configuration Bit

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated.
When not allocating an unused area of at least 32 bits at the start of the data partition, set this bit to 1 in the data section of the message buffer allocated immediately after the (last buffer of the) header partition so that the message buffer is specified as a transmit buffer.

(5) FLXAnFRWRHS1.CH

Channel Filter Control Bit

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

CH[1:0]	Transmit Buffer Transmit Frame on	Receive Buffer Store Frame Received from
00_B	No transmission	Ignore frame
01_B	Channel A	Channel A
10_B	Channel B	Channel B
11_B	Both channels (static segment only)	Channel A or B (store first semantically valid frame; static segment only)

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as CH = “00_B”)

(6) FLXAnFRWRHS1.CYC

Cycle Code Bit

The 7-bit cycle code determines the cycle set used for cycle counter filtering.
For details about the configuration of the cycle code **Section 25.3.8.2, Cycle Counter Filtering**.

(7) FLXAnFRWRHS1.FID

Frame ID Bit

Configures the frame ID of the selected message buffer. The frame ID defines the slot number for transmission/reception of the respective message.
Message buffers with frame ID = 0 are considered as not valid.

25.2.10.3 FLXAnFRWRHS2 — FlexRay Write Header Section Register 2

Access:	FLXAnFRWRHS2 can be read or written in 32-bit units. FLXAnFRWRHS2L and FLXAnFRWRHS2H can be read or written in 16-bit units. FLXAnFRWRHS2LL, FLXAnFRWRHS2LH, and FLXAnFRWRHS2HL can be read or written in 8-bit units.
Address:	FLXAnFRWRHS2: <FLXAn_base> + 0504_H, FLXAnFRWRHS2L: <FLXAn_base> + 0504_H, FLXAnFRWRHS2H: <FLXAn_base> + 0506_H, FLXAnFRWRHS2LL: <FLXAn_base> + 0504_H, FLXAnFRWRHS2LH: <FLXAn_base> + 0505_H, FLXAnFRWRHS2HL: <FLXAn_base> + 0506_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.83 FLXAnFRWRHS2 Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	PLC[6:0]	Payload Length Configured Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	CRC[10:0]	Header CRC Bit (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC is configured

(1) FLXAnFRWRHS2.PLC

Payload Length Configured Bit

Length of data section (number of 2-byte words) as configured by the Host.
During static segment the static frame payload length as configured by FLXAnFRMHDC.SFDL defines the payload length for all static frames. If the payload length configured by FLXAnFRWRHS2.PLC is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is “0000_H” (see **Section 25.3.9.3, Transmit Buffers**).

(2) FLXAnFRWRHS2.CRC

Header CRC Bit (vRF!Header!HeaderCRC)

Setting of the receive buffer is not required.
Transmitting of the message buffer needs the header CRC calculation and setting.
For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by FLXAnFRMHDC.SFDL.

25.2.10.4 FLXAnFRWRHS3 — FlexRay Write Header Section Register 3

Access: FLXAnFRWRHS3 can be read or written in 32-bit units.
FLXAnFRWRHS3L can be read or written in 16-bit units.
FLXAnFRWRHS3LL and FLXAnFRWRHS3LH can be read or written in 8-bit units.

Address: FLXAnFRWRHS3: <FLXAn_base> + 0508$_{H}$,
FLXAnFRWRHS3L: <FLXAn_base> + 0508$_{H}$,
FLXAnFRWRHS3LL: <FLXAn_base> + 0508$_{H}$, FLXAnFRWRHS3LH: <FLXAn_base> + 0509$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.84 FLXAnFRWRHS3 Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	DP[10:0]	Data Pointer Bit

(1) FLXAnFRWRHS3.DP

Data Pointer Bit

Configures the pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

25.2.10.5 FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register

Configures how the message buffer in the Message RAM selected by register FLXAnFRIBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits FLXAnFRIBCM.LHSH, FLXAnFRIBCM.LDSH, and FLXAnFRIBCM.STXRH are swapped with bits FLXAnFRIBCM.LHSS, FLXAnFRIBCM.LDSS, and FLXAnFRIBCM.STXRS.

Access: FLXAnFRIBCM can be read or written in 32-bit units.
FLXAnFRIBCML can be read or written in 16-bit units.
FLXAnFRIBCMH is a read-only register that can be read in 16-bit units.
FLXAnFRIBCMLL can be read or written in 8-bit units.
FLXAnFRIBCMHL is a read-only register that can be read in 8-bit units.

Address: FLXAnFRIBCM: <FLXAn_base> + 0510_H,
FLXAnFRIBCML: <FLXAn_base> + 0510_H, FLXAnFRIBCMH: <FLXAn_base> + 0512_H,
FLXAnFRIBCMLL: <FLXAn_base> + 0510_H, FLXAnFRIBCMHL: <FLXAn_base> + 0512_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRS	LDSS	LHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRH	LDSH	LHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 25.85 FLXAnFRIBCM Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	STXRS	Set Transmission Request Shadow Flag 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished)
17	LDSS	Load Data Section Shadow Flag 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
16	LHSS	Load Header Section Shadow Flag 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	STXRH	Set Transmission Request Host Bit 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission
1	LDSH	Load Data Section Host Bit 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM
0	LHSH	Load Header Section Host Bit 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM

(1) FLXAnFRIBCM.STXRS

Set Transmission Request Shadow Flag

(2) FLXAnFRIBCM.LDSS

Load Data Section Shadow Flag

(3) FLXAnFRIBCM.LHSS

Load Header Section Shadow Flag

(4) FLXAnFRIBCM.STXRH

Set Transmission Request Host Bit

If this bit is set to 1, the TXR flag for the selected message buffer is set in the FLXAnFRTXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed.
TXR is evaluated for transmit buffers only.

(5) FLXAnFRIBCM.LDSH

Set Load Data Section Host Bit

(6) FLXAnFRIBCM.LHSH

Set Load Header Section Host Bit

25.2.10.6 FLXAnFRIBCR — FlexRay Input Buffer Command Request Register

When the Host writes the number of the target message buffer in the Message RAM to FLXAnFRIBCR.IBRH, IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also swapped (see **Section 25.3.12.2(1), Data Transfer from Input Buffer to Message RAM**).

With this write operation the FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by FLXAnFRIBCR.IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0.

FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also swapped.

Any write access to an Input Buffer register while both FLXAnFRIBCR.IBSYS and FLXAnFRIBCR.IBSYH are set to 1 will cause the error flag FLXAnFREIR.IIBA to be set to 1.

Access: FLXAnFRIBCR can be read or written in 32-bit units.
FLXAnFRIBCRL can be read or written in 16-bit units.
FLXAnFRIBCRH is a read-only register that can be read in 16-bit units.
FLXAnFRIBCRLL can be read or written in 8-bit units.
FLXAnFRIBCRLH, FLXAnFRIBCRHL, and FLXAnFRIBCRHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRIBCR: <FLXAn_base> + 0514_H,
FLXAnFRIBCRL: <FLXAn_base> + 0514_H, FLXAnFRIBCRH: <FLXAn_base> + 0516_H,
FLXAnFRIBCRLL: <FLXAn_base> + 0514_H, FLXAnFRIBCRLH: <FLXAn_base> + 0515_H,
FLXAnFRIBCRHL: <FLXAn_base> + 0516_H, FLXAnFRIBCRHH: <FLXAn_base> + 0517_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IBSYS	—	—	—	—	—	—	—	—	IBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IBSYH	—	—	—	—	—	—	—	—	IBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.86 FLXAnFRIBCR Register Contents

Bit Position	Bit Name	Function
31	IBSYS	Input Buffer Busy Shadow Flag 0: Transfer between IBF Shadow and Message RAM completed 1: Transfer between IBF Shadow and Message RAM in progress
30 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IBRS[6:0]	Input Buffer Request Shadow Flag
15	IBSYH	Input Buffer Busy Host Flag 0: No request pending 1: Request while transfer between IBF Shadow and Message RAM in progress
14 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	IBRH[6:0]	Input Buffer Request Host Bit

(1) FLXAnFRIBCR.IBSYS

Input Buffer Busy Shadow Flag

Set to 1 after writing FLXAnFRIBCR.IBRH.
This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.
When the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0.

(2) FLXAnFRIBCR.IBRS

Input Buffer Request Shadow Flag

Number of the target message buffer actually updated/lately updated.

(3) FLXAnFRIBCR.IBSYH

Input Buffer Busy Host Flag

Set to 1 by writing FLXAnFRIBCR.IBRH while FLXAnFRIBCR.IBSYS is still 1.
This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.
After the ongoing transfer between IBF Shadow and the Message RAM has completed, the FLXAnFRIBCR.IBSYH is set back to 0.

(4) FLXAnFRIBCR.IBRH

Input Buffer Request Host Bit

Selects the target message buffer in the Message RAM for data transfer from Input Buffer.

25.2.11 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in **Section 25.3.12.2(2), Data Transfer from Message RAM to Output Buffer**.

These registers cannot be written when the output data transfer function shown in **Section 25.3.16.2, Output Data Transfer**, in Output Data Transfer is used and the FLXAnFROTS.OTS bit is 1.

25.2.11.1 FLXAnFRRDDSx — FlexRay Read Data Section Register x (x = 1 to 64)

Holds the data words read from the data section of the addressed message buffer. This register holds the data wards to be transferred to the data section of the specified message buffer. The number of data words (DWn) read from the Message RAM is defined by the payload length configured in FLXAnFRRDHS2.PLC bit.

Access: FLXAnFRRDDSx is a read-only register that can be read in 32-bit units.
FLXAnFRRDDSxL and FLXAnFRRDDSxH are the read-only registers that can be read in 16-bit units.
FLXAnFRRDDSxLL, FLXAnFRRDDSxLH, FLXAnFRRDDSxHL, and FLXAnFRRDDSxHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDDSx: <FLXAn_base> + 0600_H + (x – 1) × 4_H,
FLXAnFRRDDSxL: <FLXAn_base> + 0600_H + (x – 1) × 4_H,
FLXAnFRRDDSxH: <FLXAn_base> + 0600_H + (x – 1) × 4_H + 2_H,
FLXAnFRRDDSxLL: <FLXAn_base> + 0600_H + (x – 1) × 4_H,
FLXAnFRRDDSxLH: <FLXAn_base> + 0600_H + (x – 1) × 4_H + 1_H,
FLXAnFRRDDSxHL: <FLXAn_base> + 0600_H + (x – 1) × 4_H + 2_H,
FLXAnFRRDDSxHH: <FLXAn_base> + 0600_H + (x – 1) × 4_H + 3_H,

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.87 FLXAnFRRDDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data

(1) FLXAnFRRDDSx.MD

Message Data Flag

For information about the byte alignment of the data words in this register see **Section 25.3.17, Byte Alignment**.

CAUTION

In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.
Reset by the CHI command CLEAR_RAMS.

25.2.11.2 FLXAnFRRDHS1 — FlexRay Read Header Section Register 1

Access: FLXAnFRRDHS1 is a read-only register that can be read in 32-bit units.
FLXAnFRRDHS1L and FLXAnFRRDHS1H are the read-only registers that can be read in 16-bit units.
FLXAnFRRDHS1LL, FLXAnFRRDHS1LH, FLXAnFRRDHS1HL, and FLXAnFRRDHS1HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDHS1: <FLXAn_base> + 0700_H,
FLXAnFRRDHS1L: <FLXAn_base> + 0700_H, FLXAnFRRDHS1H: <FLXAn_base> + 0702_H,
FLXAnFRRDHS1LL: <FLXAn_base> + 0700_H, FLXAnFRRDHS1LH: <FLXAn_base> + 0701_H,
FLXAnFRRDHS1HL: <FLXAn_base> + 0702_H, FLXAnFRRDHS1HH: <FLXAn_base> + 0703_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.88 FLXAnFRRDHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	MBI	Message Buffer Interrupt Flag
28	TXM	Transmission Mode Flag
27	PPIT	Payload Preamble Indicator Transmit Flag
26	CFG	Message Buffer Direction Configuration Flag
25, 24	CH[1:0]	Channel Filter Control Flag
23	Reserved	When read, the value after reset is returned.
22 to 16	CYC[6:0]	Cycle Code
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	FID[10:0]	Frame ID

(1) FLXAnFRRDHS1.MBI

Message Buffer Interrupt Flag

Values as configured by the Host via FLXAnFRWRHS1.MBI.
In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(2) FLXAnFRRDHS1.TXM

Transmission Mode Flag

Values as configured by the Host via FLXAnFRWRHS1.TXM.
In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(3) FLXAnFRRDHS1.PPIT

Payload Preamble Indicator Transmit Flag

Values as configured by the Host via FLXAnFRWRHS1.PPIT.
In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(4) FLXAnFRRDHS1.CFG

Message Buffer Direction Configuration Flag

Values as configured by the Host via FLXAnFRWRHS1.CFG.
In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(5) FLXAnFRRDHS1.CH

Channel Filter Control Flag

Values as configured by the Host via FLXAnFRWRHS1.CH.
In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(6) FLXAnFRRDHS1.CYC

Cycle Code

Values as configured by the Host via FLXAnFRWRHS1.CYC.
In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(7) FLXAnFRRDHS1.FID

Frame ID

Values as configured by the Host via FLXAnFRWRHS1.FID
In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are holding the received frame ID.

25.2.11.3 FLXAnFRRDHS2 — FlexRay Read Header Section Register 2

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRRDHS2 is updated from data frames only.

Access: FLXAnFRRDHS2 is a read-only register that can be read in 32-bit units.
FLXAnFRRDHS2L and FLXAnFRRDHS2H are the read-only registers that can be read in 16-bit units.
FLXAnFRRDHS2LL, FLXAnFRRDHS2LH, FLXAnFRRDHS2HL, and FLXAnFRRDHS2HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDHS2: <FLXAn_base> + 0704$_H$,
FLXAnFRRDHS2L: <FLXAn_base> + 0704$_H$, FLXAnFRRDHS2H: <FLXAn_base> + 0706$_H$,
FLXAnFRRDHS2LL: <FLXAn_base> + 0704$_H$, FLXAnFRRDHS2LH: <FLXAn_base> + 0705$_H$,
FLXAnFRRDHS2HL: <FLXAn_base> + 0706$_H$, FLXAnFRRDHS2HH: <FLXAn_base> + 0707$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLR[6:0]							—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.89 FLXAnFRRDHS2 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 24	PLR[6:0]	Payload Length Received Flag (vRF!Header!Length)
23	Reserved	When read, the value after reset is returned.
22 to 16	PLC[6:0]	Payload Length Configured Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	CRC[10:0]	Header CRC Flag (vRF!Header!HeaderCRC)

(1) FLXAnFRRDHS2.PLR

Payload Length Received Flag (vRF!Header!Length)

Payload length (vRF!Header!Length) value updated from received data frames (exception: if message buffer belongs to the receive FIFO FLXAnFRRDSH2.PLR is also updated from received null frames).

(2) FLXAnFRRDHS2.PLC

Payload Length Configured Flag

Length of data section (number of 2-byte words) as configured by the Host.

(3) FLXAnFRRDHS2.CRC

Header CRC Flag (vRF!Header!HeaderCRC)

Receive Buffer: Header CRC (vRF!Header!HeaderCRC) updated from received data frames
Transmit Buffer: Header CRC configured by the Host

(4) Data Storage

When a message is stored into a message buffer the following behavior with respect to payload length received and payload length configured is implemented:

- FLXAnFRRDHS2.PLR > FLXAnFRRDHS2.PLC:
  The payload data stored in the message buffer is truncated to the payload length configured if FLXAnFRRDHS2.PLC even or else truncated to FLXAnFRRDHS2.PLC + 1.
- FLXAnFRRDHS2.PLR ≤ FLXAnFRRDHS2.PLC:
  The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by FLXAnFRRDHS2.PLC are filled with undefined data.
- FLXAnFRRDHS2.PLR = zero:
  The message buffer’s data section is filled with undefined data
- FLXAnFRRDHS2.PLC = zero:
  Message buffer has no data section configured. No data is stored into the message buffer’s data section.

CAUTIONS

1. The Message RAM is organized in 4-byte words. When received data is stored into a message buffer’s data section, the number of 2-byte data words written into the message buffer is FLXAnFRRDHS2.PLC rounded to the next even value.
2. FLXAnFRRDHS2.PLC should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.
   For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRRDHS2 is updated from data frames only.

25.2.11.4 FLXAnFRRDHS3 — FlexRay Read Header Section Register 3

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRRDHS3 is updated from data frames only.

Access:	FLXAnFRRDHS3 is a read-only register that can be read in 32-bit units. FLXAnFRRDHS3L and FLXAnFRRDHS3H are the read-only registers that can be read in 16-bit units. FLXAnFRRDHS3LL, FLXAnFRRDHS3LH, FLXAnFRRDHS3HL, and FLXAnFRRDHS3HH are the read-only registers that can be read in 8-bit units.
Address:	FLXAnFRRDHS3: <FLXAn_base> + 0708$_H$, FLXAnFRRDHS3L: <FLXAn_base> + 0708$_H$, FLXAnFRRDHS3H: <FLXAn_base> + 070A$_H$, FLXAnFRRDHS3LL: <FLXAn_base> + 0708$_H$, FLXAnFRRDHS3LH: <FLXAn_base> + 0709$_H$, FLXAnFRRDHS3HL: <FLXAn_base> + 070A$_H$, FLXAnFRRDHS3HH: <FLXAn_base> + 070B$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RES	PPI	NFI	SYN	SFI	RCI	—	—	RCC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.90 FLXAnFRRDHS3 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RES	Reserved Bit Indicator Flag (vRF!Header!Reserved)
28	PPI	Payload Preamble Indicator (vRF!Header!PPIndicator)
27	NFI	Null Frame Indicator Flag (vRF!Header!NFIndicator) 0: Up to now no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer
26	SYN	Sync Frame Indicator Flag (vRF!Header!SyFIndicator) 0: The received frame is not a sync frame 1: The received frame is a sync frame
25	SFI	Startup Frame Indicator Flag (vRF!Header!SuFIndicator) 0: The received frame is not a startup frame 1: The received frame is a startup frame
24	RCI	Received on Channel Indicator Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	RCC[5:0]	Receive Cycle Counter (vRF!Header!CycleCount)
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	DP[10:0]	Data Pointer Flag

(1) FLXAnFRRDHS3.RES

Reserved Bit Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

(2) FLXAnFRRDHS3.PPI

Payload Preamble Indicator Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

- 0 = The payload segment of the received frame does not contain a network management vector nor a message ID
- 1 = Static segment: Network management vector in the first part of the payload
  Dynamic segment: Message ID in the first part of the payload

(3) FLXAnFRRDHS3.NFI

Null Frame Indicator Flag (vRF!Header!NFIndicator)

Is set to 1 after storage of the first received data frame.

(4) FLXAnFRRDHS3.SYN

Sync Frame Indicator Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

(5) FLXAnFRRDHS3.SFI

Startup Frame Indicator Flag (vRF!Header!SuFIndicator)

A startup frame is marked by the startup frame indicator.

(6) FLXAnFRRDHS3.RCI

Received on Channel Indicator Flag (vSS!Channel)

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

(7) FLXAnFRRDHS3.RCC

Receive Cycle Counter (vRF!Header!CycleCount)

Cycle counter value updated from received data frame.

(8) FLXAnFRRDHS3.DP

Data Pointer Flag

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.
The bit value is the same as that set in the FLXAnFRWRHS3.DP bit.

25.2.11.5 FLXAnFRMBS — FlexRay Message Buffer Status Register

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer.

The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel- specific status flags of both channels are updated.

The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all FLXAnFRMBS flags are reset to zero independent of which FLXAnFRIBCM bits are set or not.

For details about receive/transmit filtering see **Section 25.3.8, Filtering and Masking**, **Section 25.3.9, Transmit Process** and **Section 25.3.10, Receive Process**.

Whenever the Message Handler changes one of the flags FLXAnFRMBS.VFRA, FLXAnFRMBS.VFRB, FLXAnFRMBS.SEOA, FLXAnFRMBS.SEOB, FLXAnFRMBS.CEOA, FLXAnFRMBS.CEOB, FLXAnFRMBS.SVOA, FLXAnFRMBS.SVOB, FLXAnFRMBS.TCIA, FLXAnFRMBS.TCIB, FLXAnFRMBS.ESA, FLXAnFRMBS.ESB, FLXAnFRMBS.MLST, FLXAnFRMBS.FTA, FLXAnFRMBS.FTB the respective message buffer’s MBC flag in registers FLXAnFRMBSC1/2/3/4 is set.

Access: FLXAnFRMBS is a read-only register that can be read in 32-bit units.
FLXAnFRMBSL and FLXAnFRMBSH are the read-only registers that can be read in 16-bit units.
FLXAnFRMBSLL, FLXAnFRMBSLH, FLXAnFRMBSHL, and FLXAnFRMBSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMBS: <FLXAn_base> + 070C$_H$,
FLXAnFRMBSL: <FLXAn_base> + 070C$_H$, FLXAnFRMBSH: <FLXAn_base> + 070E$_H$,
FLXAnFRMBSLL: <FLXAn_base> + 070C$_H$, FLXAnFRMBSLH: <FLXAn_base> + 070D$_H$,
FLXAnFRMBSHL: <FLXAn_base> + 070E$_H$, FLXAnFRMBSHH: <FLXAn_base> + 070F$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	—	—	CCS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.91 FLXAnFRMBS Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RESS	Reserved Bit Status Flag (vRF!Header!Reserved)
28	PPIS	Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator) 0: PPI indicator set to 0 1: PPI indicator set to 1

Table 25.91 FLXAnFRMBS Register Contents

Bit Position	Bit Name	Function
27	NFIS	Null Frame Indicator Status Flag (vRF!Header!NFIndicator) 0: Received frame is a null frame 1: Received frame is not a null frame
26	SYNS	Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator) 0: No sync frame received 1: The received frame is a sync frame
25	SFIS	Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator) 0: No startup frame received 1: The received frame is a startup frame
24	RCIS	Received on Channel Indicator Status Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	CCS[5:0]	Cycle Count Status Flag
15	FTB	Frame Transmitted on Channel B Flag 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B
14	FTA	Frame Transmitted on Channel A Flag 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A
13	Reserved	When read, the value after reset is returned.
12	MLST	Message Lost Flag 0: No message lost 1: Unprocessed message was overwritten
11	ESB	Empty Slot Channel B Flag 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B
10	ESA	Empty Slot Channel A Flag 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A
9	TCIB	Transmission Conflict Indication Channel B Flag (vSS!TxConflictB) 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B
8	TCIA	Transmission Conflict Indication Channel A Flag (vSS!TxConflictA) 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A
7	SVOB	Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB) 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B
6	SVOA	Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA) 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A
5	CEOB	Content Error Observed on Channel B Flag (vSS!ContentErrorB) 0: No content error observed on channel B 1: Content error observed on channel B
4	CEOA	Content Error Observed on Channel A Flag (vSS!ContentErrorA) 0: No content error observed on channel A 1: Content error observed on channel A
3	SEOB	Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB) 0: No syntax error observed on channel B 1: Syntax error observed on channel B

Table 25.91 FLXAnFRMBS Register Contents

Bit Position	Bit Name	Function
2	SEOA	Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA) 0: No syntax error observed on channel A 1: Syntax error observed on channel A
1	VFRB	Valid Frame Received on Channel B (vSS!ValidFrameB) 0: No valid frame received on channel B 1: Valid frame received on channel B
0	VFRA	Valid Frame Received on Channel A Flag (vSS!ValidFrameA) 0: No valid frame received on channel A 1: Valid frame received on channel A

(1) FLXAnFRMBS.RESS

Reserved Bit Status Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit.
The reserved bit is transmitted as 0.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(2) FLXAnFRMBS.PPIS

Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

- 0 = PPI indicator set to 0
  The payload segment of the received frame does not contain a network management vector or a message ID
- 1 = PPI indicator set to 1
  Static segment: Network management vector at the beginning of the payload
  Dynamic segment: Message ID at the beginning of the payload

(3) FLXAnFRMBS.NFIS

Null Frame Indicator Status Flag (vRF!Header!NFIndicator)

If set to 0 the payload segment of the received frame contains no usable data.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(4) FLXAnFRMBS.SYNS

Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(5) FLXAnFRMBS.SFIS

Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator)

The startup frame indicator specifies a startup frame.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(6) FLXAnFRMBS.RCIS

Received on Channel Indicator Status Flag (vSS!Channel)

Indicates the channel on which the frame was received.
For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(7) FLXAnFRMBS.CCS

Cycle Count Status Flag

Actual cycle count when status was updated.

(8) FLXAnFRMBS.FTB

Frame Transmitted on Channel B Flag

Indicates that this node has transmitted a data frame in the configured slot on channel B.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTB can only be reset by the Host.
Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where the bit is set to 1.

---

(9) FLXAnFRMBS.FTA

Frame Transmitted on Channel A Flag

Indicates that this node has transmitted a data frame in the configured slot on channel A.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTA can only be reset by the Host.
Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where this bit is set to 1.

---

(10) FLXAnFRMBS.MLST

Message Lost Flag

The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame.
Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset to 0 by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset to 0 by reading out the message buffer via OBF.

(11) FLXAnFRMBS.ESB

Empty Slot Channel B Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(12) FLXAnFRMBS.ESA

Empty Slot Channel A Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(13) FLXAnFRMBS.TCIB

Transmission Conflict Indication Channel B Flag (vSS!TxConflictB)

A transmission conflict indication is set to 1 if a transmission conflict has occurred on channel B.

(14) FLXAnFRMBS.TCIA

Transmission Conflict Indication Channel A Flag (vSS!TxConflictA)

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

(15) FLXAnFRMBS.SVOB

Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on the slot assigned to channel B.

(16) FLXAnFRMBS.SVOA

Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on the slot assigned to channel A.

(17) FLXAnFRMBS.CEOB

Content Error Observed on Channel B Flag (vSS!ContentErrorB)

A content error was observed in the slot assigned to channel B.

(18) FLXAnFRMBS.CEOA

Content Error Observed on Channel A Flag (vSS!ContentErrorA)

A content error was observed in the slot assigned to channel A.

(19) FLXAnFRMBS.SEOB

Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB)

A syntax error was observed in the assigned slot on channel B.

(20) FLXAnFRMBS.SEOA

Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA)

A syntax error was observed in the assigned slot on channel A.

(21) FLXAnFRMBS.VFRB

Valid Frame Received on Channel B Flag (vSS!ValidFrameB)

A valid frame indication is set if a valid frame was received on channel B.

(22) FLXAnFRMBS.VFRA

Valid Frame Received on Channel A Flag (vSS!ValidFrameA)

A valid frame indication is set if a valid frame was received on channel A.

25.2.11.6 FLXAnFROBCM — FlexRay Output Buffer Command Mask Register

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by FLXAnFROBCR.OBRS.

Mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to the register internal storage when a Message RAM transfer is requested by FLXAnFROBCR.REQ.

When OBF Host and OBF Shadow are swapped, mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 25.3.12.2(2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FLXAnFRMBSC1/2/3/4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag ND of the selected message buffer in the FLXAnFRNDAT1/2/3/4 registers is cleared.

Access: FLXAnFROBCM can be read or written in 32-bit units.
FLXAnFROBCML can be read or written in 16-bit units.
FLXAnFROBCMH is a read-only register that can be read in 16-bit units.
FLXAnFROBCMLL can be read or written in 8-bit units.
FLXAnFROBCMHL is a read-only register that can be read in 8-bit units.

Address: FLXAnFROBCM: <FLXAn_base> + 0710$_H$,
FLXAnFROBCML: <FLXAn_base> + 0710$_H$, FLXAnFROBCMH: <FLXAn_base> + 0712$_H$,
FLXAnFROBCMLL: <FLXAn_base> + 0710$_H$, FLXAnFROBCMHL: <FLXAn_base> + 0712$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSH	RHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSS	RHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.92 FLXAnFROBCM Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RDSH	Read Data Section Host Flag 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
16	RHSH	Read Header Section Host Flag 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 25.92 FLXAnFROBCM Register Contents

Bit Position	Bit Name	Function
1	RDSS	Read Data Section Shadow Bit 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
0	RHSS	Read Header Section Shadow Bit 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer

(1) FLXAnFROBCM.RDSH

Read Data Section Host Flag

(2) FLXAnFROBCM.RHSH

Read Header Section Host Flag

(3) FLXAnFROBCM.RDSS

Read Data Section Shadow Bit

(4) FLXAnFROBCM.RHSS

Read Header Section Shadow Bit

25.2.11.7 FLXAnFROBCR — FlexRay Output Buffer Command Request Register

After setting bit FLXAnFROBCR.REQ to 1 while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1, FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCM.OBSYS back to 0.

By setting bit FLXAnFROBCR.VIEW to 1 while FLXAnFROBCR.OBSYS is 0, OBF Host and OBF Shadow are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register FLXAnFROBCM internal storage to keep them attached to the respective Output Buffer transfer. FLXAnFROBCR.OBRH signals the number of the message buffer currently accessible by the Host.

If bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are set to 1 with the same write access while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Any write access to FLXAnFROBCR[15:8] while FLXAnFROBCR.OBSYS is set to 1 will cause the error flag FLXAnFREIR.IOBA to be set to 1. n this case, this write access has no effect and the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 25.3.12.2(2), Data Transfer from Message RAM to Output Buffer**.

Access:	FLXAnFROBCR can be read or written in 32-bit units. FLXAnFROBCRL can be read or written in 16-bit units. FLXAnFROBCRH is a read-only register that can be read in 16-bit units. FLXAnFROBCRLL and FLXAnFROBCRLH can be read or written in 8-bit units. FLXAnFROBCRHL is a read-only register that can be read in 8-bit units.
Address:	FLXAnFROBCR: <FLXAn_base> + 0714$_H$, FLXAnFROBCRL: <FLXAn_base> + 0714$_H$, FLXAnFROBCRH: <FLXAn_base> + 0716$_H$, FLXAnFROBCRLL: <FLXAn_base> + 0714$_H$, FLXAnFROBCRLH: <FLXAn_base> + 0715$_H$, FLXAnFROBCRHL: <FLXAn_base> + 0716$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	OBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OBSYS	—	—	—	—	—	REQ	VIEW	—	OBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.93 FLXAnFROBCR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	OBRH[6:0]	Output Buffer Request Host Flag
15	OBSYS	Output Buffer Busy Shadow Flag 0: No transfer in progress 1: Transfer between Message RAM and OBF Shadow in progress
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	REQ	Request Message RAM Transfer Bit 0: No request 1: Transfer to OBF Shadow requested
8	VIEW	View Shadow Buffer Bit 0: No action 1: Swap OBF Shadow and OBF Host
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	OBRS[6:0]	Output Buffer Request Shadow Bit

(1) FLXAnFROBCR.OBRH

Output Buffer Request Host Flag

Number of message buffer currently accessible by the Host via FLXAnFRRDHS1 to FLXAnFRRDHS3, FLXAnFRMBS, and FLXAnFRRDDS1 to FLXAnFRRDDS64.
By writing FLXAnFROBCR.VIEW to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host.

(2) FLXAnFROBCR.OBSYS

Output Buffer Busy Shadow Flag

Set to 1 after setting bit FLXAnFROBCR.REQ. When the transfer between the Message RAM and OBF Shadow has completed, FLXAnFROBCR.OBSYS is set back to 0.

(3) FLXAnFROBCR.REQ

Request Message RAM Transfer Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.
Requests transfer of message buffer addressed by FLXAnFROBCR.OBRS from Message RAM to OBF Shadow.

(4) FLXAnFROBCR.VIEW

View Shadow Buffer Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.
Toggles between OBF Shadow and OBF Host.

(5) FLXAnFROBCR.OBRS

Output Buffer Request Shadow Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.
Number of source message buffer to be transferred from the Message RAM to OBF Shadow.
If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see **Section 25.3.11, FIFO Function**) to OBF Shadow.

25.2.12 Data Transfer Control Register

25.2.12.1 FLXAnFRITC — FlexRay Input Transfer Configuration Register

Access: FLXAnFRITC can be read or written in 32-bit units.
FLXAnFRITCL and FLXAnFRITCH can be read or written in 16-bit units.
FLXAnFRITCLL, FLXAnFRITCLH, and FLXAnFRITCHL can be read or written in 8-bit units.

Address: FLXAnFRITC: <FLXAn_base> + 0800_H,
FLXAnFRITCL: <FLXAn_base> + 0800_H, FLXAnFRITCH: <FLXAn_base> + 0802_H,
FLXAnFRITCLL: <FLXAn_base> + 0800_H, FLXAnFRITCLH: <FLXAn_base> + 0801_H,
FLXAnFRITCHL: <FLXAn_base> + 0802_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	ITM[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IQEIE	IQFIE	—	—	—	—	—	—	IQHR	ITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.94 FLXAnFRITC Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	ITM[6:0]	Input queue Table Max Bit These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIE	Input Queue Empty Interrupt Enable Bit 0: Disabled 1: Enabled
8	IQFIE	Input Queue Full Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IQHR	Input Queue Halt Request Bit 0: Input queue run request 1: Input queue halt request
0	ITE	Input Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFRITC.ITM

Input queue Table Max Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.
These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.
Valid values are 00_H (1 queue entry) to $7F_H$ (128 queue entries).
Note that each entry requires two long words in the input pointer table.

(2) FLXAnFRITC.IQEIE

Input Queue Empty Interrupt Enable Bit

This bit controls the input queue empty interrupt.

- 0: Disabled
  No interrupt will be requested and the input queue empty interrupt line will be released.
- 1: Enabled
  Input queue empty interrupt will be asserted when FLXAnFRITS.IQEIS is 1.

(3) FLXAnFRITC.IQFIE

Input Queue Full Interrupt Enable Bit

This bit controls the input queue full interrupt.

- 0: Disabled
  No interrupt will be requested and the input queue full interrupt line will be released.
- 1: Enabled
  Input queue full interrupt will be asserted when FLXAnFRITS.IQFIS is 1.

(4) FLXAnFRITC.IQHR

Input Queue Halt Request Bit

The IQHR bit should not be set to 1 when FLXAnFRITS.ITS is 0.
This bit requests a halt of the input queue.
The status of the halt request is shown in the FLXAnFRITS.IQH register.
Refer to **Section 25.3.16.1(5), Halting the Input Queue** about usage of this bit.

- 0: Input queue run request
  The input queue resumes their operation.
- 1: Input queue halt request
  The input queue gets halted. An active input transfer will be completed but no further transfer request will start.

(5) FLXAnFRITC.ITE

Input Transfer Enable Bit

The user should only set this bit to 1 when FLXAnFRIBCR.IBSYS is 0.
The user should only set this bit to 0 when FLXAnFRITC.IQHR 0. Otherwise committed input transfers get lost.
This bit controls the operation mode of the input transfer queue.
The operation status of the input transfer queue function is shown in FLXAnFRITS.ITS.
Refer to **Section 25.3.16.1(1), Activation and Deactivation** about usage of this bit.

- 0: Operation Disable request
  The input transfer queue gets disabled when it becomes empty.
- 1: Operation Enable request
  The input transfer queue gets enabled. Input data structures are transferred to the FlexRay internal message RAM.

25.2.12.2 FLXAnFROTC — FlexRay Output Transfer Configuration Register

Access: FLXAnFROTC can be read or written in 32-bit units.
FLXAnFROTCL and FLXAnFROTCH can be read or written in 16-bit units.
FLXAnFROTCLL, FLXAnFROTCLH, and FLXAnFROTCHL can be read or written in 8-bit units.

Address: FLXAnFROTC: <FLXAn_base> + 0804_H,
FLXAnFROTCL: <FLXAn_base> + 0804_H, FLXAnFROTCH: <FLXAn_base> + 0806_H
FLXAnFROTCLL: <FLXAn_base> + 0804_H, FLXAnFROTCLH: <FLXAn_base> + 0805_H,
FLXAnFROTCHL: <FLXAn_base> + 0806_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	FTM[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FWIE	OWIE	FIE	OIE	—	—	—	—	—	—	OTCS	OTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 25.95 FLXAnFROTC Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FTM[4:0]	FIFO Table Max Bit Configures the number of FIFO entries the output transfer handler is capable to maintain in the local RAM/global RAM.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIE	FIFO transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
10	OWIE	Output transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
9	FIE	FIFO transfer Interrupt Enable Bit 0: Disabled 1: Enabled
8	OIE	Output transfer Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OTCS	Output Transfer Condition Select Bit 0: New data only mode 1: New data and status changed mode
0	OTE	Output Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFROTC.FTM

FIFO Table Max Bit

The user can only write to these bits when FLXAnFROTS.OTS is 0.
Configures the number of FIFO entries the output transfer handler is capable to maintain in the local RAM/global RAM.
Valid values are 00_H (1 FIFO entry) to $1F_H$ (32 FIFO entries).

(2) FLXAnFROTC.FWIE

FIFO transfer Warning Interrupt Enable Bit

This bit controls the FIFO transfer warning interrupt.

- 0: Disabled
  No interrupt will be requested and the FIFO transfer warning interrupt line will be released.
- 1: Enabled
  FIFO transfer warning interrupt will be asserted when FLXAnFROTS.FWIS is 1.

(3) FLXAnFROTC.OWIE

Output transfer Warning Interrupt Enable Bit

This bit controls the output transfer warning interrupt.

- 0: Disabled
  No interrupt will be requested and the output transfer warning interrupt line will be released.
- 1: Enabled
  Output transfer warning interrupt will be asserted when FLXAnFROTS.OWIS is 1.

(4) FLXAnFROTC.FIE

FIFO transfer Interrupt Enable Bit

This bit controls the FIFO transfer interrupt.

- 0: Disabled
  No interrupt will be requested and the FIFO transfer interrupt line will be released.
- 1: Enabled
  FIFO transfer interrupt will be asserted when FLXAnFROTS.FIS is 1.

(5) FLXAnFROTC.OIE

Output transfer Interrupt Enable Bit

This bit controls the output transfer interrupt.

- 0: Disabled
  No interrupt will be requested and the output transfer interrupt line will be released.
- 1: Enabled
  Output transfer interrupt will be asserted when FLXAnFROTS.OTIS is 1.

(6) FLXAnFROTC.OTCS

Output Transfer Condition Select Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0. This bit controls the output transfer condition.

- 0: New data only mode
  The ND bits in the FLXAnFRNDATi registers are used to detect a transfer condition for dedicated receive buffer
- 1: New data and status changed mode
  The ND bits in the FLXAnFRNDATi registers and the MBC bits in the FLXAnFRMBSCi register are used to detect a transfer condition for dedicated transmit and receive buffer

(7) FLXAnFROTC.OTE

Output Transfer Enable Bit

The user should only set this bit to 1 when FLXAnFROBCR.OBSYS is 0.
This bit controls the operation mode of the output transfer function.
The operation status of the output buffer transfer function is shown in FLXAnFROTS.OTS.
Refer to **Section 25.3.16.2(2), Output Transfer Data Structure** about usage of this bit.

- 0: Operation Disable request
  The output buffer transfer gets disabled.
  An active message buffer transfer will be completed but no further transfer will start.
- 1: Operation Enable request
  The output buffer transfer gets enabled. Message buffers are transferred from the FlexRay internal message RAM to output data structures.

The user cannot change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

25.2.12.3 FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register

Access:	FLXAnFRIBA can be read or written in 32-bit units. FLXAnFRIBAL and FLXAnFRIBAH can be read or written in 16-bit units. FLXAnFRIBALL, FLXAnFRIBALH, FLXAnFRIBAHL, and FLXAnFRIBAHH can be read or written in 8-bit units.
Address:	FLXAnFRIBA: <FLXAn_base> + 0808_H, FLXAnFRIBAL: <FLXAn_base> + 0808_H, FLXAnFRIBAH: <FLXAn_base> + $080A_H$, FLXAnFRIBALL: <FLXAn_base> + 0808_H, FLXAnFRIBALH: <FLXAn_base> + 0809_H, FLXAnFRIBAHL: <FLXAn_base> + $080A_H$, FLXAnFRIBAHH: <FLXAn_base> + $080B_H$
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ITA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.96 FLXAnFRIBA Register Contents

Bit Position	Bit Name	Function
31 to 0	ITA[31:0]	Input Table Address Bit These bits configure the base address of the input pointer table.

(1) FLXAnFRIBA.ITA

Input Table Address Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.
The address should be 32 bit aligned, thus the bits FLXAnFRIBA.ITA[1:0] are always 0.
These bits configure the base address of the input pointer table.
The table is used for the input transfer queue transferring message buffers from the local RAM/global RAM into the FlexRay internal message RAM.
The size of the input queue is configured in FLXAnFRITC.ITM.
Note that each entry requires two long words in the input pointer table.

25.2.12.4 FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register

Access: FLXAnFRFBA can be read or written in 32-bit units.
FLXAnFRFBAL and FLXAnFRFBAH can be read or written in 16-bit units.
FLXAnFRFBALL, FLXAnFRFBALH, FLXAnFRFBAHL, and FLXAnFRFBAHH can be read or written in 8-bit units.

Address: FLXAnFRFBA: <FLXAn_base> + 080C$_H$,
FLXAnFRFBAL: <FLXAn_base> + 080C$_H$, FLXAnFRFBAH: <FLXAn_base> + 080E$_H$,
FLXAnFRFBALL: <FLXAn_base> + 080C$_H$, FLXAnFRFBALH: <FLXAn_base> + 080D$_H$,
FLXAnFRFBAHL: <FLXAn_base> + 080E$_H$, FLXAnFRFBAHH: <FLXAn_base> + 080F$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.97 FLXAnFRFBA Register Contents

Bit Position	Bit Name	Function
31 to 0	FTA[31:0]	FIFO pointer Table Address Bit These bits configure the base address of the FIFO pointer table.

(1) FLXAnFRFBA.FTA

FIFO pointer Table Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.
The address should be 32 bit aligned, thus the bits FLXAnFRFBA.FTA[1:0] are always 0.
These bits configure the base address of the FIFO pointer table.
The table is used for message buffers transferred from the FlexRay internal FIFO to the local RAM/global RAM.
The size of the FIFO is configured in FLXAnFROTC.FTM.

25.2.12.5 FLXAnFROBA — FlexRay Output Pointer Table Base Address Register

Access: FLXAnFROBA can be read or written in 32-bit units.
FLXAnFROBAL and FLXAnFROBAH can be read or written in 16-bit units.
FLXAnFROBALL, FLXAnFROBALH, FLXAnFROBAHL, and FLXAnFROBAHH can be read or written in 8-bit units.

Address: FLXAnFROBA: <FLXAn_base> + 0810_H,
FLXAnFROBAL: <FLXAn_base> + 0810_H, FLXAnFROBAH: <FLXAn_base> + 0812_H,
FLXAnFROBALL: <FLXAn_base> + 0810_H, FLXAnFROBALH: <FLXAn_base> + 0811_H,
FLXAnFROBAHL: <FLXAn_base> + 0812_H, FLXAnFROBAHH: <FLXAn_base> + 0813_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.98 FLXAnFROBA Register Contents

Bit Position	Bit Name	Function
31 to 0	OTA[31:0]	Output pointer Table Address Bit These bits configure the base address of the output pointer table.

(1) FLXAnFROBA.OTA

Output pointer Table Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.
The address should be 32 bit aligned, thus the bits FLXAnFROBA.OTA[1:0] are always 0.
These bits configure the base address of the output pointer table.
The table is used for message buffers transferred from the FlexRay internal message RAM to the local RAM/global RAM.
The size of the table depends on the utilization of the FlexRay internal message RAM and can have up to 128 entries.

25.2.12.6 FLXAnFRIQC — FlexRay Input Queue Control Register

Access: FLXAnFRIQC is a write-only register that can be written in 32-bit units.
FLXAnFRIQCL is a write-only register that can be written in 16-bit units.
FLXAnFRIQCLL is a write-only register that can be written in 8-bit units.

Address: FLXAnFRIQC: <FLXAn_base> + 0814_H,
FLXAnFRIQCL: <FLXAn_base> + 0814_H,
FLXAnFRIQCLL: <FLXAn_base> + 0814_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W

Table 25.99 FLXAnFRIQC Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When writing, write the value after reset.
6 to 0	IMBNR[6:0]	Input Message Buffer NumbeR Bit Message buffer number added to the input queue

(1) FLXAnFRIQC.IMBNR

Input Message Buffer NumbeR Bit

The user can only write to this bit when FLXAnFRITS.IQFP is 0.
The user cannot write to this register when FLXAnFRITS.ITS is 0 or when FLXAnFRITC.ITE is 0.
These bits are read as 0.
This value specifies the message buffer added to the input queue.
The number has to be identical to FLXAnFRWRHS4.IMBNR (see **Section 25.3.16.1(3), Input Pointer Table**) of the input pointer table.
The address to the input data structure has to be provided in the input pointer table at the put index (FLXAnFRITS.IPIDX) before writing to this register.
Writing to this register increments the input put index (FLXAnFRITS.IPIDX).

25.2.12.7 FLXAnFRUIR — FlexRay User Input Transfer Request Register

Access:	FLXAnFRUIR can be read or written in 32-bit units. FLXAnFRUIRL can be read or written in 16-bit units. FLXAnFRUIRLL can be read or written in 8-bit units.
Address:	FLXAnFRUIR: <FLXAn_base> + 0818_H, FLXAnFRUIRL: <FLXAn_base> + 0818_H, FLXAnFRUIRLL: <FLXAn_base> + 0818_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.100 FLXAnFRUIR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	UIDX[7:0]	User requested Input index Bit Input pointer table index requested for input transfer

(1) FLXAnFRUIR.UIDX

User requested Input index Bit

The user can only write to this bit when FLXAnFRITS.UIRP is 0.
The user should not write to this register when FLXAnFRITS.ITS is 0.
The user should not write to this register when FLXAnFRITS.UIRP is 1.
The user should not write to this register when FLXAnFRITS.IQH is 1.
The user should only write FLXAnFRITC.ITM + 1 to this register.
This value specifies the input pointer table index for the user requested input transfer.
The address to the input data structure has to be provided in the input pointer table at the index UIDX before writing to this register.
When writing to this register, the requested input data structure will be transferred from input data structure position to the FlexRay internal message RAM.
In opposite to queued input transfers the related DA flag in the FLXAnFRDA register is not influenced by the user input transfer.

25.2.12.8 FLXAnFRUOR — FlexRay User Output Transfer Request Register

Access: FLXAnFRUOR can be read or written in 32-bit units.
FLXAnFRUORL can be read or written in 16-bit units.
FLXAnFRUORLL and FLXAnFRUORLH can be read or written in 8-bit units.

Address: FLXAnFRUOR: <FLXAn_base> + 081C$_H$,
FLXAnFRUORL: <FLXAn_base> + 081C$_H$,
FLXAnFRUORLL: <FLXAn_base> + 081C$_H$, FLXAnFRUORLH: <FLXAn_base> + 081D$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	URDS	—	—	UMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.101 FLXAnFRUOR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	URDS	User request Read Data Section Bit 0: Data section is not transferred 1: Data section is transferred
8, 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	UMBNR[6:0]	User requested output Message Buffer NumbeR Bit Message buffer number requested for output transfer

(1) FLXAnFRUOR.URDS

User request Read Data Section Bit

The user can only write to this bit when FLXAnFROTS.UORP is 0.
The user should not write to this register when FLXAnFROTS.OTS is 0.
The user should not write to this register when FLXAnFROTS.UORP is 1.

- 0: Data section is not transferred
  The data section of the message buffer selected by the bits UMBNR is not requested
- 1: Data section is transferred
  The data section of the message buffer selected by the bits UMBNR is requested

(2) FLXAnFRUOR.UMBNR

User requested output Message Buffer NumbeR Bit

The user can only write to this bit when FLXAnFROTS.UORP is 0.
The user should not write to this register when FLXAnFROTS.OTS is 0.
The user should not write to this register when FLXAnFROTS.UORP is 1.
The user should restrict UMBNR to dedicated receive and transmit buffers when the FlexRay module is not in the CONFIG state.
When writing to this register, the header sections and optionally the data section (configurable by URDS) of the requested message buffer will be transferred from the FlexRay internal message RAM to the output data structure position defined by the output structure data pointer in the output pointer table.

25.2.12.9 FLXAnFRAHBC — FlexRay H-Bus Configuration Register

In this product, this register is not used because FLXAnFRAHBC.HPROT[3:0] has no function.

Access:	FLXAnFRAHBC can be read or written in 32-bit units. FLXAnFRAHBCL can be read or written in 16-bit units. FLXAnFRAHBCLL can be read or written in 8-bit units.
Address:	FLXAnFRAHBC: <FLXAn_base> + 0840_H, FLXAnFRAHBCL: <FLXAn_base> + 0840_H, FLXAnFRAHBCLL: <FLXAn_base> + 0840_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	HPROT[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.102 FLXAnFRAHBC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	HPROT[3:0]	No function

25.2.13 Data Transfer Status Register

25.2.13.1 FLXAnFRITS — FlexRay Input Transfer Status Register

Access:	FLXAnFRITS can be read or written in 32-bit units. FLXAnFRITSL can be read or written in 16-bit units. FLXAnFRITSH is a read-only register that can be read in 16-bit units. FLXAnFRITSLL, FLXAnFRITSHL, and FLXAnFRITSHH are the read-only registers that can be read in 8-bit units. FLXAnFRITSLH can be read or written in 8-bit units.
Address:	FLXAnFRITS: <FLXAn_base> + 0820_H, FLXAnFRITSL: <FLXAn_base> + 0820_H, FLXAnFRITSH: <FLXAn_base> + 0822_H, FLXAnFRITSLL: <FLXAn_base> + 0820_H, FLXAnFRITSLH: <FLXAn_base> + 0821_H, FLXAnFRITSHL: <FLXAn_base> + 0822_H, FLXAnFRITSHH: <FLXAn_base> + 0823_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IGIDX[6:0]							—	IPIDX[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IQFP	—	—	IQEIS	IQFIS	—	—	—	—	—	UIRP	IQH	ITS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 25.103 FLXAnFRITS Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	IGIDX[6:0]	Input queue Get Index Bit Represents the get index of the input pointer table
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IPIDX[6:0]	Input queue Put Index Bit Represents the put index of the input pointer table
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	IQFP	Input Queue Full condition Pending Bit 0: Entries in the input queue are available 1: All entries in the input queue are occupied
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIS	Input Queue Empty Interrupt Status Bit 0: No input queue empty condition detected 1: Input queue empty condition detected
8	IQFIS	Input Queue Full Interrupt Status Bit 0: No input queue full condition detected 1: Input queue full condition detected
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	UIRP	User Input transfer Request Pending Bit 0: No user input transfer request pending 1: User input transfer request pending

Table 25.103 FLXAnFRITS Register Contents

Bit Position	Bit Name	Function
1	IQH	Input Queue Halted Bit 0: Input queue not halted 1: Input queue halted
0	ITS	Input Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFRITS.IGIDX

Input queue Get Index Bit

These bits are only valid when FLXAnFRITS.IQH is 1.
These bits indicate the input pointer index the input queue handler will transfer next.
Valid values are 00_H to FLXAnFRITC.ITM.
The get index is incremented when the input data structure has been transferred from the local RAM/global RAM and the related DA flag in the FLXAnFRDA register is cleared.
The index is set to 00_H when FLXAnFRITS.ITS changes from 0 to 1.

(2) FLXAnFRITS.IPIDX

Input queue Put Index Bit

These bits indicate the index where the next input data structure pointer in the input pointer table should be stored.
Valid values are 00_H to FLXAnFRITC.ITM.
After reaching the maximum value the put index continues from 00_H.
The index is incremented when writing to FLXAnFRIQC.IMBNR.
The index is set to 00_H when FLXAnFRITS.ITS changes from 0 to 1.

(3) FLXAnFRITS.IQFP

Input Queue Full condition Pending Bit

This bit indicates that the input queue is full.
There should be no further input transfer requests, by writing to FLXAnFRIQC.IMBNR, as long as FLXAnFRITS.IQFP is 1.

[Clearing condition]
This bit is cleared when there is one free entry in the input queue.

[Setting condition]
This bit is set when all entries in the input queue are occupied.

(4) FLXAnFRITS.IQEIS

Input Queue Empty Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFRITC.IQEIE the input queue empty interrupt is generated when FLXAnFRITS.IQEIS is 1.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRITS.IQEIS.
This bit is cleared when FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]
This bit is set when all pending input transfers have been processed and consequently the input queue becomes empty.

(5) FLXAnFRITS.IQFIS

Input Queue Full Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFRITC.IQFIE the input queue full interrupt is generated when FLXAnFRITS.IQFIS is 1.
This flag is intended as interrupt status flag. It does not indicate the current input queue status; for this status refer to FLXAnFRITS.IQFP.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRITS.IQFIS.
This bit is cleared when FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]
This bit is set when all entries in the input queue are occupied.

(6) FLXAnFRITS.UIRP

User Input transfer Request Pending Bit

This bit indicates that a user input transfer is still pending.
There should be no further write access to FLXAnFRUIR.UIDX when this bit is 1.

[Clearing condition]
This bit is cleared when the user input transfer request is processed by the input transfer handler.

[Setting condition]
This bit is set when writing to FLXAnFRUIR.UIDX.

(7) FLXAnFRITS.IQH

Input Queue Halted Bit

This bit indicates the status of the input queue.
There should be no further write access to FLXAnFRUIR.UIDX when this bit is 1.

[Clearing condition]
This bit is cleared when FLXAnFRITC.IQHR is set to 0.

[Setting condition]
This bit is set immediately when the FLXAnFRITC.IQHR is set to 1 and there is no ongoing input transfer.
This bit is set only after an ongoing input transfer has been completed and FLXAnFRITC.IQHR is set to 1.

(8) FLXAnFRITS.ITS

Input Transfer Status Bit

This bit indicates the status of the input queue handler.
While this bit is 1, there can be no read or write access to the address area <FLXAn_base> + 0400_H to <FLXAn_base> + $05FF_H$ and there should be no CLEAR_RAMS command applied to FLXAnFRSUCC1.CMD register.
The input transfer queue indices and related status flags are set to 0 when FLXAnFRITS.ITS changes from 0 to 1.

[Clearing condition]
This bit is cleared immediately when FLXAnFRITC.ITE is set to 0 and there are no pending input transfers.
This bit is cleared after all pending requests have been processed and FLXAnFRITC.ITE is 0.

[Setting condition]
This bit is set when FLXAnFRITC.ITE is set to 1.

25.2.13.2 FLXAnFROTS — FlexRay Output Transfer Status Register

Access: FLXAnFROTS can be read or written in 32-bit units.
FLXAnFROTSL can be read or written in 16-bit units.
FLXAnFROTSH is a read-only register that can be read in 16-bit units.
FLXAnFROTSLL, FLXAnFROTSHL, and FLXAnFROTSHH are the read-only registers that can be read in 8- bit units.
FLXAnFROTSLH can be read or written in 8-bit units.

Address: FLXAnFROTS: <FLXAn_base> + 0824_H,
FLXAnFROTSL: <FLXAn_base> + 0824_H, FLXAnFROTSH: <FLXAn_base> + 0826_H,
FLXAnFROTSLL: <FLXAn_base> + 0824_H, FLXAnFROTSLH: <FLXAn_base> + 0825_H,
FLXAnFROTSHL: <FLXAn_base> + 0826_H, FLXAnFROTSHH: <FLXAn_base> + 0827_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	FFL[5:0]						—	—	—	FGIDX[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWP	OWP	FDA	—	FWIS	OWIS	FIS	OTIS	—	—	—	—	—	UORP	—	OTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 25.104 FLXAnFROTS Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	FFL[5:0]	FIFO Fill Level Bit Represent the number of unprocessed output FIFO structures
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FGIDX[4:0]	FIFO Get Index Bit Represent the get index in the FIFO pointer table
15	FWP	FIFO transfer Warning condition Pending Bit 0: No FIFO transfer warning condition pending 1: FIFO transfer warning condition pending
14	OWP	Output transfer Warning condition Pending 0: No output transfer warning condition pending 1: Output transfer warning condition pending
13	FDA	FIFO Data Available Bit 0: No available FIFO structures 1: FIFO structures available at current FLXAnFROTS.FGIDX index
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIS	FIFO transfer Warning Interrupt Status Bit 0: No FIFO transfer warning condition detected 1: FIFO transfer warning condition detected
10	OWIS	Output transfer Warning Interrupt Status Bit 0: No output transfer warning condition detected 1: Output transfer warning condition detected
9	FIS	FIFO transfer Interrupt Status Bit 0: No FIFO structure updated in local RAM/global RAM 1: FIFO structure updated in local RAM/global RAM

Table 25.104 FLXAnFROTS Register Contents

Bit Position	Bit Name	Function
8	OTIS	Output transfer Interrupt Status Bit 0: No output structure updated in local RAM/global RAM 1: Output structure updated in local RAM/global RAM
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	UORP	User Output transfer Request Pending Bit 0: No user output transfer request pending 1: User output transfer request pending
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OTS	Output Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFROTS.FFL

FIFO Fill Level Bit

These bits indicate the number of available output FIFO structures in the local RAM/global RAM. Valid values are 00_H to FLXAnFROTC.FTM + 1.
The value 00_H indicates that the FIFO is empty.
The value FLXAnFROTC.FTM + 1 indicates that the FIFO is full and no further FIFO transfers will be done.
The FIFO fill level is incremented when a FIFO data structure has been transferred from the FlexRay internal FIFO into the local RAM/global RAM.
The FIFO fill level is decremented when the user releases a FIFO data structure in the local RAM/global RAM by writing 1 to FLXAnFROTS.FDA.
The FIFO fill level is set to 00_H when the bit FLXAnFROTS.OTS changes from 0 to 1.

(2) FLXAnFROTS.FGIDX

FIFO Get Index Bit

These bits indicate the index where the current output data structure pointer in the FIFO pointer table is available for reading.
Valid values are 00_H to FLXAnFROTC.FTM.
After reaching the maximum value the get index continues from 00_H.
The index is incremented when a FIFO data structure is released by writing 1 to FLXAnFROTS.FDA.
The index is set to 00_H when FLXAnFROTS.OTS changes from 0 to 1.

(3) FLXAnFROTS.FWP

FIFO transfer Warning condition Pending Bit

This bit indicates the FIFO transfer warning condition.

[Clearing condition]
This bit is cleared when there are free output data structures (FLXAnFROTS.FFL ≤ FLXAnFROTC.FTM).
This bit is cleared when the FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1).

(4) FLXAnFROTS.OWP

Output transfer Warning condition Pending Bit

This bit indicates the output transfer warning condition.

[Clearing condition]
This bit is cleared, when all output structure pointers that have a pending output handler transfer condition detected are released (for dedicated transmit and receive message buffers or a user output transfer request).

[Setting condition]
This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(5) FLXAnFROTS.FDA

FIFO Data Available Bit

Writing 0 has no effect on the bit value.
When this bit is 1, the next valid output data structure is available.
The related data structure pointer is in the FIFO pointer table at FLXAnFROTS.FGIDX.
Writing 1 to FLXAnFROTS.FDA

- increments FLXAnFROTS.FGIDX and
- decrements the FIFO fill level (FLXAnFROTS.FFL)

If there are still unprocessed data structures FLXAnFROTS.FDA remains 1.

[Clearing condition]
This bit is cleared when writing 1 to FLXAnFROTS.FDA and the FIFO fill level becomes 00_H.
This bit is cleared when the FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when there is at least one FIFO data structure available in the local RAM/global RAM.

(6) FLXAnFROTS.FWIS

FIFO transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFROTC.FWIE the FIFO transfer warning interrupt is generated when FLXAnFROTS.FWIS is 1.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFROTS.FWIS.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1).

(7) FLXAnFROTS.OWIS

Output transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFROTC.OWIE the FIFO transfer warning interrupt is generated when FLXAnFROTS.OWIS is 1.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFROTS.OWIS.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(8) FLXAnFROTS.FIS

FIFO transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFROTC.FIE the FIFO transfer interrupt is generated when FLXAnFROTS.FIS is 1.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFROTS.FIS.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when a FIFO data structure is updated by the transfer handler or the FFL bit changes from 00_H to 01_H.

(9) FLXAnFROTS.OTIS

Output transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.
If enabled in FLXAnFROTC.OIE the output transfer interrupt is generated when FLXAnFROTS.OTIS is 1.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFROTS.OTIS.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when an output data structure is updated by the transfer handler (from a dedicated transmit or receive message buffer or by a user output transfer request).

(10) FLXAnFROTS.UORP

User Output transfer Request Pending Bit

This bit indicates that a user output transfer is still pending.
There should be no further write access to FLXAnFRUOR.UMBNR when this bit is 1.

[Clearing condition]
This bit is cleared when the user output transfer request is processed by the output transfer handler.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]
This bit is set when writing to FLXAnFRUOR.UMBNR.

(11) FLXAnFROTS.OTS

Output Transfer Status Bit

This bit indicates the status of the output transfer handler.
While this bit is 1, there should be no read or write access to the address area <FLXAn_base> + 0600_H to <FLXAn_base> + $07FF_H$ and there should be no CLEAR_RAMS command applied to FLXAnFRSUCC1.CMD register.
While this bit is 1, the user cannot change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.
The output hander transfer indices and related status flags are set to 0 when FLXAnFROTS.OTS changes from 0 to 1.

[Clearing condition]
This bit is cleared immediately when FLXAnFROTC.OTE is set to 0 and there are no ongoing output transfers.
This bit is cleared after an ongoing transfer has been completed and FLXAnFROTC.OTE is 0.

[Setting condition]
This bit is set when FLXAnFROTC.OTE is set to 1.

25.2.13.3 FLXAnFRAES — FlexRay Access Error Status Register

Access: FLXAnFRAES can be read or written in 32-bit units.
FLXAnFRAESL can be read or written in 16-bit units.
FLXAnFRAESLL is a read-only register that can be read in 8-bit units.
FLXAnFRAESLH can be read or written in 8-bit units.

Address: FLXAnFRAES: <FLXAn_base> + 0828$_{H}$,
FLXAnFRAESL: <FLXAn_base> + 0828$_{H}$,
FLXAnFRAESLL: <FLXAn_base> + 0828$_{H}$, FLXAnFRAESLH: <FLXAn_base> + 0829$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MAE	FAE	OAE	IAE	EIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 25.105 FLXAnFRAES Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MAE	Multiple Access Errors Bit 0: No multiple access errors occurred 1: Multiple access errors occurred
10	FAE	FIFO transfer Access Error Bit 0: No access error occurred during FIFO transfer 1: Access error occurred during FIFO transfer
9	OAE	Output transfer Access Error Bit 0: No access error occurred during output transfer 1: Access error occurred during output transfer
8	IAE	Input transfer Access Error Bit 0: No access error occurred during input transfer 1: Access error occurred during input transfer
7 to 0	EIDX[7:0]	Error Index Bit Data structure pointer index number

(1) FLXAnFRAES.MAE

Multiple Access Errors Bit

Writing 0 has no effect on the bit value.
This bit indicates that there were multiple access errors during a data transfer.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRAES.MAE.

[Setting condition]
This bit is set when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE are set and

- an access to an protected address occurred during a FIFO data transfer or
- an access to an protected address occurred during an output data transfer or
- an access to an protected address occurred during an input data transfer

(2) FLXAnFRAES.FAE

FIFO transfer Access Error Bit

Writing 0 has no effect on the bit value.
This bit indicates that there was an access error during a FIFO data transfer.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRAES.FAE.

[Setting condition]
This bit is set when a local RAM/global RAM access error was detected during a FIFO transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(3) FLXAnFRAES.OAE

Output transfer Access Error Bit

Writing 0 has no effect on the bit value.
This bit indicates that there was an access error during a output data transfer.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRAES.OAE.

[Setting condition]
This bit is set when a local RAM/global RAM access error was detected during an output transfer and the bits FLXAnFRAES.FAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(4) FLXAnFRAES.IAE

Input transfer Access Error Bit

Writing 0 has no effect on the bit value.
This bit indicates that there was an access error during an input data transfer.

[Clearing condition]
This bit is cleared when writing a 1 to FLXAnFRAES.IAE.

[Setting condition]
This bit is set when a local RAM/global RAM access error was detected during an output transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.FAE and FLXAnFRAES.MAE are 0.

(5) FLXAnFRAES.EIDX

Error Index Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is 1.
When the bit FLXAnFRAES.FAE is 1, FLXAnFRAES.EIDX holds the used FIFO put index when the access error has occurred.
When the bit FLXAnFRAES.OAE is 1, FLXAnFRAES.EIDX holds the input pointer table get index used when an access error occurred during an input transfer or when the user request an input transfer.
When the bit FLXAnFRAES.IAE is 1, FLXAnFRAES.EIDX holds the used input pointer table get index when the access error has occurred.
These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is changed from 0 to 1.

25.2.13.4 FLXAnFRAEA — FlexRay Access Error Address Register

Access: FLXAnFRAEA is a read-only register that can be read in 32-bit units.
FLXAnFRAEAL and FLXAnFRAEAH are the read-only registers that can be read in 16-bit units.
FLXAnFRAEALL, FLXAnFRAEALH, FLXAnFRAEAHL, and FLXAnFRAEAHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRAEA: <FLXAn_base> + $082C_H$,
FLXAnFRAEAL: <FLXAn_base> + $082C_H$, FLXAnFRAEAH: <FLXAn_base> + $082E_H$,
FLXAnFRAEALL: <FLXAn_base> + $082C_H$, FLXAnFRAEALH: <FLXAn_base> + $082D_H$,
FLXAnFRAEAHL: <FLXAn_base> + $082E_H$, FLXAnFRAEAHH: <FLXAn_base> + $082F_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AEA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.106 FLXAnFRAEA Register Contents

Bit Position	Bit Name	Function
31 to 0	AEA[31:0]	Access Error Address Bit Address in the local RAM/global RAM when an access error has occurred

(1) FLXAnFRAEA.AEA

Access Error Address Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is 1.
These bits indicate the address of the access error indicated in the FLXAnFRAES register.
These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is changed from 0 to 1.

25.2.13.5 FLXAnFRDAi — FlexRay Message Data Available Register i (i = 0 to 3)

Access: FLXAnFRDAi can be read or written in 32-bit units.
FLXAnFRDAiL and FLXAnFRDAiH can be read or written in 16-bit units.
FLXAnFRDAiLL, FLXAnFRDAiLH, FLXAnFRDAiHL, and FLXAnFRDAiHH can be read or written in 8-bit units.

Address: FLXAnFRDAx: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxL: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxH: <FLXAn_base> + 0830_H + i × 4_H + 2_H,
FLXAnFRDAxLL: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxLH: <FLXAn_base> + 0830_H + i × 4_H + 1_H,
FLXAnFRDAxHL: <FLXAn_base> + 0830_H + i × 4_H + 2_H,
FLXAnFRDAxHH: <FLXAn_base> + 0830_H + i × 4_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAb (b = i × 32 + 31 to i × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAb (b = i × 32 + 15 to i × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.107 FLXAnFRDAi Register Contents

Bit Position	Bit Name	Function
31 to 0	DAb	Data Available Bit 0: No data available for destination 1: Data available for destination

(1) FLXAnFRDAi.DAb (b = i × 32 to (i + 1) × 32 – 1)

Data Available Bit b

The user should not write a 1 to bits that are 0.
To maintain the status of input transfers, the user cannot clear bits related to input transfers.
This register is used for input and output transfers.
Each flag corresponds to a FlexRay message buffer.

[Clearing condition]
Input transfer:
This bit is cleared when the input data structure has been transferred from the local RAM/global RAM. The data structure and the data structure pointer can be changed when the related flag is 0.

Output transfer:
This bit is cleared by writing 1 to it.

[Setting condition]
Input transfer:
This bit is set when the corresponding message buffer number has been written to FLXAnFRIQC.IMBNR.
As long as this bit is 1, the input data structure and the data structure pointer corresponding to this input transfer request cannot be changed.

Output transfer:
This bit is set when the output data structure corresponding to this message buffer has been updated. As long as this bit is 1, the data structure is stable; no further update of the data structure will be done by the output handler. While this bit is 1, the application is allowed to change the output data structure pointer in the output pointer table for this message buffer number.

25.3 Functional Description

This chapter describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification.

25.3.1 FlexRay Module Operation Control

25.3.1.1 FlexRay Module Enable

After hardware reset or after the FlexRay module has been disabled (following **Section 25.3.1.2, FlexRay Module Disable**) the FlexRay module is in the reset state (FLXAnFROS.OS is 0) and the clocks of the FlexRay core module are disabled.

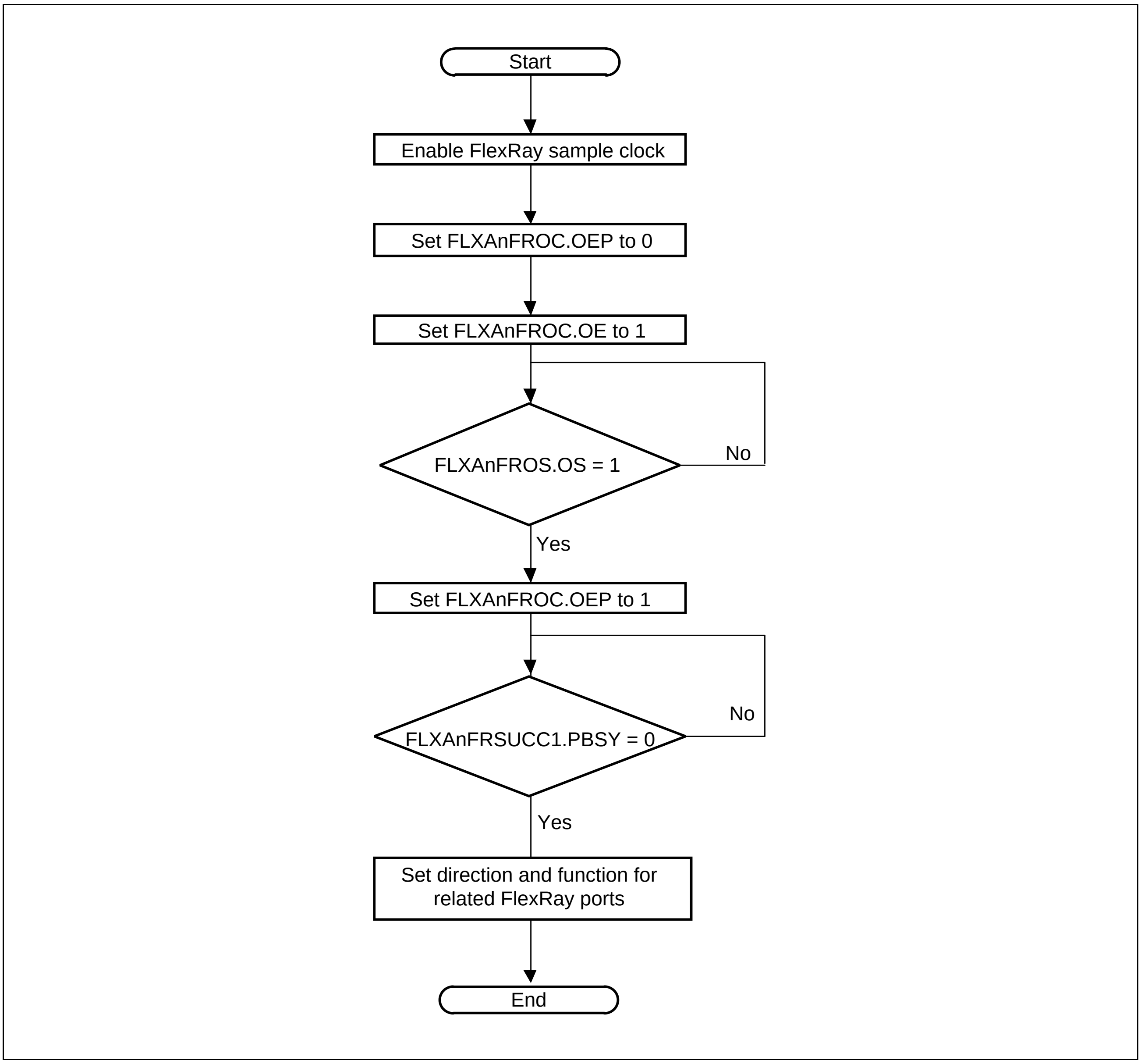


Figure 25.3 FlexRay Enable Flow

25.3.1.2 FlexRay Module Disable

The FlexRay module can be disabled at any time. However, it is recommended to disable the FlexRay module using the FLXAnFROC.OE register only when the FlexRay module is in HALT, CONFIG or DEFAULT_CONFIG state. Resetting the FlexRay module in any other state will terminate any ongoing FlexRay communication.

If the data transfer function is used, it is also required to disable this function before disabling the FlexRay module (see **Section 25.3.16.1(1), Activation and Deactivation** for suspending input transfer function and **Section 25.3.16.2(2), Output Transfer Data Structure** for suspending output transfer).

The following flow should be executed to disable the FlexRay module.

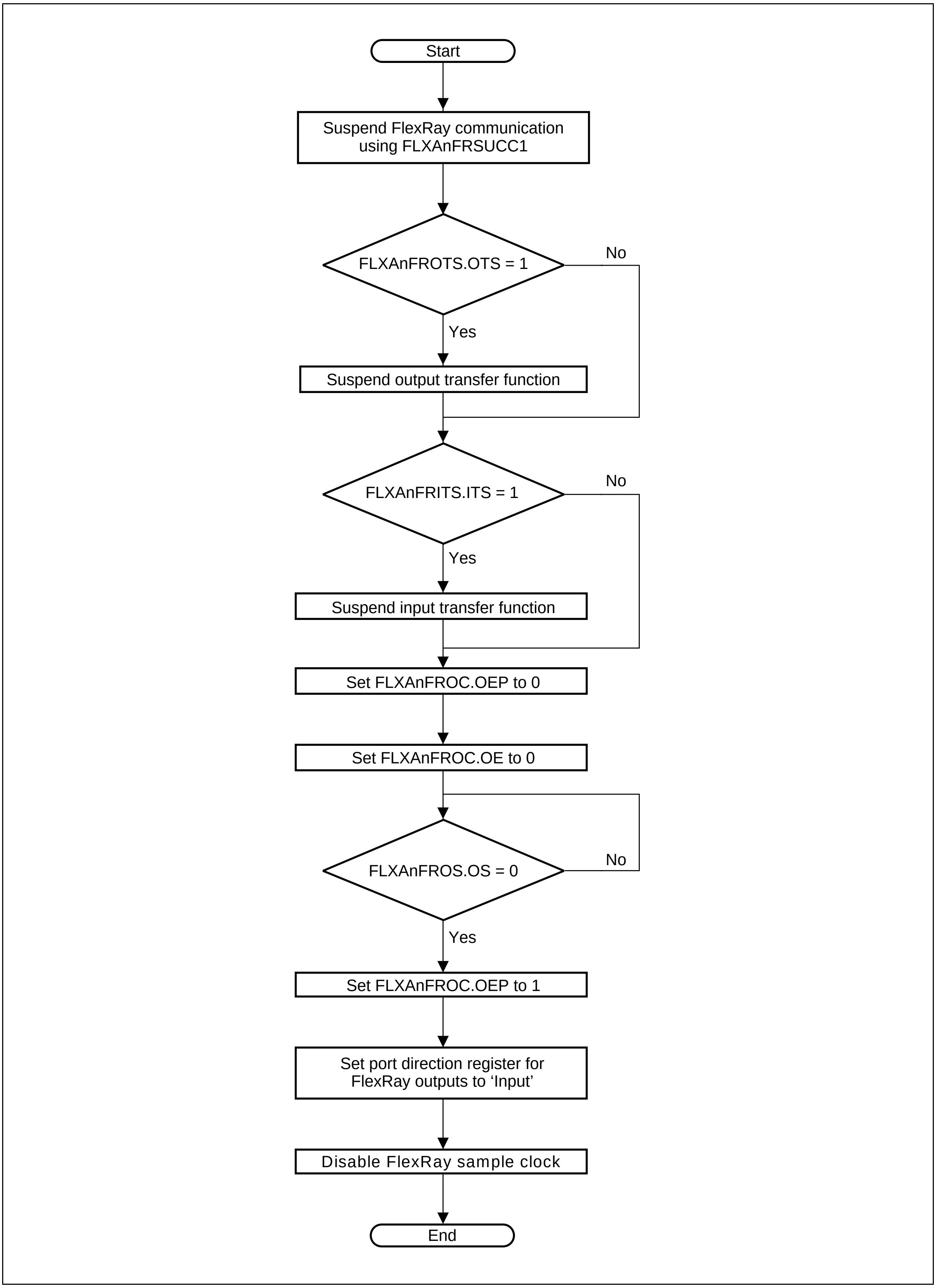


Figure 25.4 FlexRay Disable Flow

25.3.2 Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

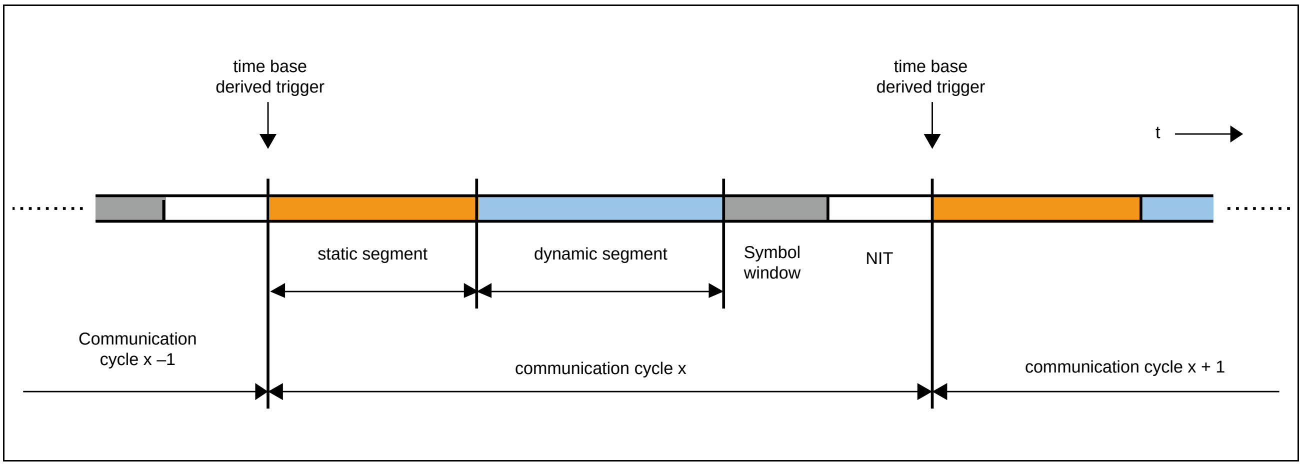


Figure 25.5 Structure of Communication Cycle

25.3.2.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters:

Number of Static Slots (FLXAnFRGTUC7.NSS)

Static Slot Length (FLXAnFRGTUC7.SSL)

Payload Length Static (FLXAnFRMHDC.SFDL)

Action Point Offset (FLXAnFRGTUC9.APO)

25.3.2.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters:

Number of Minislots (FLXAnFRGTUC8.NMS) Minislot Length (FLXAnFRGTUC8.MSL)

Minislot Action Point Offset (FLXAnFRGTUC9.MAPO)

Start of Latest Transmit (last minislot) (FLXAnFRMHDC.SLT)

25.3.2.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are send in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters:

Symbol Window Action Point Offset (FLXAnFRGTUC9.APO) (same as for static slots)

Network Idle Time Start Position (FLXAnFRGTUC4.NIT)

25.3.2.4 Network Idle Time (NIT)

During network idle time the CC performs the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters:

Network idle time start position configuration bit (FLXAnFRGTUC4.NIT)

Offset correction start position configuration bit (FLXAnFRGTUC4.OCS)

25.3.2.5 Configuration of NIT Start and Offset Correction Start

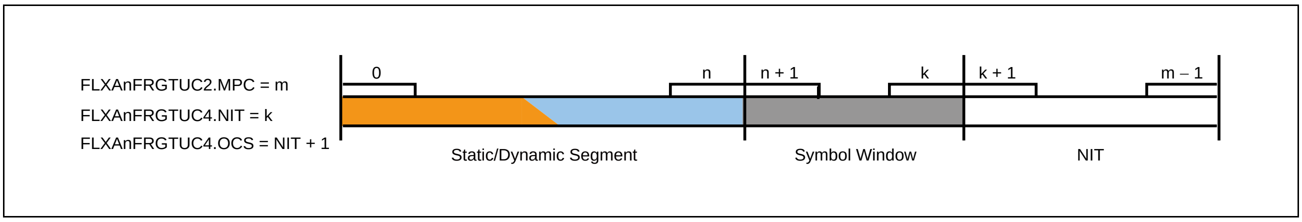


Figure 25.6 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming FLXAnFRGTUC2.MPC = m.

The static/dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset + dynamic segment length – 1MT

n = gNumberOfStaticSlots o gdStaticSlot + dynamic segment offset + gNumberOfMinislots o gdMinislot – 1 MT

The static segment length is configured by FLXAnFRGTUC7.SSL and FLXAnFRGTUC7.NSS.

The dynamic segment length is configured by FLXAnFRGTUC8.MSL and FLXAnFRGTUC8.NMS. The dynamic segment offset is:

If gdActionPointOffset ≤ gdMinislotActionPointOffset: dynamic segment offset = 0 MT

Else if gdActionPointOffset > gdMinislotActionPointOffset:

dynamic segment offset = gdActionPointOffset – gdMinislotActionPointOffset

The NIT starts with macrotick k + 1 and ends with the last macrotick of cycle m – 1. It has to be configured by setting FLXAnFRGTUC4.NIT = k.

For the FlexRay module the offset correction start is required to be FLXAnFRGTUC4.OCS ≥ FLXAnFRGTUC4.NIT + 1 = k + 1.

The length of symbol window results from the number of macroticks between the end of the static/dynamic segment and the beginning of the NIT. It can be calculated by the number of macroticks (k – n).

25.3.3 Communication Modes

The FlexRay Protocol Specification defines the Time-Triggered Distributed (TT-D) mode.

25.3.3.1 Time-Triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- Pure static: Minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

25.3.4 Clock Synchronization

In TT-D mode, distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

25.3.4.1 Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values: the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster definitions:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (µT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

25.3.4.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node’s local time difference measurements is a microtick (µT).

Node definitions:

- Oscillator clock → prescaler → microtick (µT)
- µT = basic unit of time measurement in a CC, clock correction is done in units of → µT
- Cycle counter + macrotick counter = nodes local view of the global time

25.3.4.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXAnFRGTUC2.SNM) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see *the FlexRay protocol specification v2.1, chapter 8*.

(1) Offset (Phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of µTs
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened/shortened)

(2) Rate (Frequency) Correction

- Pairs of deviation values measured and stored in even/odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of µTs
- Distributed over macroticks comprising the next even/odd cycle pair (MTs lengthened/shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit FLXAnFRMRC.SPLM has to be programmed to 1.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames FLXAnFRSUCC1.TXSY must be set to 1.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset/rate correction value is a signed integer
- External offset/rate correction value is added to calculated offset/rate correction value
- Aggregated offset/rate correction term (external + internal) is not checked against configured limits

25.3.5 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FLXAnFREIR.PEMC to 1 and may trigger an interrupt to the Host if enabled. The actual error mode is signaled by FLXAnFRCCEV.ERRM.

Table 25.108 Error Modes of the POC (Degradation Model)

Error Mode	Activity
ACTIVE	Full operation, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
PASSIVE	Reduced operation, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
COMM_HALT	Operation halted, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. The bus drivers are disabled.

25.3.5.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the “maximum without clock correction passive” limit defined by FLXAnFRSUCC3.WCP, the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the “maximum without clock correction fatal” limit defined by FLXAnFRSUCC3.WCF, it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter FLXAnFRCCEV.CCFC allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FLXAnFRSFS.MOCS or the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FLXAnFRSFS.MOCS nor the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value FLXAnFRSUCC3.WCF is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set to 1.

25.3.5.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FLXAnFRSUCC1.PTA defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FLXAnFRSUCC1.PTA is set to zero the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

25.3.5.3 HALT Command

If the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing FLXAnFRSUCC1.CMD = “0110_B”. In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to confirm that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state FLXAnFRSUCC1.CMD will be reset to “0000_B” = command_not_accepted and bit FLXAnFREIR.CNA is set to 1. If enabled an interrupt to the Host is generated.

25.3.5.4 FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing FLXAnFRSUCC1.CMD = “0111_B”. The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

CAUTION

When the communication is stopped by the FREEZE or READY command and the communication is restarted as Leading ColdStart, the startup frame is not transmitted on cycle0 depending on the internal condition of the FlexRay module. This case occurs when the Startup frame is set in one of the slot 1 to slot 7.

This does not occur in ColdStart after a hardware reset.

Even if this occurs, the second trial of ColdStart will succeed. ColdStart time becomes longer, but ColdStart will not be affected by the occurrence.

To avoid this, allocate the Startup/Sync frame in the static slot 8 or higher.

25.3.6 Communication Controller States

25.3.6.1 Communication Controller State Diagram

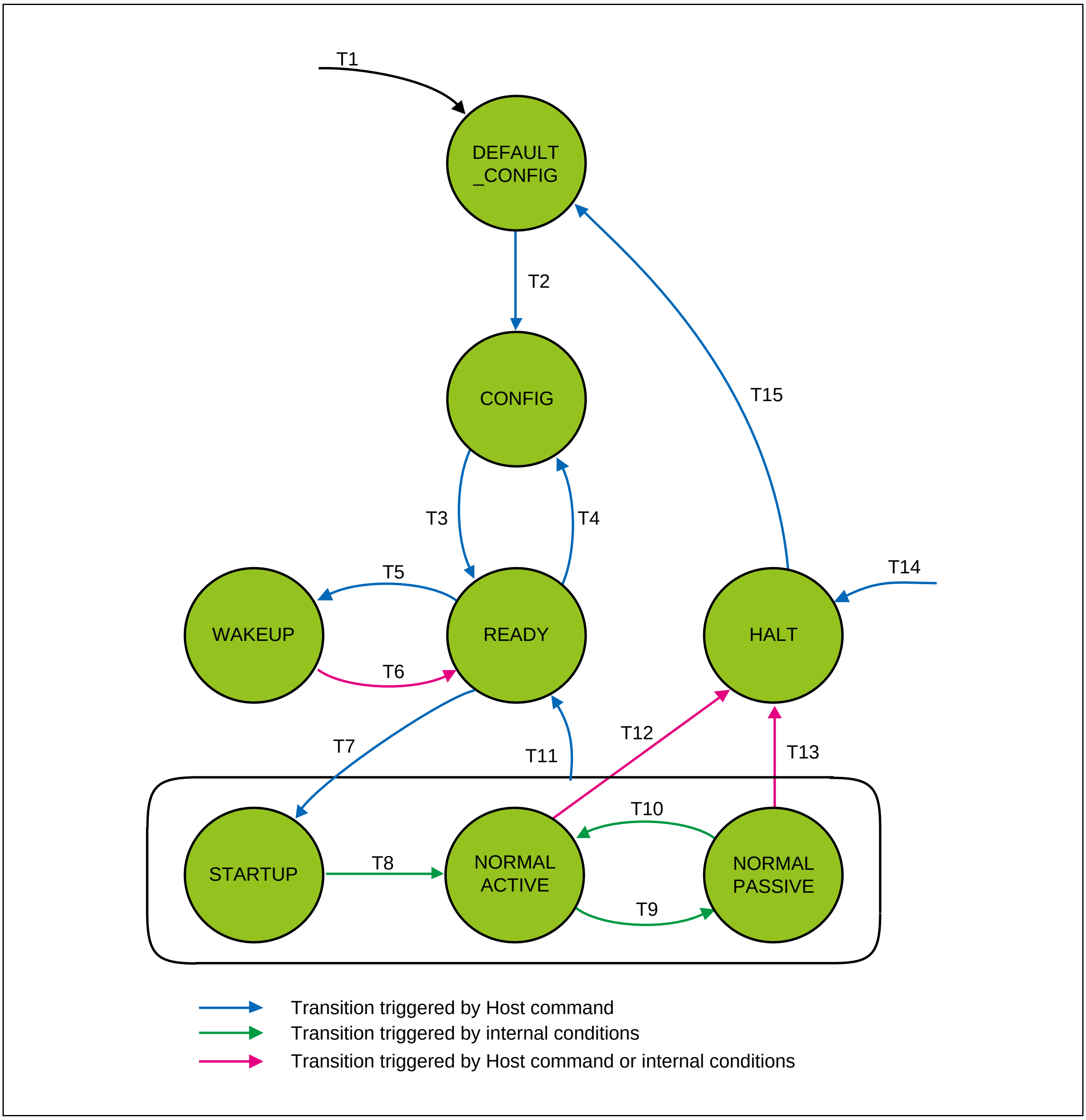


Figure 25.7 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by reset, FLXAnFR0RXDA, FLXAnFR0RXDB, by the POC state machine, and by the CHI Command Vector FLXAnFRSUCC1.CMD.

The CC transits from all states to HALT state after application of the FREEZE command (FLXAnFRSUCC1.CMD = “0111_B”).

Table 25.109 State Transitions of FlexRay Overall State Machine

T#	Condition	From	To
1	Reset	All States	DEFAULT_CONFIG
2	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command READY, FLXAnFRSUCC1.CMD = “0010_B”	CONFIG	READY
4	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	READY	CONFIG
5	Command WAKEUP, FLXAnFRSUCC1.CMD = “0011_B”	READY	WAKEUP
6	Complete transmission of wakeup pattern or received WUP or received frame header or wakeup collision detection or command READY, FLXAnFRSUCC1.CMD = “0010_B”	WAKEUP	READY
7	Command RUN, FLXAnFRSUCC1.CMD = “0100_B”	READY	STARTUP
8	Successful STARTUP	STARTUP	NORMAL_ACTIVE
9	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by FLXAnFRSUCC3.WCP	NORMAL_ACTIVE	NORMAL_PASSIVE
10	Number of valid correction terms reached the Passive to Active limit configured by FLXAnFRSUCC1.PTA	NORMAL_PASSIVE	NORMAL_ACTIVE
11	Command READY, FLXAnFRSUCC1.CMD = “0010_B”	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
12	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD = “0110_B”	NORMAL_ACTIVE	HALT
13	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD = “0110_B”	NORMAL_PASSIVE	HALT
14	Command FREEZE, FLXAnFRSUCC1.CMD = “0111_B”	All States	HALT
15	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	HALT	DEFAULT_CONFIG

25.3.6.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When the reset is applied (HW reset or SW reset)
- When exiting from HALT state

To leave DEFAULT_CONFIG state the Host has to write FLXAnFRSUCC1.CMD = “0001_B”. The CC then transits to CONFIG state.

25.3.6.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyze status information and configuration. Before leaving CONFIG state the Host has to confirm that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described is **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. Directly after unlocking the CONFIG state the Host has to write FLXAnFRSUCC1.CMD to enter the next state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (bus clock and sample clock). To do this the Host has to confirm that all Message RAM transfers have finished before turning off the clocks.

25.3.6.4 READY State

After unlocking CONFIG state and writing FLXAnFRSUCC1.CMD = “0010_B”, the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing FLXAnFRSUCC1.CMD = “0010_B” (READY command).

The CC exits from this state

- To CONFIG state by writing FLXAnFRSUCC1.CMD = “0001_B” (CONFIG command)
- To WAKEUP state by writing FLXAnFRSUCC1.CMD = “0011_B” (WAKEUP command)
- To STARTUP state by writing FLXAnFRSUCC1.CMD = “0100_B” (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

25.3.6.5 WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in *the FlexRay protocol specification v2.1, section 7.1*.

The CC enters this state

- When exiting from READY state by writing FLXAnFRSUCC1.CMD = “0011_B” (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern (WUP)
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing FLXAnFRSUCC1.CMD = “0010_B” (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing FLXAnFRSUCC1.WUCS. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag FLXAnFRSIR.WST. The wakeup status vector can be read from FLXAnFRCCSV.WSV. If a valid wakeup pattern was received also either flag FLXAnFRSIR.WUPA or flag FLXAnFRSIR.WUPB is set to 1.

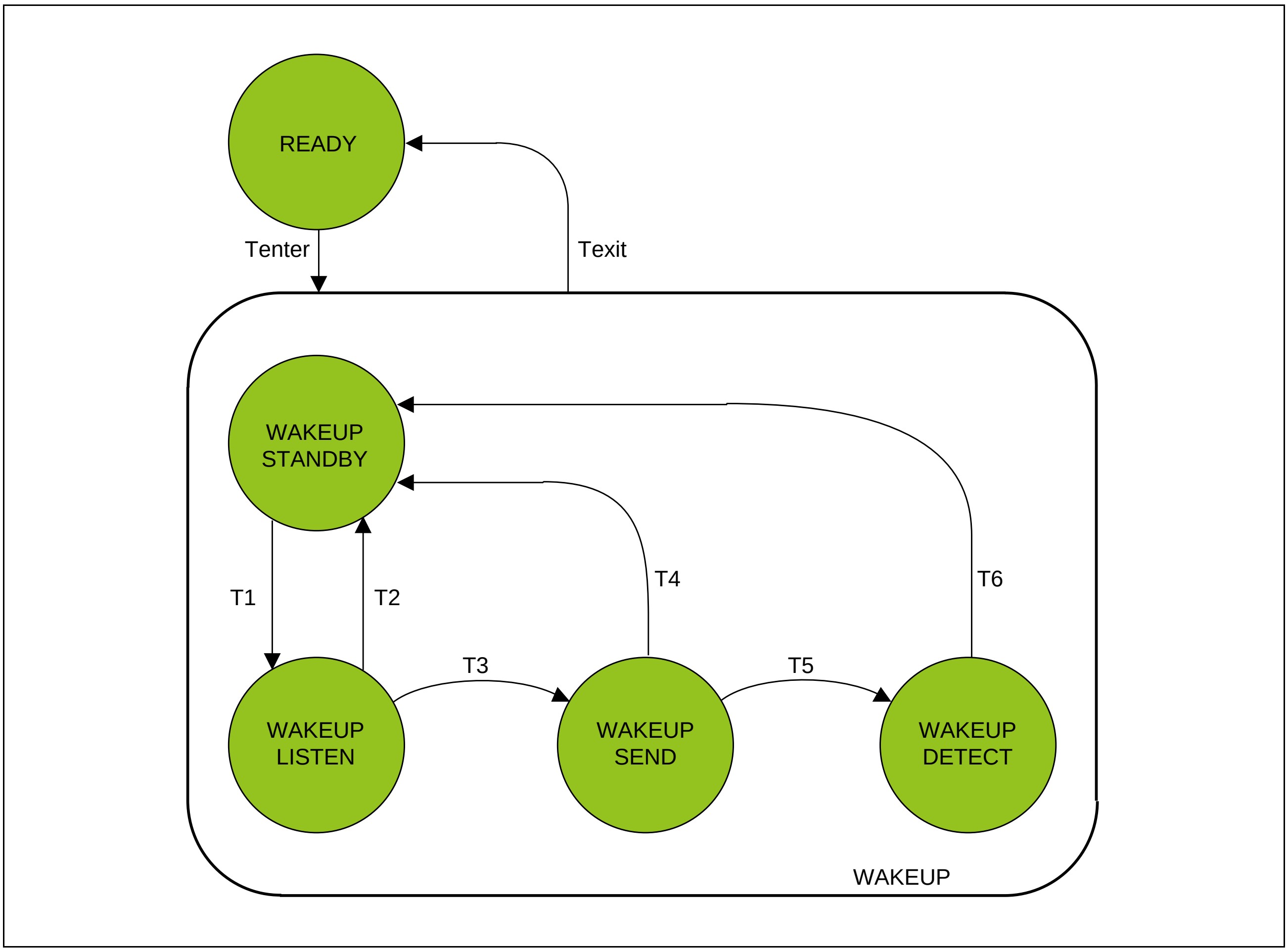


Figure 25.8 Structure of POC State WAKEUP

Table 25.110 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing FLXAnFRSUCC1.CMD = “0011_B” (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired or WUP detected on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header received on either available	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) or Host commands change to READY state by writing FLXAnFRSUCC1.CMD = “0010_B” (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the ListenTimeout (FLXAnFRSUCC2.LT bit) and ListenTimeoutNoise (FLXAnFRSUCC2.LTN bit) values. ListenTimeout enables a fast cluster wakeup in case of a noise free environment, while ListenTimeoutNoise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by FLXAnFRSUCC2.LT. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification recommends that two different CCs shall awake the two channels.

(1) Host Activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signaled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the CC in CONFIG state
  - Select wakeup channel by programming FLXAnFRSUCC1.WUCS
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing FLXAnFRSUCC1.CMD = “0011_B”
  - CC enters WAKEUP
  - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
  - In a dual channel cluster wait for WUP on the other channel
  - Reset coldstart inhibit flag FLXAnFRCCSV.CSI by writing FLXAnFRSUCC1.CMD = “1001_B” (ALLOW_COLDSTART command)
- Command CC to enter startup by writing FLXAnFRSUCC1.CMD = “0100_B” (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing FLXAnFRSUCC1.CMD = “0100_B” (RUN command)

(2) Wakeup Pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FLXAnFRPRTC1 and FLXAnFRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by FLXAnFRPRTC2.TXL
- Wakeup symbol idle time used to listen for activity on the bus, configured by FLXAnFRPRTC2.TXI
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by FLXAnFRPRTC1.RWP (2 to 63 repetitions)
- Wakeup symbol receive window length configured by FLXAnFRPRTC1.RXW
- Wakeup symbol receive low time configured by FLXAnFRPRTC2.RXL
- Wakeup symbol receive idle time configured by FLXAnFRPRTC2.RXI

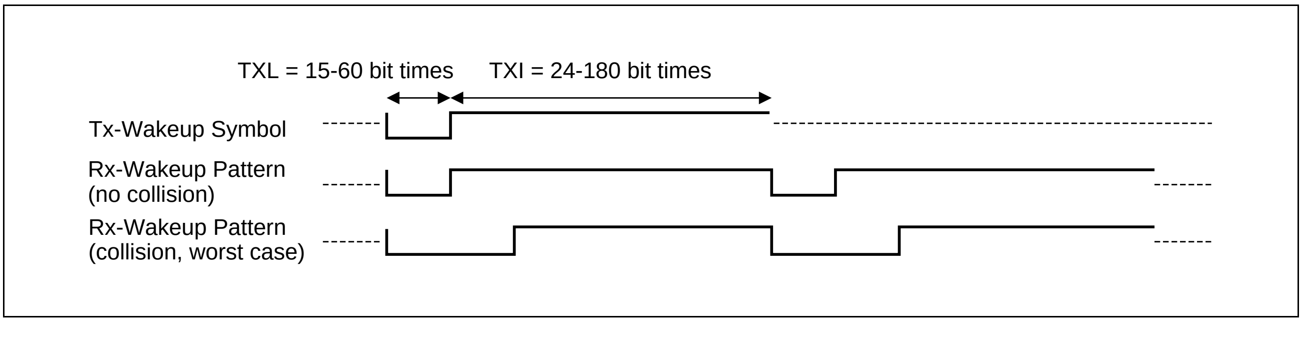


Figure 25.9 Timing of Wakeup Pattern

25.3.6.6 STARTUP State

The description below is intended to help configuring startup for the FlexRay module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in *the FlexRay protocol specification v2.1, section 7.2*.

Any node entering STARTUP state that has coldstart capability should confirm that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is used to ensure that all nodes are synchronized initially. In general, a node may enter NORMAL_ACTIVE state via (see **Figure 25.10, State Diagram Time — Triggered Startup**):

- Coldstart path initiating the schedule synchronization (LeadingColdstart node)
- Coldstart path joining other coldstart nodes (FollowingColdstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY set to 1. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set to 1.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by FLXAnFRSUCC1.CSA.

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

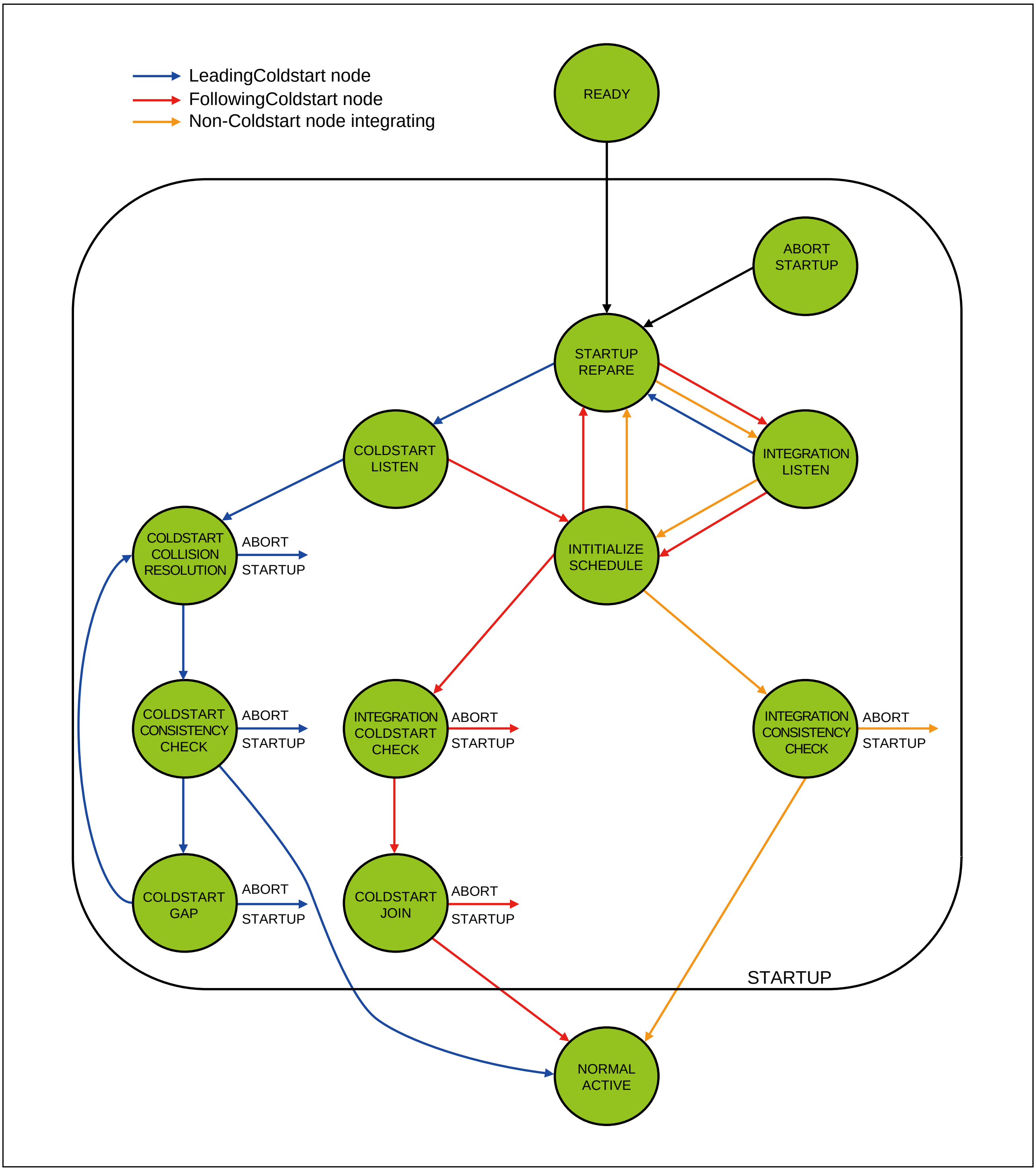


Figure 25.10 State Diagram Time — Triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit FLXAnFRCCSV.CSI is set to 1, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXAnFRCCSV.CSI is set to 1 whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = “1001_B”)

(2) Startup Timeouts

The CC supplies two different µT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

CAUTION

The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXAnFRSUCC2.LT and FLXAnFRSUCC2.LTN.

(a) Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming FLXAnFRSUCC2.LT (see **Section 25.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup timeout is:

pdListenTimeout = FLXAnFRSUCC2.LT

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

(b) Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming FLXAnFRSUCC2.LTN
(see **Section 25.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup noise timeout is:

pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT × (FLXAnFRSUCC2.LTN + 1)

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won’t be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (Initiating Coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FLXAnFRSUCC1.CSA. The number of remaining coldstarts attempts can be read from FLXAnFRCCSV.RCA. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (Responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is confirmed that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still valid. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-Coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node’s own schedule. Clock correction is activated, and if any errors are signaled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

25.3.6.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even/odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing FLXAnFRSUCC1.CMD = “0110_B” (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing FLXAnFRSUCC1.CMD = “0010_B” (READY command)

25.3.6.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing FLXAnFRSUCC1.CMD = “0110_B” (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA – 1
- To READY state by writing FLXAnFRSUCC1.CMD = “0010_B” (READY command)

25.3.6.9 HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing FLXAnFRSUCC1.CMD = “0110_B” (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1

The CC exits from this state to DEFAULT_CONFIG state

- By writing FLXAnFRSUCC1.CMD = “0001_B” (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analyzing purposes.

When the Host writes FLXAnFRSUCC1.CMD = “0110_B” (HALT command), the CC sets bit FLXAnFRCCSV.HRQ to 1 and enters HALT state at the next end of cycle.

When the Host writes FLXAnFRSUCC1.CMD = “0111_B” (FREEZE command), the CC enters HALT state immediately and sets bit FLXAnFRCCSV.FSI to 1.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

25.3.7 Network Management

The accrued Network Management (NM) vector can be read from registers FLXAnFRNMV1 to FLXAnFRNMV3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by FLXAnFRNEMC.NML. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set as 1, bit PPIT in the header section of the respective transmit buffer has to be set to 1 via FLXAnFRWRHS1.PPIT. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

Section 25.3.17, Byte Alignment, for byte alignment of the received NM vector in registers FLXAnFRNMV1 to FLXAnFRNMV3.

CAUTIONS

1. In case a message buffer is configured for transmission/reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by FLXAnFRNEMC.NML.
2. When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FLXAnFRNMV1 to FLXAnFRNMV3 holds the value from the cycle before.

25.3.8 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated/transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance/transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

CAUTION

For the FIFO the acceptance filter is configured by the FIFO Rejection Filter (FLXAnFRFRF) and the FIFO Rejection Filter Mask (FLXAnFRFRFM).

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

25.3.8.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID and channel ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

25.3.8.2 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 or 1 is configured to hold the startup/sync frame or the single slot frame by bits FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, cycle counter filtering for message buffer 0 or 1 shall be disabled.

CAUTION

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in **Table 25.111, Definition of Cycle Set**.

Table 25.111 Definition of Cycle Set

Cycle Code	Matching Cycle Counter Values
0b000000x	all Cycles
0b000001c	every second Cycle at (Cycle Count) mod2 = c
0b00001cc	every fourth Cycle at (Cycle Count) mod4 = cc
0b0001ccc	every eighth Cycle at (Cycle Count) mod8 = ccc
0b001cccc	every sixteenth Cycle at (Cycle Count) mod16 = cccc
0b01ccccc	every thirty-second Cycle at (Cycle Count) mod32 = ccccc
0b1cccccc	every sixty-fourth Cycle at (Cycle Count) mod64 = cccccc

Table 25.112, Examples for Valid Cycle Sets below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 25.112 Examples for Valid Cycle Sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7-….-63
0b0000100	0-4-8-12-….-60
0b0001110	6-14-22-30-….-62
0b0011000	8-24-40-56
0b0100011	3-35
0b1001001	9

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer’s cycle set. Channel ID and frame ID must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Channel ID and frame ID must also match.

25.3.8.3 Channel ID Filtering

There is a 2-bit channel filtering field (CH) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see **Table 25.113, Channel Filtering Configuration**).

Table 25.113 Channel Filtering Configuration

CH[1:0]	Transmit Buffer Transmit Frame	Receive Buffer Store Valid Receive Frame
00_B	no transmission	ignore frame
01_B	on channel A	received on channel A
10_B	on channel B	received on channel B
11_B	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CH = “11_B”).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CH = “11_B”).

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as CH = “00_B”).

25.3.8.4 FIFO Filtering

For FIFO filtering registers FLXAnFRFRF and FLXAnFRFRFM are used. The FIFO filter consists of channel filter FLXAnFRFRF.CH, frame ID filter FLXAnFRFRF.FID, and cycle counter filter FLXAnFRFRF.CYF. Registers FLXAnFRFRF and FLXAnFRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

25.3.9 Transmit Process

25.3.9.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

25.3.9.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by FLXAnFRMHDC.SLT defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

25.3.9.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit CFG in the header section of the respective message buffer to 1 via FLXAnFRWRHS1.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B
- Dynamic segment: channel A or channel B

Message buffers 0 and 1 are dedicated to holding the startup frame, the sync frame, or the designated single slot frame as configured by FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup/sync frame per communication cycle. Transmission of startup/sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 25.3.12.1, Reconfiguration of Message Buffers**). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the PPIT bit in the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer (see **Section 25.3.7, Network Management**).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by FLXAnFRMHDC.SFDL, the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is “0000_H”.

CAUTION

In case of an odd payload length (PLC = 1, 3, 5, ...) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is “0000_H”.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode the CC resets the respective TXR flag to 0 after transmission has completed. Now the Host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR to 0 after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset to 0 by the Host by writing the respective message buffer number to the FLXAnFRIBCR register while bit FLXAnFRIBCM.STXRH is set to 0.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

25.3.9.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Write the data section of the transmit buffer via FLXAnFRWRDSx
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR
- If configured in register FLXAnFRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = 0) in the FLXAnFRTXRQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FLXAnFRTXRQ1/2/3/4 register is reset to 0 (single-shot mode), and, if bit MBI in the header section of the message buffer is set to 1, flag FLXAnFRSIR.TXI is set to 1. If enabled, an interrupt is generated.

25.3.9.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag to 1 before transmit time, the CC transmits a null frame with the null frame indication bit set to 0 and the payload data set to zero.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (TXR = 0) to 1.
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status FLXAnFRMBS is updated.

Null frames are not transmitted in the dynamic segment.

25.3.10 Receive Process

25.3.10.1 Dedicated Receive Buffers

A portion of the FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to 0 via FLXAnFRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 25.3.12.1, Reconfiguration of Message Buffers**). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

25.3.10.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FLXAnFRNDAT1/2/3/4 registers is set to 1, and, if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.RXI is set to 1. If enabled, an interrupt is generated.

In case that bit ND was already set to 1 when the Message Handler updates the message buffer, bit FLXAnFRMBS.MLST of the respective message buffer is set to 1 and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status FLXAnFRMBS is updated.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 registers is set to 1, and if bit MBI in the header section of that message buffer is set, flag FLXAnFRSIR.MBSI is set to 1. If enabled an interrupt is generated.

If the payload length of a received frame PLR is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in **Section 25.3.12.2(2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

The ND and MBC flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

25.3.10.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status FLXAnFRMBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 register is set to 1, and if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.MBSI is set to 1. If enabled, an interrupt is generated.

25.3.11 FIFO Function

25.3.11.1 Description

A portion of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXAnFRMRC.FFB and ending with the message buffer referenced by FLXAnFRMRC.LCB. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status FLXAnFRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit FLXAnFRSIR.RFNE shows that the FIFO is not empty, bit FLXAnFRSIR.RFCL is set to 1 when the receive FIFO fill level FLXAnFRFSR.RFFL is equal or greater than the critical level as configured by FLXAnFRSIR.RFCL, bit FLXAnFREIR.RFO shows that a FIFO overrun has been detected. If enabled, interrupts are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXAnFREIR.RFO to 1.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag FLXAnFRSIR.RFNE is set to 1. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in **Figure 25.11, FIFO Status: Empty, Not Empty, Overrun** for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FLXAnFRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXAnFRFRF.RSS is set to 1, all messages received in the static segment are rejected by the FIFO. If bit FLXAnFRFRF.RNF is set to 1, received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FLXAnFRFRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked “don’t care” for rejection filtering.

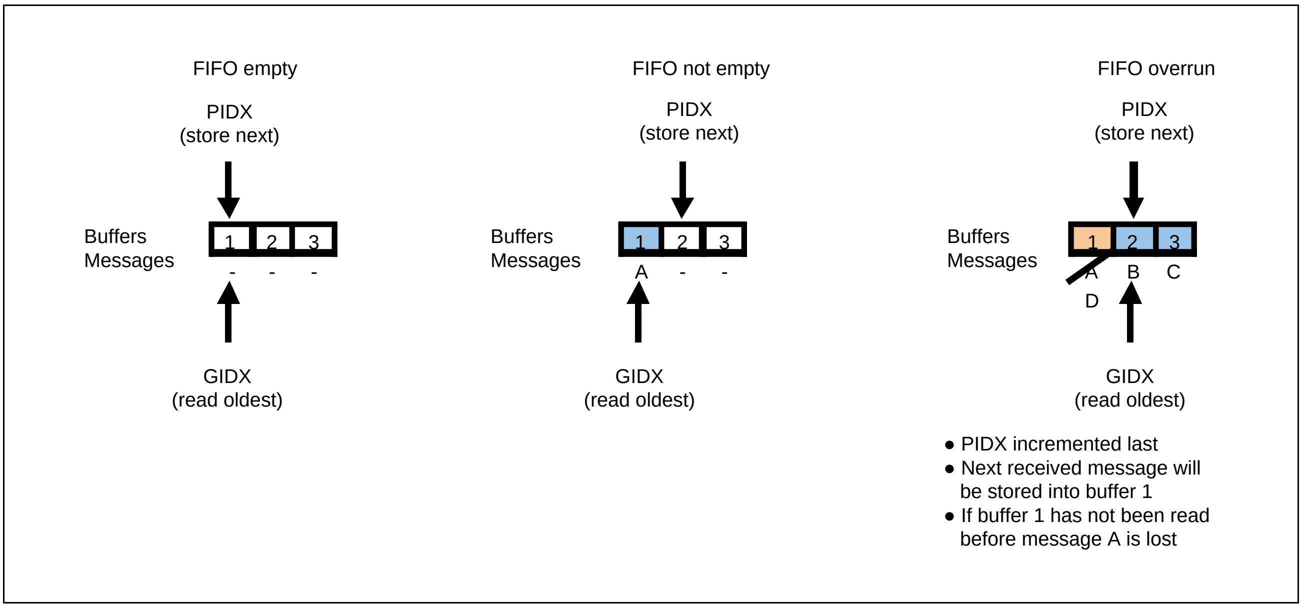


Figure 25.11 FIFO Status: Empty, Not Empty, Overrun

25.3.11.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via FLXAnFRWRHS2.PLC. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via FLXAnFRWRHS3.DP.

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

CAUTIONS

1. It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via FLXAnFRWRHS1.MBI to avoid generation of interrupts.
2. If the payload length of a received frame is longer than the value programmed by FLXAnFRWRHS2.PLC in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

25.3.11.3 Access to the FIFO

(1) When the Output Buffer is Used:

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FLXAnFRMRC.FFB) to the register FLXAnFROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

(2) When the Data Transfer Function is Used:

The message received in FIFO can be transferred to the local RAM/global RAM by using the output data transfer function. For the output data transfer function, see **Section 25.3.16.2, Output Data Transfer**.

25.3.12 Message Handling

The Message Handler controls data transfers between the Input/Output Buffer and the Message RAM and between the Message RAM and the two Temporary buffers.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to FLXAnFRGTUC7.NSS. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FLXAnFRGTUC7.NSS + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

Access of the Host to the message buffer contents using the input or output buffer function is described in this subsection. Access to the message buffer contents using the data transfer function is mentioned in **Section 25.3.16, Usage of Data Transfer**.

25.3.12.1 Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers FLXAnFRWRHS1 to FLXAnFRWRHS3.

Reconfiguration has to be enabled via control bits FLXAnFRMRC.SEC in the Message RAM Configuration register.

If a message buffer has not been transmitted/updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission/reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted/updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to **Table 25.114, Scan of Message RAM** below.

Table 25.114 Scan of Message RAM

Start of Scan in Slot	Scan for Slots
1	2 … 15, 1 (next cycle)
8	16 … 23, 1 (next cycle)
16	24 … 31, 1 (next cycle)
24	32 … 39, 1 (next cycle)
…	…

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by FLXAnFRMRC.FDB. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by FLXAnFRMRC.FDB.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the “Static Buffers”, it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the “Static + Dynamic Buffers”, it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

CAUTION

Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted/updated from a received frame.

25.3.12.2 Host Access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target/source message buffer to be accessed to FLXAnFRIBCR or FLXAnFROBCR register.

The FLXAnFRIBCM and FLXAnFROBCM registers can be used to write/read header and data section of the selected message buffer separately.

If bit FLXAnFRIBCM.STXR is set to 1, the transmission request flag TXR of the selected message buffer is automatically set to 1 after the message buffer has been updated. If bit FLXAnFRIBCM.STXR is reset to 0, the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are configured as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host/OBF Host), while the other half (IBF Shadow/OBF Shadow) is accessed by the Message Handler for data transfers between IBF/OBF and Message RAM.

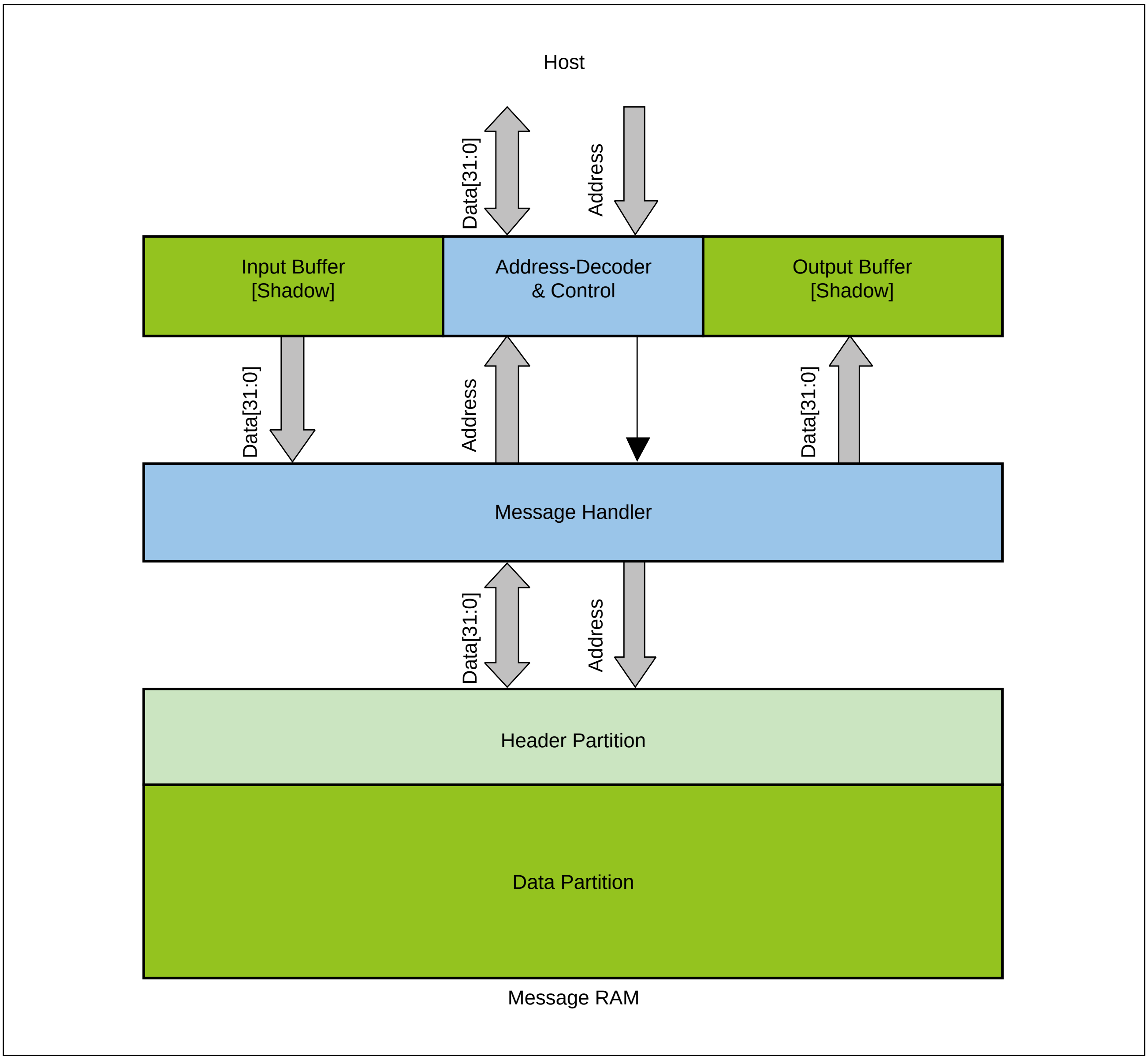


Figure 25.12 Host Access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure/update a message buffer in the Message RAM, the Host has to write the data to FLXAnFRWRDSx and the header to FLXAnFRWRHS1 to FLXAnFRWRHS3. The specific action is selected by configuring the FlexRay Input Buffer Command Mask FLXAnFRIBCM.

When the Host writes the number of the target message buffer in the Message RAM to FLXAnFRIBCR.IBRH, IBF Host and IBF Shadow are swapped (see **Figure 25.13, Double Buffer Structure Input Buffer**).

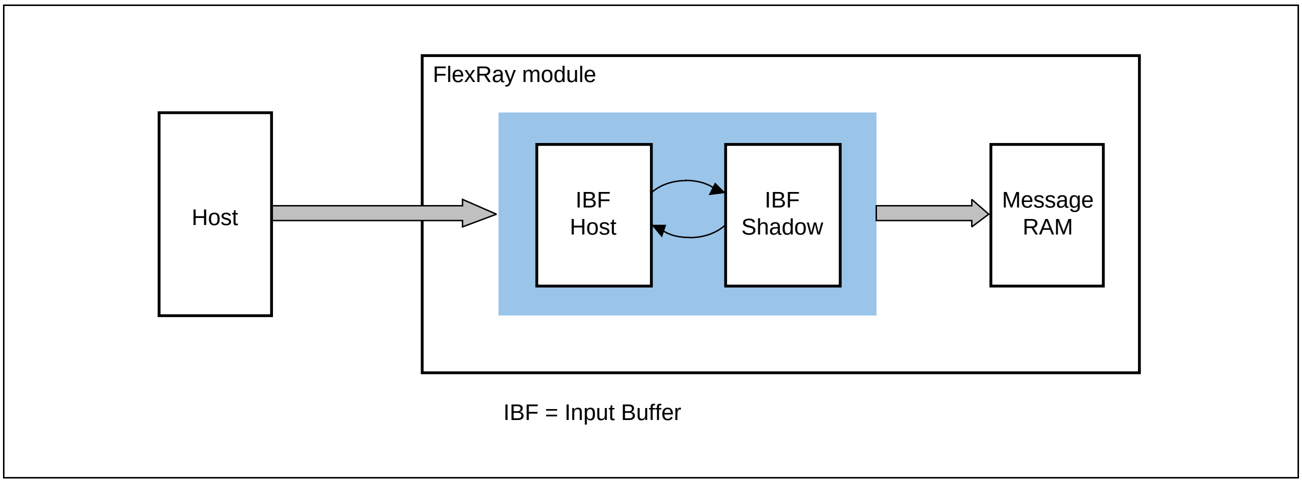


Figure 25.13 Double Buffer Structure Input Buffer

In addition the bits in the FLXAnFRIBCM and FLXAnFRIBCR registers are also swapped to keep them attached to the respective IBF section (see **Figure 25.14, Swapping of FLXAnFRIBCM and FLXAnFRIBCR Bits**).

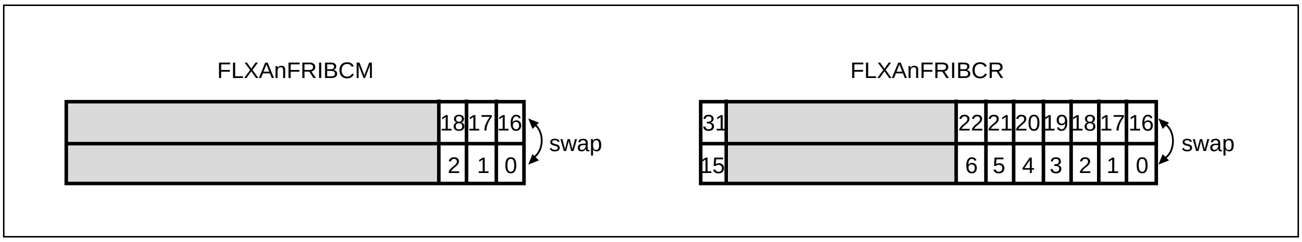


Figure 25.14 Swapping of FLXAnFRIBCM and FLXAnFRIBCR Bits

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by FLXAnFRIBCR.IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0, FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS and the command mask flags are also swapped.

Example of 8/16/32-bit Host Access Sequence:

Configure/update n-th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

Configure/update (n + 1)th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

CAUTION

Any write access to IBF while FLXAnFRIBCR.IBSYH is 1 will set error flag FLXAnFREIR.IIBA to 1. In this case the write access has no effect.

Table 25.115 Assignment of FLXAnFRIBCM Bits

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 25.116 Assignment of FLXAnFRIBCR Bits

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22 to 16	r	IBRS	IBF Request Shadow, number of message buffer from which a message is currently being transferred or was last transferred
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH
6 to 0	r/w	IBRH	IBF Request Host, number of message buffer from which a message is to be transferred next

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FLXAnFROBCR to trigger the data transfer as configured in FLXAnFROBCM. After the transfer has completed, the Host can read the transferred data from FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS.

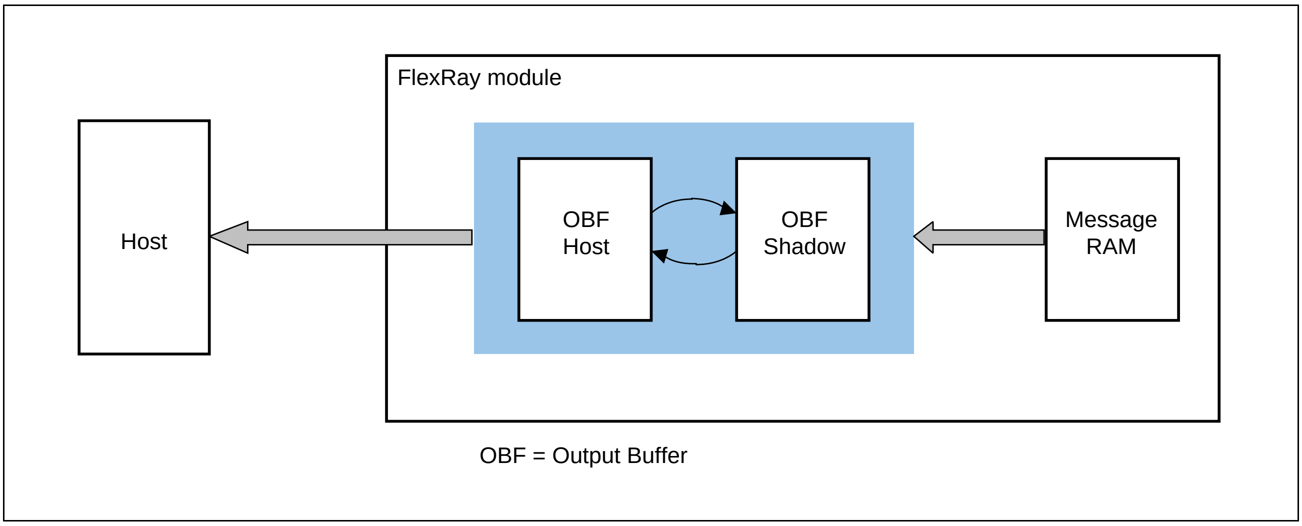


Figure 25.15 Double Buffer Structure Output Buffer

OBF Host and OBF Shadow as well as bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS, FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH and bits FLXAnFROBCR.OBRS, FLXAnFROBCR.OBRH are swapped under control of bits FLXAnFROBCR.VIEW and FLXAnFROBCR.REQ.

Writing bit FLXAnFROBCR.REQ to 1 copies bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS and bits FLXAnFROBCR.OBRS to an internal storage (see **Figure 25.16, Swapping of FLXAnFROBCM and FLXAnFROBCR Bits**).

After setting FLXAnFROBCR.REQ to 1, FLXAnFROBCR.OBSYS is set to 1, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the FLXAnFROBCR.OBSYS bit is set back to 0. Bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW can only be set to 1 while FLXAnFROBCR.OBSYS is 0.

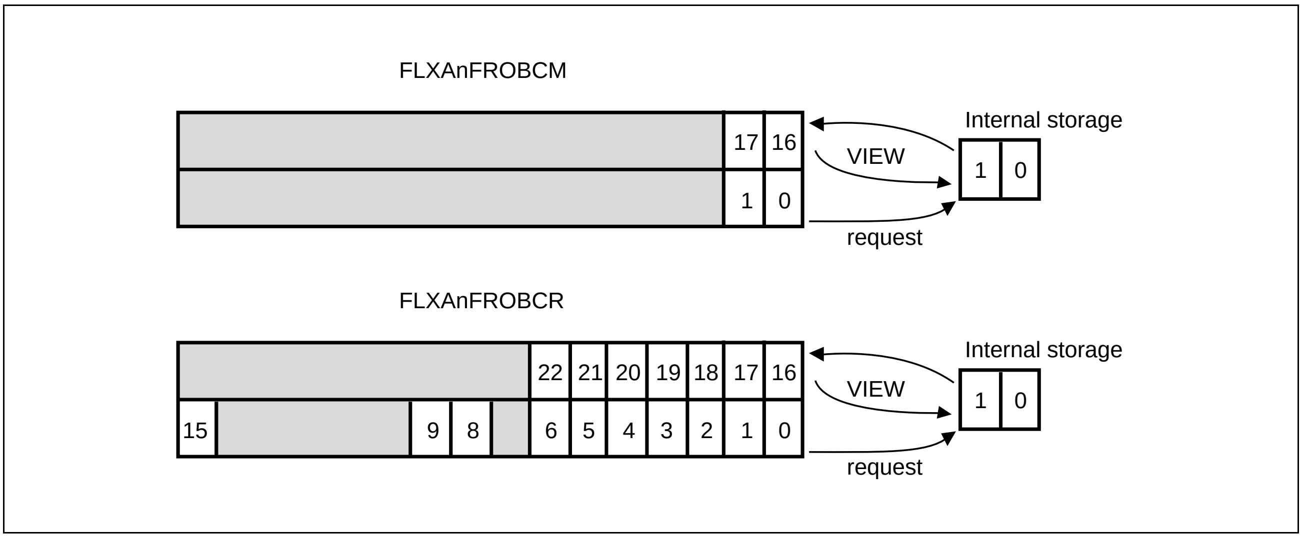


Figure 25.16 Swapping of FLXAnFROBCM and FLXAnFROBCR Bits

OBF Host and OBF Shadow are swapped by setting bit FLXAnFROBCR.VIEW to 1 while bit FLXAnFROBCR.OBSYS is 0 (see **Figure 25.15, Double Buffer Structure Output Buffer**).

In addition bits FLXAnFROBCR.OBRH and bits FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH are swapped with the registers internal storage thus assuring that the message buffer number stored in FLXAnFROBCR.OBRH and the mask configuration stored in FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH matches the transferred data stored in OBF Host (see **Figure 25.16, Swapping of FLXAnFROBCM and FLXAnFROBCR Bits**).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits REQ and VIEW are set to 1 with the same write access while FLXAnFROBSYS is 0, FLXAnFROBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host.

When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Example of an 8/16/32-bit Host Access to a Single Message Buffer:

If a single message buffer has to be read out, two separate write accesses to FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are necessary:

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS
- Request transfer of message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of and 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).
- Wait until FLXAnFROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing FLXAnFROBCR.VIEW = 1
- Read out transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Example of an 8/16/32-bit Host Access Sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of an 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 2nd message buffer

- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing FLXAnFROBCR.OBRS of 2nd message buffer, FLXAnFROBCR.REQ, and FLXAnFROBCR.VIEW (in case of and 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW).
- Read out 1st transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Demand access to last requested message buffer without request of another message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Demand access to last transferred message buffer by writing FLXAnFROBCR.VIEW
- Read out last transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Table 25.117 Assignment of FLXAnFROBCM Bits

Pos.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 25.118 Assignment of FLXAnFROBCR Bits

Pos.	Access	Bit	Function
22 to 16	r	OBRH	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6 to 0	r/w	OBRS	OBF Request Shadow, number of message buffer from which a message is to be transferred next

25.3.12.3 FlexRay Protocol Controller Access to Message RAM

The two Temporary buffers (TBF A, B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Temporary buffer is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If, for example, the Message Handler writes the next message to be sent to Temporary buffer Tx, the FlexRay Channel Protocol Controller can access Temporary buffer Rx to store the message it is actually receiving. During transmission of the message stored in Temporary buffer Tx, the Message Handler transfers the last received message stored in Temporary buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Temporary buffers and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

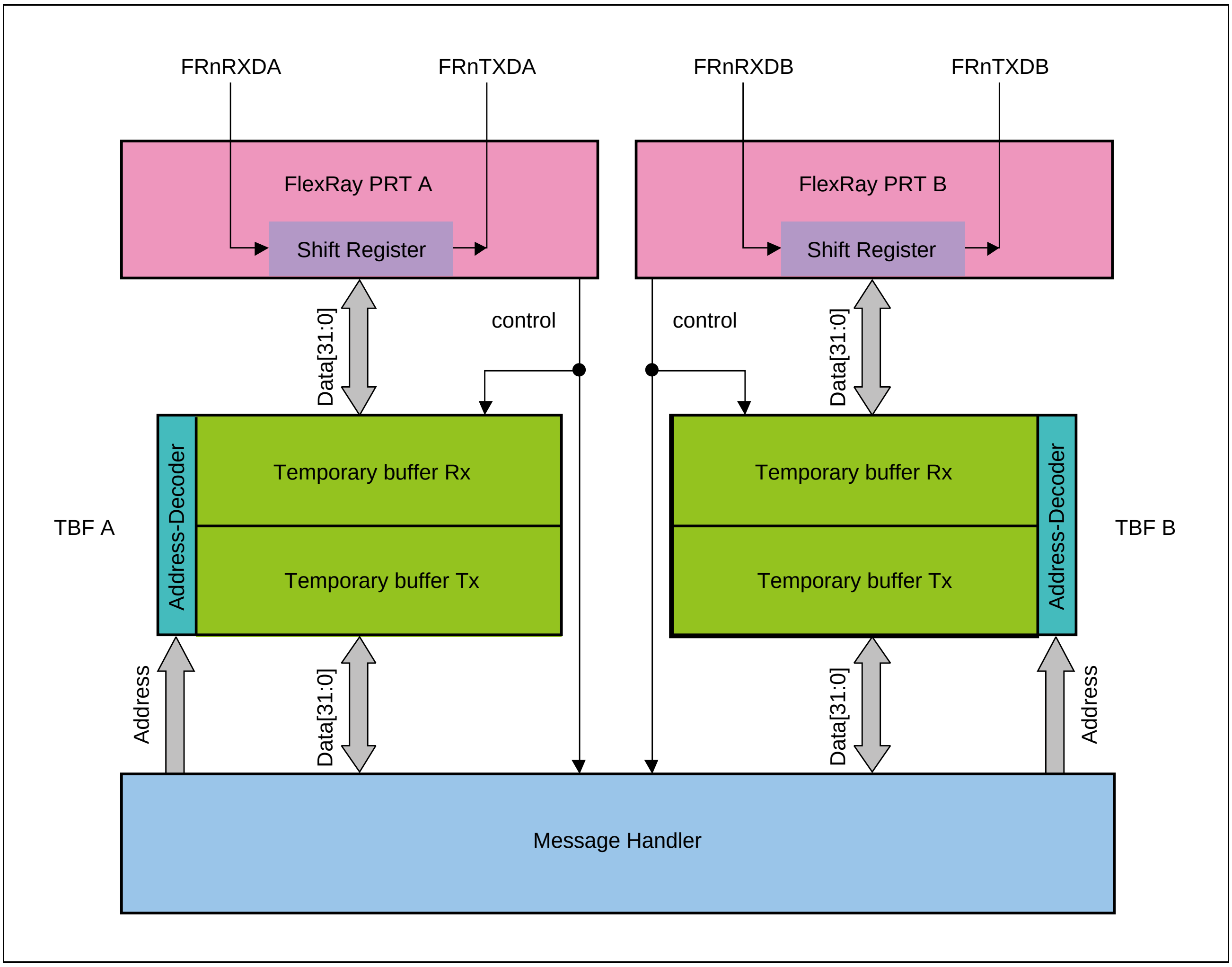


Figure 25.17 Access to Temporary Buffers

25.3.13 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception/transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is able to store up to 2048 32-bit words. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in **Figure 25.18, Configuration Example of Message Buffers in the Message RAM**.

When specifying in the data section of the message buffer allocated immediately after the header partition that the message buffer is a receive buffer (FLXAnFRWRHS1.CFG bit = 0) or a receive FIFO buffer, configure an unused area of at least 32 bits at the start of the data partition. This means that the data partition can start from the message RAM word No. calculated by ((Value set by FLXAnFRMRC.LCB[7:0] bits + 1) × 4) + 1.

When specifying in the data section of the message buffer allocated immediately after the header partition that the message buffer is a transmit buffer (FLXAnFRWRHS1.CFG bit = 1), the data partition can start from the message RAM word No. calculated by ((Value set by FLXAnFRMRC.LCB[7:0] bits + 1) × 4).

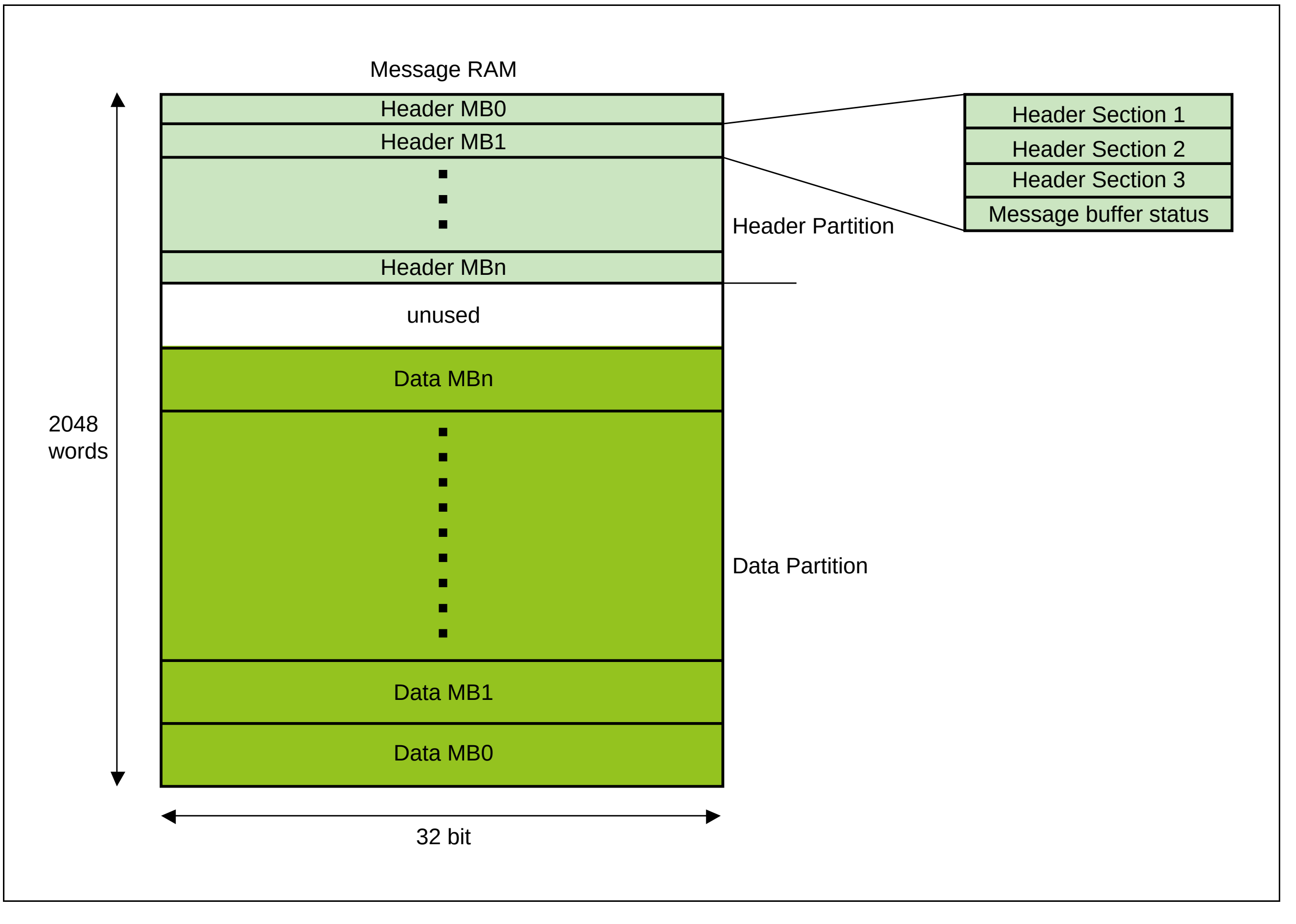


Figure 25.18 Configuration Example of Message Buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

CAUTION

Header partition + data partition may not occupy more than 2048 32-bit words.

25.3.13.1 Header Partition

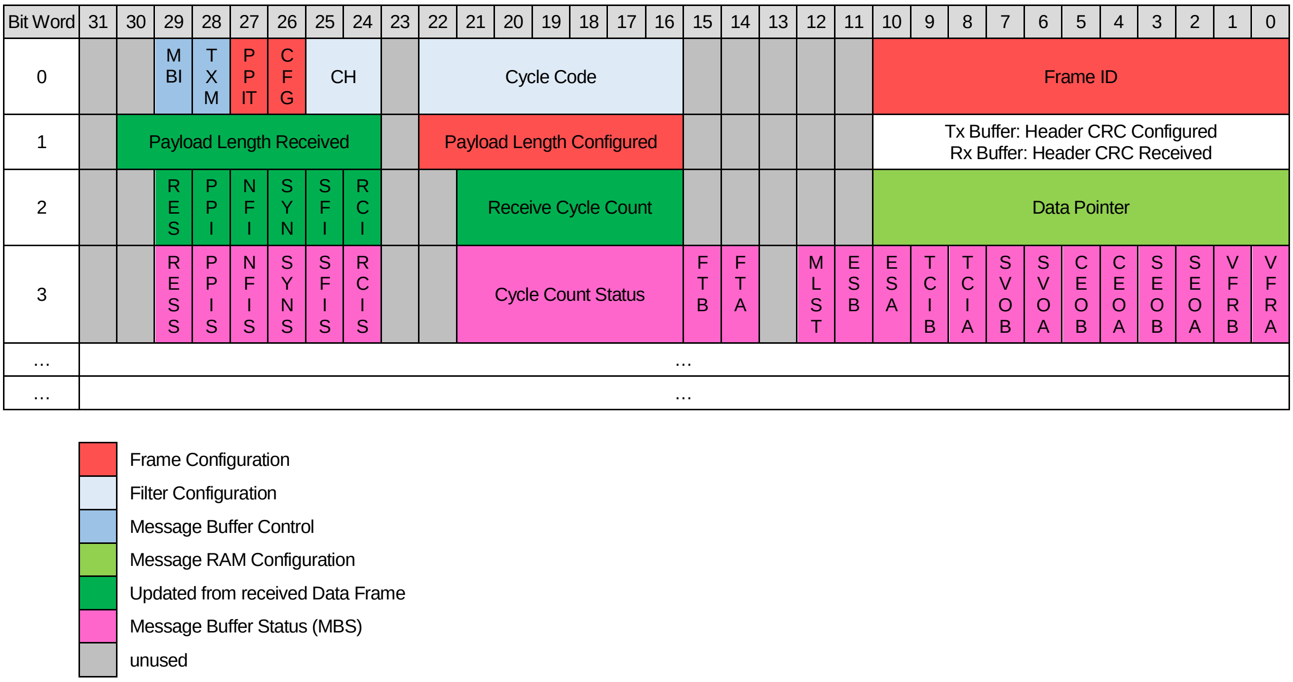
The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in **Table 25.119, Header Section of a Message Buffer in the Message RAM** below. Configuration of the header sections of the message buffers is done via IBF (FLXAnFRWRHS1 to FLXAnFRWRHS3). Read access to the header sections is done via OBF (FLXAnFRRDHS1 to FLXAnFRRDHS3 + FLXAnFRMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 32-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received PLR, Receive Cycle Count RCC, Received on Channel Indicator RCI, Startup Frame Indicator SFI, Sync Frame Indicator SYN, Null Frame Indicator NFI, Payload Preamble Indicator PPI, and Reserved Bit RES are updated from received valid data frames only.

Table 25.119 Header Section of a Message Buffer in the Message RAM



Bit Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			MBI	TXM	PPIT	CFG	CH			Cycle Code												Frame ID										
1		Payload Length Received								Payload Length Configured												Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Received										
2			RES	PPI	NFI	SYN	SFI	RCI			Receive Cycle Count											Data Pointer										
3			RESS	PPIS	NFIS	SYNS	SFIS	RCIS			Cycle Count Status						FTB	FTA		MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
…	…																															
…	…																															

🟥 Frame Configuration
⬜ Filter Configuration
🟦 Message Buffer Control
🟩 Message RAM Configuration
🟩 Updated from received Data Frame
🟪 Message Buffer Status (MBS)
⬜ unused

(1) Header Section 1 (Word 0)

Write access via FLXAnFRWRHS1, read access via FLXAnFRRDHS1:

- Frame ID
  - Slot counter filtering configuration
- Cycle Code
  - Cycle counter filtering configuration
- CH
  - Channel filtering configuration
- CFG
  - Message buffer direction configuration: receive/transmit
- PPIT
  - Payload Preamble Indicator Transmit
- TXM
  - Transmit mode configuration: single-shot/continuous
- MBI
  - Message buffer receive/transmit interrupt enable

(2) Header Section 2 (Word 1)

Write access via FLXAnFRWRHS2, read access via FLXAnFRRDHS2:

- Header CRC
  - Transmit Buffer: Configured by the Host (calculated from frame header)
  - Receive Buffer: Updated from received frame
- Payload Length Configured
  - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
  - Length of payload segment (2-byte words) stored from received frame

(3) Header Section 3 (Word 2)

Write access via FLXAnFRWRHS3, read access via FLXAnFRRDHS3:

- Data Pointer
  - Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXAnFRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count
  - Cycle count from received frame
- RCI
  - Received on Channel Indicator
- SFI
  - Startup Frame Indicator
- SYN
  - Sync Frame Indicator
- NFI
  - Null Frame Indicator
- PPI
  - Payload Preamble Indicator
- RES
  - Reserved bit

(4) Message Buffer Status FLXAnFRMBS (Word 3)

Read access via FLXAnFRMBS, updated by the CC at the end of the configured slot.

- VFRA
  - Valid Frame Received on channel A
- VFRB
  - Valid Frame Received on channel B
- SEOA
  - Syntax Error Observed on channel A
- SEOB
  - Syntax Error Observed on channel B
- CEOA
  - Content Error Observed on channel A
- CEOB
  - Content Error Observed on channel B
- SVOA
  - Slot boundary Violation Observed on channel A

- SVOB
  - Slot boundary Violation Observed on channel B
- TCIA
  - Transmission Conflict Indication channel A
- TCIB
  - Transmission Conflict Indication channel B
- ESA
  - Empty Slot Channel A
- ESB
  - Empty Slot Channel B
- MLST
  - Message Lost
- FTA
  - Frame Transmitted on Channel A
- FTB
  - Frame Transmitted on Channel B
- Cycle Count Status
  - Actual cycle count when status was updated
- RCIS
  - Received on Channel Indicator Status
- SFIS
  - Startup Frame Indicator Status
- SYNS
  - Sync Frame Indicator Status
- NFIS
  - Null Frame Indicator Status
- PPIS
  - Payload Preamble Indicator Status
- RESS
  - Reserved bit Status

25.3.13.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception/transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the user has to confirm that the data pointers point to addresses within the data partition. **Table 25.120, Example for Structure of the Data Partition in the Message RAM** below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see **Table 25.120, Example for Structure of the Data Partition in the Message RAM** below).

Table 25.120 Example for Structure of the Data Partition in the Message RAM

Bit Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
…	unused								unused								unused								unused							
…	unused								unused								unused								unused							
…	MBn Data3								MBn Data2								MBn Data1								MBn Data0							
…	…								…								…								…							
…	…								…								…								…							
…	MBn Data (m)								MBn Data (m – 1)								MBn Data (m – 2)								MBn Data (m – 3)							
…	…								…								…								…							
…	…								…								…								…							
…	…								…								…								…							
…	MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
…	…								…								…								…							
2046	MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
2047	unused								unused								MB0 Data5								MB0 Data4							

25.3.13.3 Message Data Integrity Check

There is a data integrity checking mechanism implemented in the FlexRay core to ensure the integrity of the data stored in the related RAM. Each RAM has a checksum generator/checker attached as shown in **Figure 25.19, Checksum Generation and Check**.

When data is written to a RAM, the local checksum generator generates the checksum. The checksum is stored together with the respective data word. The checksum is checked each time a data word is read from a RAM.

If a checksum error is detected, the respective access error flag is set. The access error flags FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, FLXAnFRMHDS.ATBF2 and the faulty message buffer indicators FLXAnFRMHDS.FMBD, FLXAnFRMHDS.MFMB, FLXAnFRMHDS.FMB are located in the FlexRay Message Handler Status register. These single access error flags control the error interrupt flag FLXAnFREIR.AERR.

Figure 25.19, Checksum Generation and Check shows the data paths between the Input Buffer, Temporary Buffer and Message RAM.

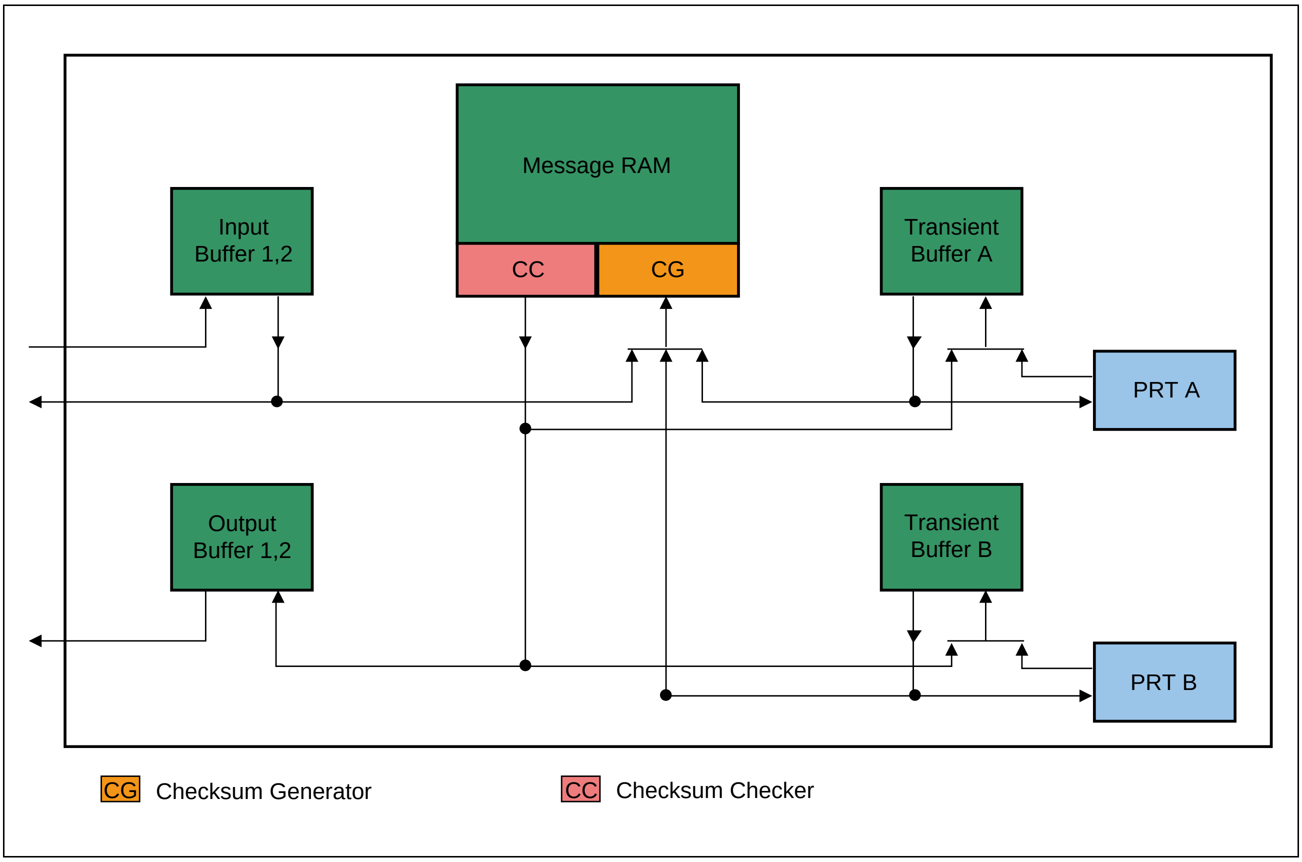


Figure 25.19 Checksum Generation and Check

When an access error has been detected the following actions will be performed:

In all cases:

- The respective access error flag in FLXAnFRMHDS register is set
- The access error flag FLXAnFREIR.AERR is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases:

(1) Access Error during Data Transfer from Input Buffer 1, 2 to Message RAM When Reading Header Section of Respective Message Buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.
- Transmit buffer: Transmission request for the respective message buffer is not set.

(2) Access Error during Scan of Header Sections in Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- Ignore message buffer (message buffer is skipped).

(3) Access Error during Data Transfer from Message RAM to Temporary Buffer 1, 2:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero.

(4) Access Error during Data Transfer from Temporary Buffer 1, 2 to Message RAM When Reading Header Section of Respective Message Buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.

(5) Access Error during Data Transfer from Message RAM to Output Buffer:

- The access error flag FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.

(6) Access Error during a Data Transfer from Temporary Buffer 1, 2 to Protocol Controller 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- Frames already in transmission are invalidated by setting the frame CRC to zero.

(7) Access Error during Data Transfer from Temporary Buffer 1, 2 to Message RAM When Reading Temporary Buffer RAM 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.

(8) Access Error during Data Read of Temporary Buffer RAM 1, 2:

- When an access error occurs while the Message Handler read a frame with network management information (PPI = 1) from the Temporary Buffer RAM 1, 2, the corresponding network management vector registers FLXAnFRNMV1 to 3 are not updated from this frame.

25.3.13.4 Host Handling of Access Errors

Access error caused by temporary bit flips can be fixed by:

(1) Self-Healing

Access errors located in the Data Section of Message RAM, Temporary Buffer RAM A or Temporary Buffer RAM B are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

(2) CLEAR_RAMS Command

The POC command CLEAR_RAMS initializes the message RAM to zero, when called in the DEFAULT_CONFIG or CONFIG state.

(3) Temporary Unlocking of Header Section

An access error in the header section of a locked message buffer can be fixed by a transfer from the Input Buffer to the locked buffer Header Section. For this transfer, the write access to the FLXAnFRIBCR (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see **Section 25.2.3.3, FLXAnFRLCK — FlexRay Lock Register**).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by FLXAnFRMRC.SEC, and will be updated with new data.

25.3.14 Module Interrupt

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable/disable controls for each individual interrupt source separately.

An interrupt request may be triggered when

- An error was detected
- A status flag is set to 1
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A message transfer from the local RAM/global RAM to Message RAM or from Message RAM to local RAM/global RAM has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers FLXAnFREIR, FLXAnFRSIR, FLXAnFROS, FLXAnFROTS and FLXAnFRITS.

The general purpose interrupt lines to the Host, FlexRay Interrupt 0, FlexRay Interrupt 1, are controlled by the enabled interrupts in FLXAnFREIES and FLXAnFRSIES. In addition each of the two interrupt lines can be enabled/disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

The input data transfer interrupt lines to the Host, FlexRay input queue empty interrupt, FlexRay input queue full interrupt, are controlled by the enabled interrupts in FLXAnFRITS. In addition each of the input data transfer interrupts can be enabled/disabled separately by programming the related bits in FLXAnFRITC.

The output data transfer interrupt lines to the Host, FlexRay FIFO transfer warning interrupt, FlexRay output transfer warning interrupt, FlexRay FIFO transfer interrupt, FlexRay output transfer interrupt, are controlled by the enabled interrupts in FLXAnFROTS. In addition each of the output data transfer interrupts can be enabled/disabled separately by programming the related bits in FLXAnFROTC.

The three timer interrupts lines to the Host are controlled by the enabled interrupts in FLXAnFROS. In addition each of the interrupt lines can be enabled/disabled separately by programming bit FLXAnFROC.T0IE, FLXAnFROC.T1IE and FLXAnFROC.T2IE.

When a transfer between IBF/OBF and the Message RAM has completed bit FLXAnFRSIR.TIBC or FLXAnFRSIR.TOBC is set to 1.

A stop watch event may be triggered via input pin FLXAnSTPWT.

25.3.15 Assignment of FlexRay Configuration Parameters

Table 25.121 FlexRay Configuration Parameters

Parameter	Bit (Field)
pKeySlotusedForStartup	FLXAnFRSUCC1.TXST
pKeySlotUsedForSync	FLXAnFRSUCC1.TXSY
gColdStartAttempts	FLXAnFRSUCC1.CSA
pAllowPassiveToActive	FLXAnFRSUCC1.PTA
pWakeupChannel	FLXAnFRSUCC1.WUCS
pSingleSlotEnabled	FLXAnFRSUCC1.TSM
pAllowHaltDueToClock	FLXAnFRSUCC1.HCSE
pChannels	FLXAnFRSUCC1.CCH
pdListenTimeOut	FLXAnFRSUCC2.LT
gListenNoise	FLXAnFRSUCC2.LTN
gMaxWithoutClockCorrectionPassive	FLXAnFRSUCC3.WCP
gMaxWithoutClockCorrectionFatal	FLXAnFRSUCC3.WCF
gNetworkManagementVectorLength	FLXAnFRNEMC.NML
gdTSSTransmitter	FLXAnFRPRTC1.TSST
gdCASRxLowMax	FLXAnFRPRTC1.CASM
gdSampleClockPeriod	FLXAnFRPRTC1.BRP
pSamplesPerMicrotick	FLXAnFRPRTC1.BRP
gdWakeupSymbolRxWindow	FLXAnFRPRTC1.RXW
pWakeupPattern	FLXAnFRPRTC1.RWP
gdWakeupSymbolRxIdle	FLXAnFRPRTC2.RXI
gdWakeupSymbolRxLow	FLXAnFRPRTC2.RXL
gdWakeupSymbolTxIdle	FLXAnFRPRTC2.TXI
gdWakeupSymbolTxLow	FLXAnFRPRTC2.TXL
gPayloadLengthStatic	FLXAnFRMHDC.SFDL
pLatestTx	FLXAnFRMHDC.SLT
pMicroPerCycle	FLXAnFRGTUC1.UT
gMacroPerCycle	FLXAnFRGTUC2.MPC
gSyncNodeMax	FLXAnFRGTUC2.SNM
pMicroInitialOffset[A]	FLXAnFRGTUC3.UIOA
pMicroInitialOffset[B]	FLXAnFRGTUC3.UIOB
pMacroInitialOffset[A]	FLXAnFRGTUC3.MIOA
pMacroInitialOffset[B]	FLXAnFRGTUC3.MIOB
gdNIT	FLXAnFRGTUC4.NIT
gOffsetCorrectionStart	FLXAnFRGTUC4.OCS
pDelayCompensation[A]	FLXAnFRGTUC5.DCA
pDelayCompensation[B]	FLXAnFRGTUC5.DCB
pClusterDriftDamping	FLXAnFRGTUC5.CDD
pDecodingCorrection	FLXAnFRGTUC5.DEC
pdAcceptedStartupRange	FLXAnFRGTUC6.ASR
pdMaxDrift	FLXAnFRGTUC6.MOD
gdStaticSlot	FLXAnFRGTUC7.SSL
gNumberOfStaticSlots	FLXAnFRGTUC7.NSS
gdMinislot	FLXAnFRGTUC8.MSL
gNumberOfMinislots	FLXAnFRGTUC8.NMS

Table 25.121 FlexRay Configuration Parameters

Parameter	Bit (Field)
gdActionPointOffset	FLXAnFRGTUC9.APO
gdMinislotActionPointOffset	FLXAnFRGTUC9.MAPO
gdDynamicSlotIdlePhase	FLXAnFRGTUC9.DSI
pOffsetCorrectionOut	FLXAnFRGTUC10.MOC
pRateCorrectionOut	FLXAnFRGTUC10.MRC
pExternOffsetCorrection	FLXAnFRGTUC11.EOC
pExternRateCorrection	FLXAnFRGTUC11.ERC

25.3.16 Usage of Data Transfer

A mechanism is implemented to allow storage of FlexRay messages directly into the local RAM/global RAM (user RAM) and have transfers between the FlexRay internal message RAM and the local RAM/global RAM and vice versa with minimum CPU support. The data in the local RAM/global RAM should be indexed by data structure pointers located in data pointer tables stored in the local RAM/global RAM.

Data transfer from the local RAM/global RAM to the FlexRay internal message RAM (input transfer) needs to be initiated by the application. These transfers can be used to configure message buffers or to update transmit data.

A data transfer from the FlexRay internal message RAM to the local RAM/global RAM (output transfer) is initiated automatically by a reception into a receive message buffer or FlexRay internal FIFO or by a change in the slot status. It can be initiated also by a specific user transfer request.

The input and output data transfer can be activated independently. When the input data transfer is activated the application should not directly access message buffers using the FlexRay input buffer. When the output data transfer is activated the application should not directly access message buffers using the FlexRay output buffer.

25.3.16.1 Input Data Transfer

When the automatic input data transfer function is enabled, committed input data structures are transferred from the local RAM/global RAM to the FlexRay internal message RAM with minimum CPU support.

(1) Activation and Deactivation

The input data transfer function should be activated before usage. The activation of the input transfer handler initializes the input queue put index (FLXAnFRITS.IPIDX) and get index (FLXAnFRITS.IGIDX) to 0. Also the interrupt status flags in the FLXAnFRITS register (IQEIS and IQFIS) are set to 0.

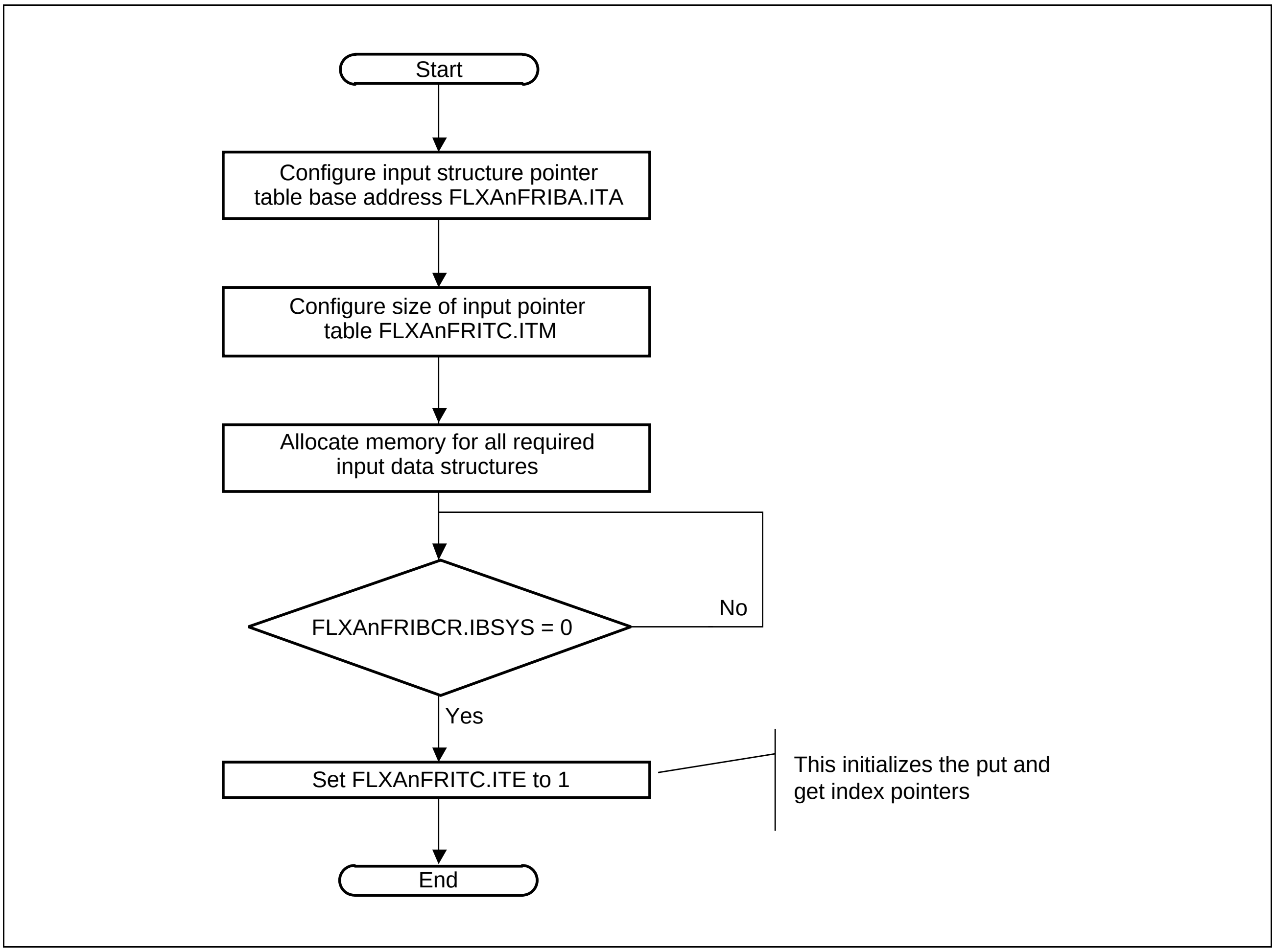


Figure 25.20 Input Transfer Enable Flow

A deactivation request of the input transfer function can be made at any time. The input queue put index and the input queue status are maintained independently from the input transfer function state.

Before the transfer function gets disabled (status indicated by FLXAnFRITS.ITS), user requested input transfers and all committed input transfers will be completed.

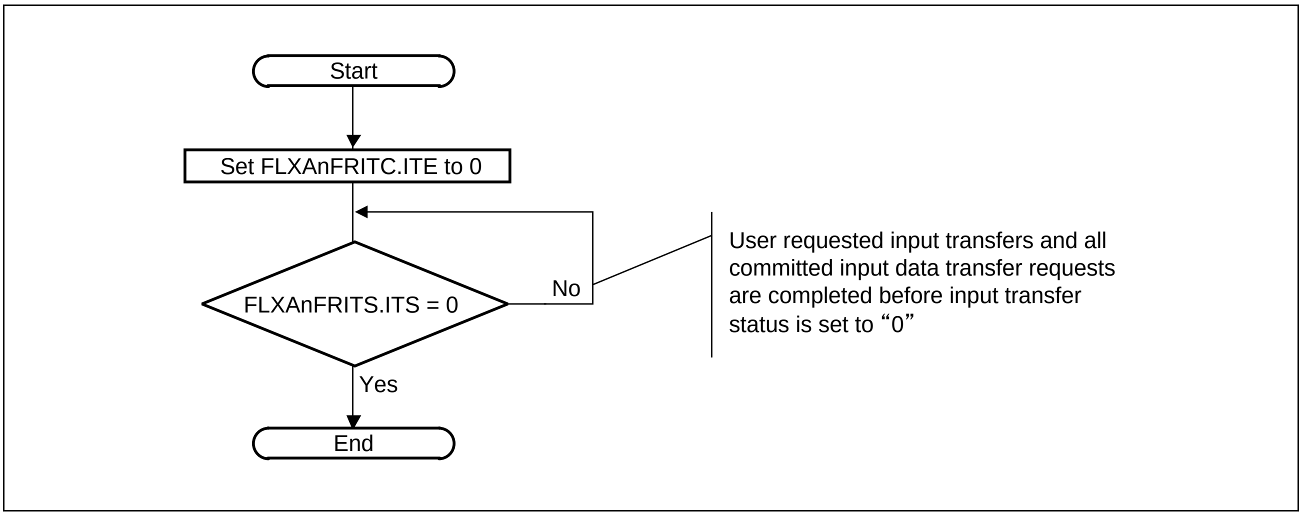


Figure 25.21 Input Transfer Disable Flow

(2) Input Data Structure

The application has to reserve a location in the local RAM/global RAM to provide the content for message buffer configuration (input data structure).

The location of this input data structure needs to be defined by an input data structure pointer also located in the local RAM/global RAM.

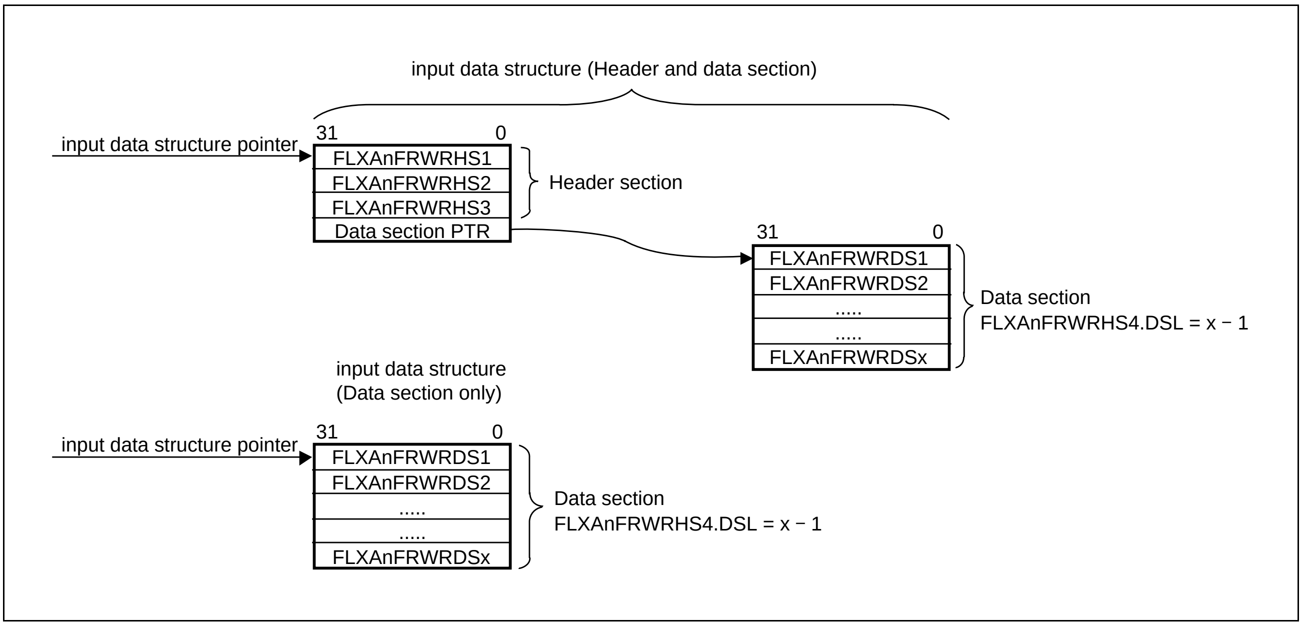


Figure 25.22 Input Data Structure

In general the input data structure consists of two sections, the header and the data section.

The header section consists of FLXAnFRWRHS1, FLXAnFRWRHS2, FLXAnFRWRHS3 and the data section pointer.

For bit alignment and bit function in the header section, see **Section 25.3.13.1, Header Partition**.

Depending on the settings in the control field (FLXAnFRWRHS4) located in the input pointer table, the data structure pointer is a reference to the address of FLXAnFRWRHS1 or FLXAnFRWRDS1. The data structure pointer has to be aligned to a 32 bit address.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to 1 it is required to provide a valid header section. In this case FLXAnFRWRDS1 is the first element of the data structure.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to 0 a header section is not required. In this case FLXAnFRWRDS1 is the first element of the data structure.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to 1 it is required to provide a valid data section. The pointer to the data section is a reference to the address of the first payload long word (FLXAnFRWRDS1) and has to be aligned to a 32 bit address.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to 0 a data section is not required. The data section pointer is not evaluated by the input handler.

The byte order for the FlexRay payload data in the input data structure is determined by FLXAnFROC.BEC. For information about the payload data alignment within the data section refer to **Section 25.3.13.2, Data Partition** and **Section 25.3.17, Byte Alignment**.

The length of the data section and the size to be allocated in the local RAM/global RAM depends on the configuration of the bits DSL in the address related to FLXAnFRWRHS4.

For the transfer into the FlexRay core internal message RAM the number of 16 bit words configured by FLXAnFRWRHS2.PLC is used. The application has to ensure, that a proper number of data words is provided in the local RAM/global RAM. In case the buffer is configured by FLXAnFRWRHS2.PLC to hold an odd payload length, the application has to write zero to the last 16 bit of the payload section to ensure that the padding data is all zero.

(3) Input Pointer Table

To transfer data from the input data structures located in the local RAM/global RAM to the FlexRay internal message RAM the related input data structure pointer and control field needs to be added to the input pointer table which is located in the local RAM/global RAM.

The location of the first element of this table is identified by the input pointer table base address (FLXAnFRIBA.ITA). This base address has to be aligned to a 32 bit address.

The maximum number of input requests that can be queued is defined by the Input queue Table Max register (FLXAnFRITC.ITM).

Each Input pointer table entry requires two long words. The required address range of the input pointer table for the queued transfer requests can be calculated by

Input pointer table size (byte) = (((FLXAnFRITC.ITM + 1) × 2) × 4)

Equation 1

The input pointer entry for the user requested input transfer should be added after the end of the input pointer table.

The pointer table index related to this entry and hence the number to be written to FLXAnFRUIR.UIDX, is FLXAnFRITC.ITM+1. The address in the input pointer table related to the user requested input transfer (user input address) can be calculated by

User input address = FLXAnFRIBA.ITA + Input pointer table size

Equation 2

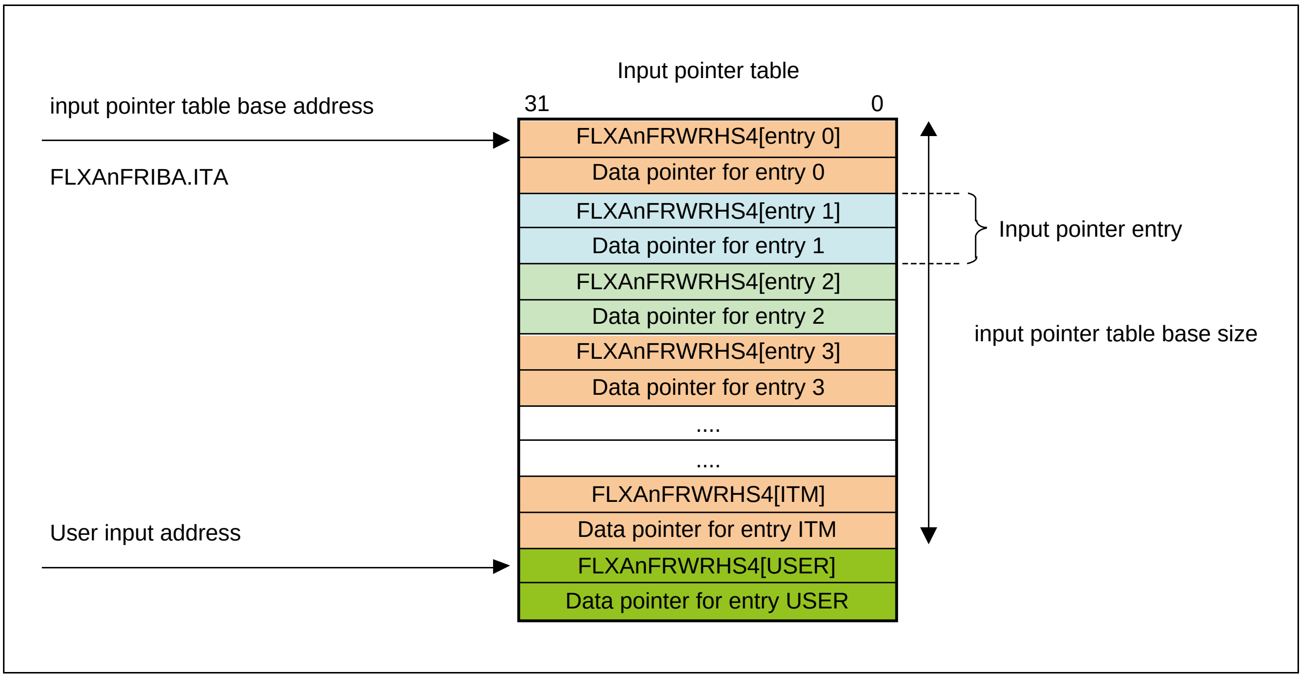


Figure 25.23 Input Pointer Table

The input pointer table holds the control field FLXAnFRWRHS4 and the pointers to the local RAM/global RAM location where the message buffer content (header section and/or data section) is stored.

The application has to write FLXAnFRWRHS4 and the input data structure pointer at the addresses in the input pointer table related to the put index position before a transfer request is initiated.

FLXAnFRWRHS4:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	DSL[5:0]					

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	INV	STR	LDS	LHS	—	IMBNR[6:0]						

Table 25.122 FLXAnFRWRHS4 Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, an undefined value is read. When writing, always write 0.
21 to 16	DSL[5:0]	Data Section Length Bit Specifies the length of the data section in terms of 32 bit values.
15 to 12	Reserved	When read, an undefined value is read. When writing, always write 0.
11	INV	Invalidate entry Bit 0: The data structure is valid and will be transferred to the FlexRay internal message RAM. 1: The data structure is invalid. FlexRay internal message RAM is not updated using this input pointer entry.
10	STR	Set transmission request Bit 0: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to 0. No data from this message buffer is transmitted. 1: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to 1 to release the message buffer for transmission The application should not set the bit STR to 1 for receive buffers.
9	LDS	Load data section Bit 0: No update of data section. 1: Data section for the message buffer selected by the bits IMBNR is updated.
8	LHS	Load header section Bit 0: No update of header section. 1: Header section for the message buffer selected by the bits IMBNR is updated.
7	Reserved	When read, an undefined value is read. When writing, always write 0.
6 to 0	IMBNR[6:0]	Message buffer number to be updated Bit Selects the target message buffer number in the FlexRay internal message RAM for transfer

Note that the LHS bit should not be set for protected message buffers.

The bit LDS defines if the data section of the message buffer selected by the bits IMBNR should be updated.

If LDS is set to 1 (DSL + 1) 32 bit words of payload data are transferred from the local RAM/global RAM to the message buffer selected by the bits IMBNR.

If LDS is set to 0 no payload data is transferred from the local RAM/global RAM.

Note that the payload transferred is independent from the configured payload length (bits PLC in the address related to FLXAnFRWRHS2).

The bit INV can be used to invalidate a committed data structure. This bit should be only used to cancel the transfer of committed data structures when the input queue is halted (see **Section 25.3.16.2(5), Transfer Function of Dedicated Message Buffers**).

When this bit is set to 1 the message buffer number IMBNR is not updated. When the bit is set to 0 the message buffer number IMBNR is updated.

(4) Transfer Function of Input Data Structure

To use the input data structure transfer function the input transfer has to be activated (see **Section 25.3.16.1(1), Activation and Deactivation**). The activation process requires the setup of the input pointer table (see **Section 25.3.16.1(3), Input Pointer Table**) in order to specify the source location (input data structures) for the data structures to be transferred. When the input transfer gets enabled the get index pointer is initialized to zero.

All FlexRay internal message buffers can be updated using the input transfer queue which is built in the input pointer table. The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. For that purpose the application has to maintain a put index for the input pointer table that indicates where the pointer has to be written to.

To commit this table entry to the input handler, the application has to write the target message buffer number to the input queue control register (FLXAnFRIQC.IMBNR). Afterwards the application has to increment the application internal put index.

By writing to the input queue control register the data available flag (FLXAnFRDAi.DA[IMBNR]) is automatically set to 1. The input transfer handler also maintains the put index pointer in the status register (FLXAnFRITS.IPIDX).

In case the input queue gets full (number of queued input transfer requests is equal to the input queue table size) FLXAnFRITS.IQFP and FLXAnFRITS.IQFIS are set to 1. The input queue full condition pending flag (FLXAnFRITS.IQFP) changes from 1 to 0 when there are entries in the input queue available, whereby the input queue full interrupt status flag (FLXAnFRITS.IQFIS) needs to be cleared by the application.

The application cannot make any further write access to the bit IMBNR in the FLXAnFRIQC register as long as the bit IQFP in the FLXAnFRITS register is 1.

In case the input queue gets empty (number of queued input transfer requests changes to zero) FLXAnFRITS.IQEIS is set to 1. The input queue empty interrupt status flag (FLXAnFRITS.IQEIS) needs to be cleared by the application.

The transfer of the input data structures to the FlexRay message RAM is controlled by a get index pointer which is handled inside the FlexRay module and flagged in FLXAnFRITS.IGIDX. Note that the index is referring to the input entry and not the address offset in the input pointer table.

If the input queue is not empty, the transfer handler reads out the input pointer table entry of the transfer queue and starts the transfer of the input data structure from the address the input pointer is referring to. When all required data words are transferred to the FlexRay module, the data available flag for the transferred message buffer number is set to 0 and the get index in the transfer handler is incremented by one.

In case of an invalidated data structure (see **Section 25.3.16.1(5), Halting the Input Queue**) no FlexRay internal message buffer is updated and the related data available flag is automatically set to 0. The change of the data available flag can be used to confirm the cancellation a transmit request.

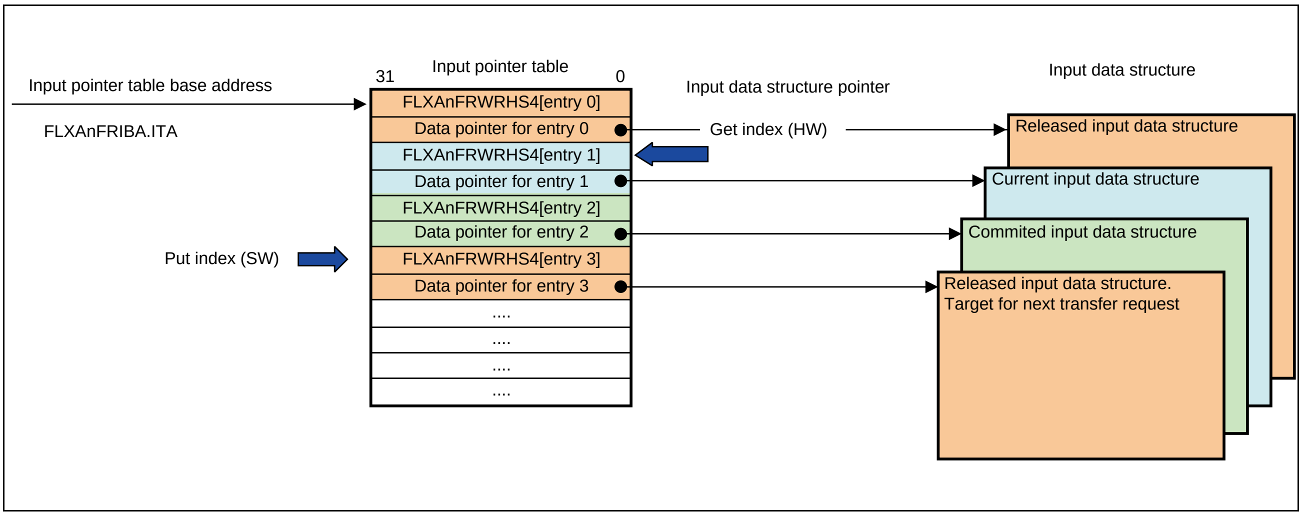


Figure 25.24 Input Pointer Table

Receive message buffers can be also configured using the input data transfer by setting up the required header sections and mark only the header section (FLXAnFRWRHS4.LDS = 0, FLXAnFRWRHS4.LHS = 1) to be updated in the FlexRay module.

(5) Halting the Input Queue

Committed data structures cannot be removed, but can be invalidated or updated when the input queue is halted.

To cancel data structures already committed to the input queue, the queue can be halted by writing a 1 to FLXAnFRITC.IQHR.

After the ongoing input transfer has been completed the queue is halted and FLXAnFRITS.IQH changes from 0 to 1.

To invalidate an entry of the input queue FLXAnFRWRHS4.INV has to be set to 1. All other bits in FWRHS4 should be unchanged.

Following flow shall be used to analyze whether a committed message has been already transferred to the FlexRay internal message RAM or not.

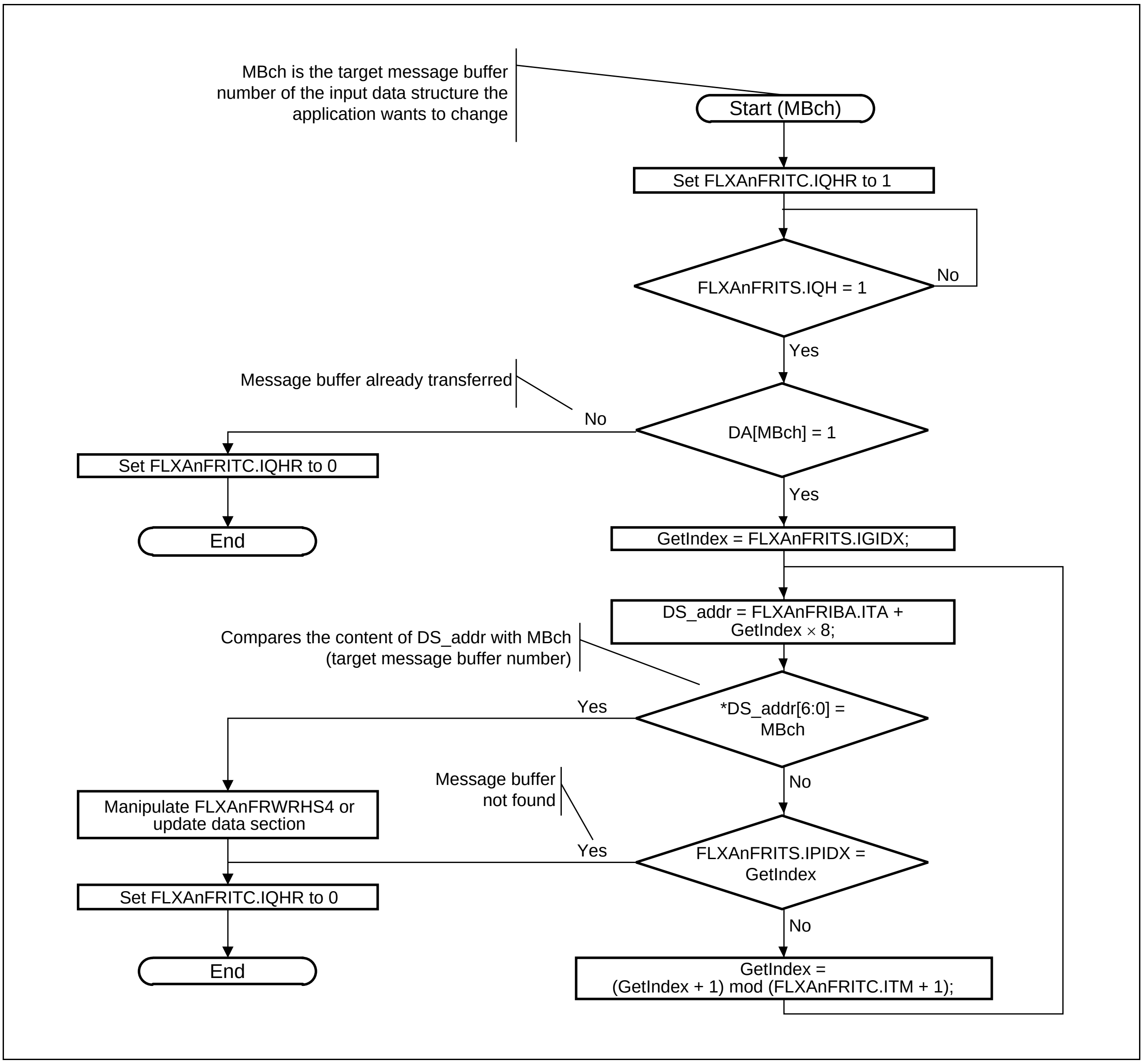


Figure 25.25 Input Table Analysis

In case the message buffer was already transferred to the FlexRay internal message RAM, the user input transfer request can be used to bypass the actual queue and update the required message buffer (see **Section 25.3.16.1(6), Transfer Function of User Requested Input Transfers**).

(6) Transfer Function of User Requested Input Transfers

To use this function the input transfer has to be activated (see **Section 25.3.16.1(1), Activation and Deactivation**).

The application is capable, by using FLXAnFRUIR.UIDX, to request a transfer of an input data structure. The user input transfer request is serviced first.

The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. The table entry for the user input transfer request should be added after the end of the input pointer table (see **Section 25.3.16.1(3), Input Pointer Table**).

To commit this table entry to the input handler, the application has to write the index (FLXAnFRITC.ITM + 1) to the user input transfer request register (FLXAnFRUIR.UIDX).

By writing to the user input transfer request register, the user input transfer request pending flag (FLXAnFRITS.UIRP) is automatically set to 1.

As long this flag is 1 the application should not make any further user input transfer requests.

The user input transfer request pending flag (FLXAnFRITS.UIRP) changes from 1 to 0 when the requested input transfer is completed. As next the pending transfers are processed.

25.3.16.2 Output Data Transfer

When the output data transfer function is enabled, received messages (either in dedicated message buffers or in the FlexRay receive FIFO) are transferred to the local RAM/global RAM by the output data handler. The output data handler can also transfer the message buffer content to the local RAM/global RAM on application request. When enabled the output handler is also capable to initiate a transfer when the message buffer status has changed.

(1) Activation and Deactivation

The output data transfer function should be activated before usage. The activation of the output transfer hander will initialize the FIFO put and get index pointer and FIFO fill level (FLXAnFROTS.FGIDX and FLXAnFROTS.FFL) to zero, set the bits FLXAnFROTS.FDA, FLXAnFROTS.OWP, FLXAnFROTS.FWP and FLXAnFROTS.UORP to 0. Also the interrupt status flags (FLXAnFROTS.OTIS, FLXAnFROTS.FIS, FLXAnFROTS.OWIS and FLXAnFROTS.FWIS) are set to 0.

The activation has no influence to the data available flags (FLXAnFRDAi.DA) which are related to the dedicated buffers; these flags have to be cleared by the application.

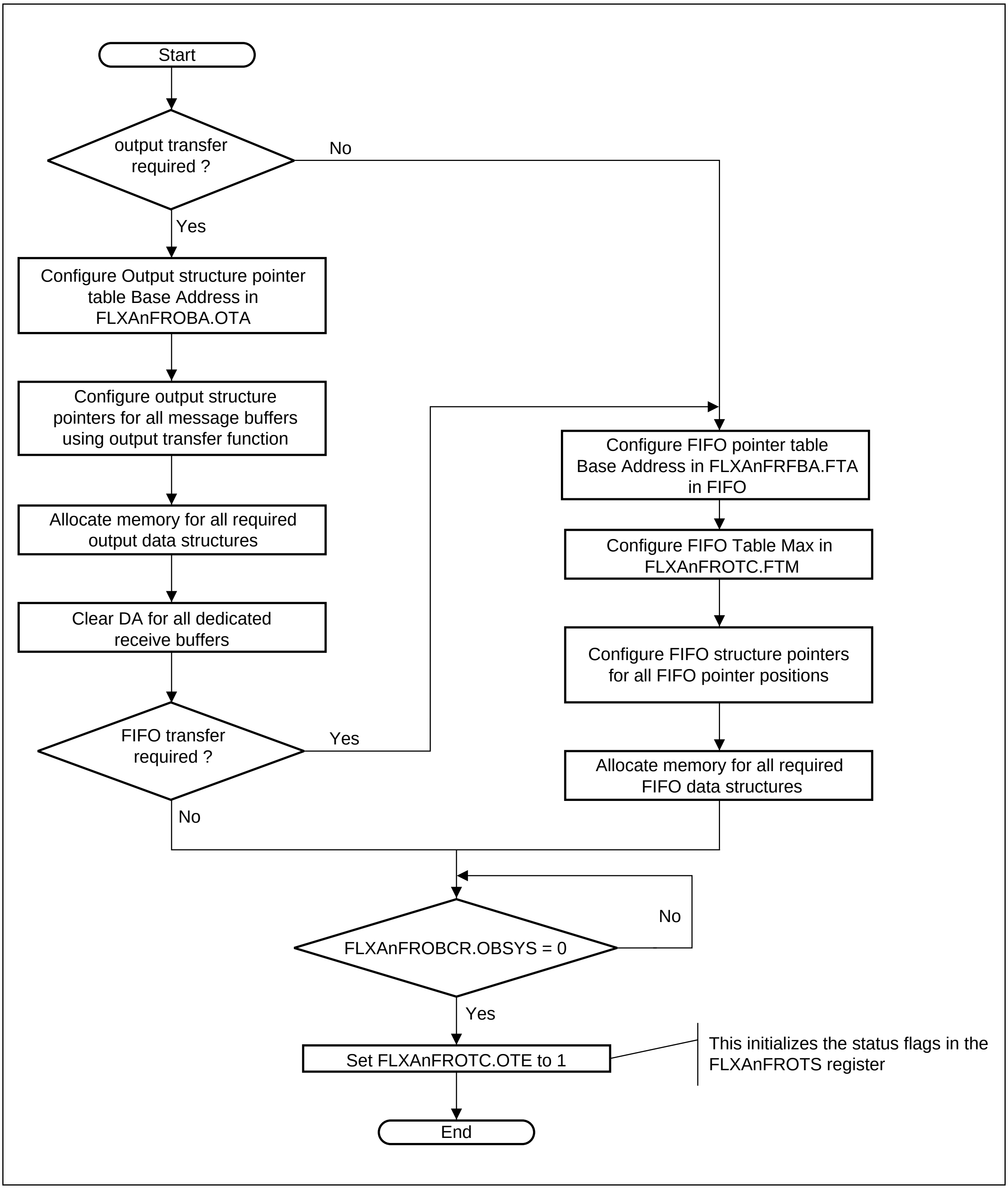


Figure 25.26 Output Transfer Enable Flow

A deactivation request of the output data transfer function can be made at any time. An ongoing transfer will be completed and the completion of this transfer will be flagged. During this time FLXAnFROTS.OTS remains 1.

When FLXAnFROTS.OTS changes from 1 to 0, the output transfer function is deactivated. The data available status flags and the FIFO get index are still maintained when the output transfer function is disabled.

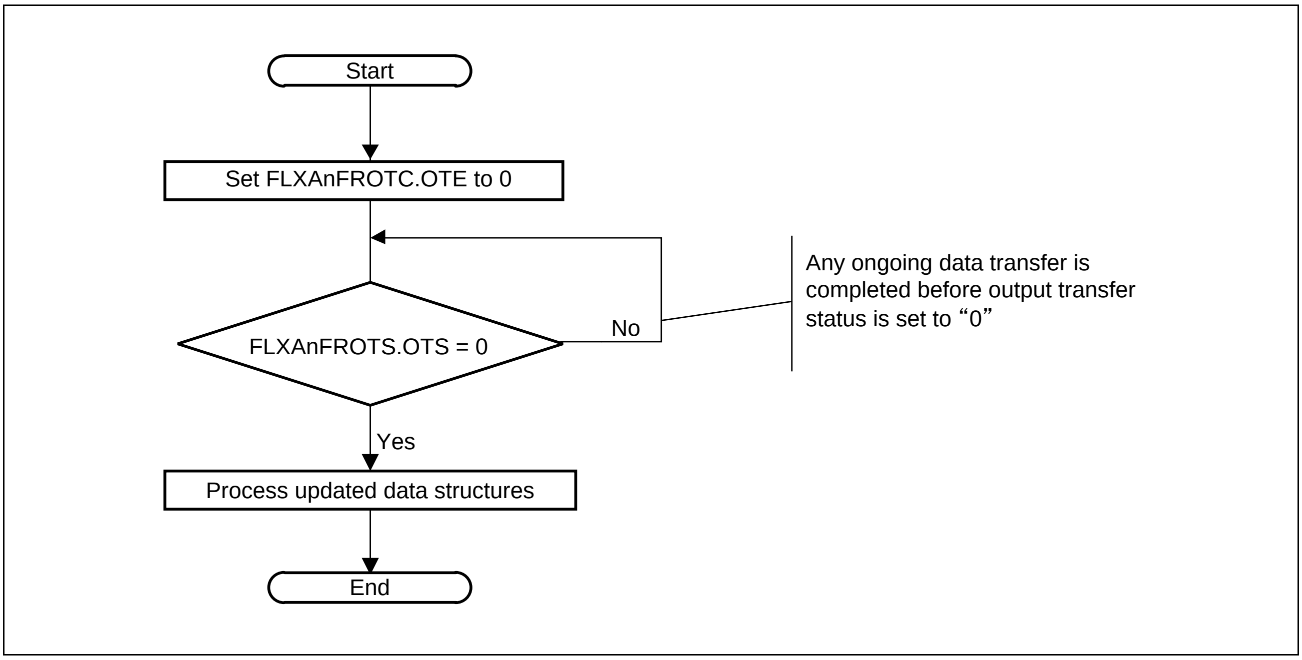


Figure 25.27 Output Transfer Disable Flow

(2) Output Transfer Data Structure

The data in the local RAM/global RAM is stored in an output data structure. The location of the output data structures are determined by output data structure pointers also located in the local RAM/global RAM. The output data structure and indexing is visualized in **Figure 25.28, Output Data Structure**.

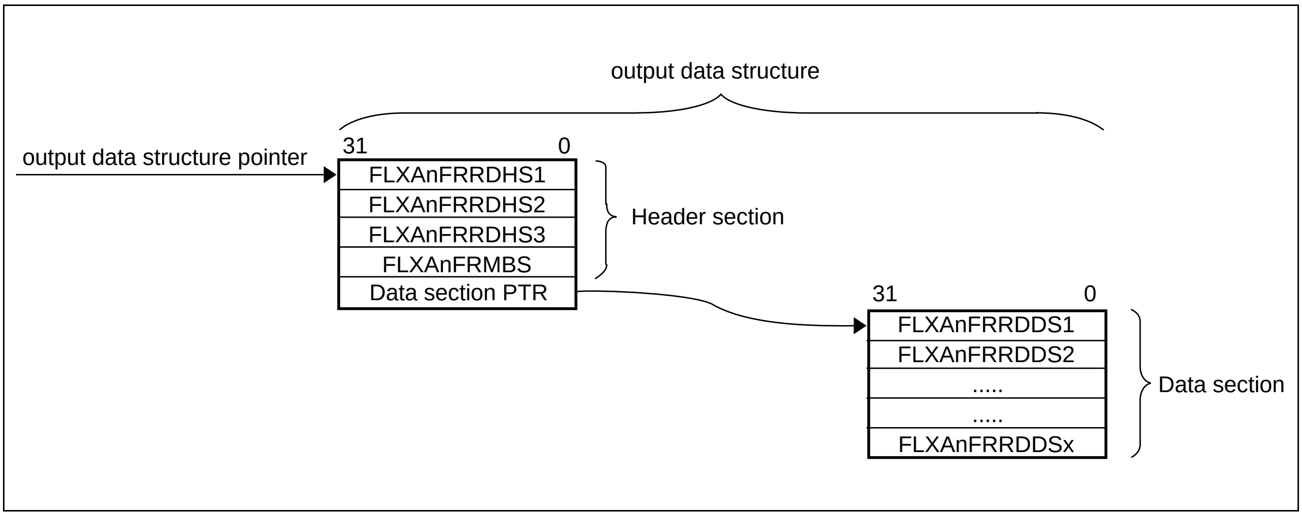


Figure 25.28 Output Data Structure

The output data structure consists of two sections, the header and data section. The header section consists of FLXAnFRRDHS1, FLXAnFRRDHS2, FLXAnFRRDHS3, FLXAnFRMBS and the data section pointer. FLXAnFRRDHS1 is the first element of the structure and has to be aligned to a 32 bit address. The data structure pointer is a reference to the address of FLXAnFRRDHS1. For information about the bit alignment and bit function within the header section refer to **Section 25.3.13.1, Header Partition**.

FLXAnFRRDDS1 is the first element of the data section. The data section pointer is a reference to the address of FLXAnFRRDDS1 and has to be aligned to a 32 bit address. The byte order for the FlexRay payload data in the output data structure is determined by the bit BEC in the FLXAnFROC register. For information about the payload data alignment within the data section refer to **Section 25.3.13.2, Data Partition**.

The length of the data section as well as the total structure size to be allocated in the local RAM/global RAM depends on the configured payload length (FLXAnFRRDHS2.PLC) of the related message buffer. In case the configured payload length is an odd number of words or the received payload length (FLXAnFRRDHS2.PLR) is smaller than the configured payload length, the remaining data words in the local RAM/global RAM are unused and cannot be used by the application.

The output data structure is identical for all three kinds of output transfers. In case only the header section is transferred the data section pointer is not evaluated by the output handler and the data section remains unchanged.

(3) Output Pointer Table

For the output data transfer function the application needs to set up an output pointer table in the local RAM/global RAM. The location of the first element of this table should be programmed into the output pointer table base address (FLXAnFROBA.OTA). This base address has to be aligned to a 32 bit address.

The size of the output pointer table is defined by the maximum of: the last configured dedicated message buffer and the highest message buffer number which will be used for the user output transfer request.

The output pointer table holds pointers (output data structure pointers) to the local RAM/global RAM location where a memory space is reserved for the target message buffer content (header section and data section).

There is a fixed linear relationship between the address of the entries in the output pointer table and the number of the related message buffers (see **Figure 25.29, Output Data Structure and Indexing**): the output pointer table starts with the entry for message buffer number 0 at the address configured in FLXAnFROBA.OTA and continues in ascending order for each following message buffer number, by 32 bit aligned address (e.g. message buffer 1 at address OTA + 4, message buffer 2 at address OTA + 8, etc.) for all possible message buffers.

When a set ND bit is the only transfer condition (FLXAnFROTC.OTCS is set to 0) only message buffers configured as a dedicated receive buffer or that will be used for user output transfer requests need have valid pointer entries.

When a set ND bit or a set MBC bit is the transfer condition (FLXAnFROTC.OTCS is set to 1) all dedicated receive buffer and dedicated transmit buffers need to have valid pointer entries.

(4) FIFO Output Pointer Table

The FlexRay module internal FIFO can be extended by a queued buffer structure in the local RAM/global RAM.

If the FlexRay module internal FIFO is used the application needs to setup the FIFO output pointer table. The location of the first element of this table is identified by the FIFO pointer table base address (FLXAnFRFBA.FTA). This base address has to be aligned to a 32 bit address.

The size of the FIFO pointer table and hence the maximum number of messages that can be added to the queue, is defined by FIFO Table Max (FLXAnFROTC.FTM).

The FIFO pointer table holds pointers (output data structure pointers) to the local RAM/global RAM location where a memory space is reserved the target message buffer content (header section and data section). For each table entry a data pointer shall be configured in this table.

(5) Transfer Function of Dedicated Message Buffers

To use this transfer function the output transfer has to be activated (see **Section 25.3.16.2(1), Activation and Deactivation**). The activation process requires to setup the output pointer table (see **Section 25.3.16.2(3), Output Pointer Table**) in order to specify the destination location (output data structures) for the data to transfer.
Figure 25.29, Output Data Structure and Indexing shows how the output pointer table references the output data structures.

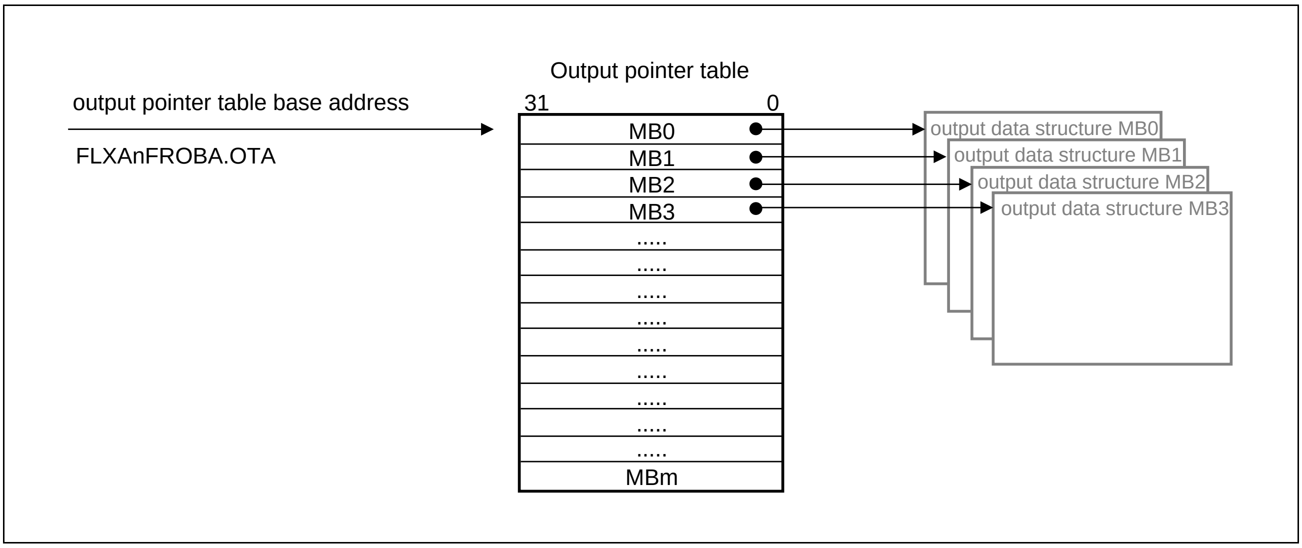


Figure 25.29 Output Data Structure and Indexing

With FLXAnFROTC.OTCS the output transfer condition can be selected between the “New data only mode” and the “New data and status changed mode”.

In the “New data only mode” an output data transfer is initiated when a valid FlexRay data frame has been stored into a dedicated receive buffer which causes the related ND flag in the FLXAnFRNDATi register to set. The ND flag in the FLXAnFRNDATi register is automatically set to 0 during the transfer procedure. The header section is also transferred and hence the MBC flag in the FLXAnFRMBSCi register is set to 0.

In the “New data and status changed mode” an output data transfer is initiated as described in the “New data only mode”. In addition an output data transfer is initiated when only the message buffer status has been changed which causes the related MBC flag in the FLXAnFRMBSCi register to be set. In this case only the header section is transferred. The MBC flag in the FLXAnFRMBSCi register is automatically set to 0 during the transfer procedure.

After transferring the message buffer data from the FlexRay internal message RAM to the output data structure the corresponding data available flag in the FLXAnFRDAi (i = 0 to 3) registers is set to 1. The update of the output data structure is also flagged by the setting of the output transfer interrupt status flag (FLXAnFROTS.OTIS).

As long as the data available flag remains 1 the corresponding output data structure will not be updated. In the case

- the data available flag is 1 and a valid received message was stored or
- when FLXAnFROTC.OTCS is 1 and the message buffer status was updated,

the output transfer warning interrupt flag (FLXAnFROTS.OWIS) is set to 1 notifying the application that new data is available but the output data structure transfer cannot be processed. In addition FLXAnFROTS.OWP is set to 1 that continuously flags that status of the output transfer warning condition.

If a valid receive message in the FlexRay internal message RAM is overwritten by an additional receive message, the message lost flag (FLXAnFRMBS.MLST) is set to 1. This flag can be evaluated after the message buffer has been transferred into an output data structure.

Following sections are giving a guidance how output data structures can be handled.

(a) Data Section Copy Method

One option is to copy the information from the output data structure to a different location of the local RAM/global RAM and then release the output data structure by clearing the related data available flag. The application should use the copied information for further processing.

(b) Data Structure Pointer Method

A different option is to modify the output data structure pointer in the output pointer table and to release the output data structure by clearing the related data available flag. The changed output data pointer should refer to a free data structure. The application should use the old data structure for further processing.

(c) Data Section Pointer Method

A third option is to modify the data section pointer in the output data structure and to release the output data structure by clearing the related data available flag. The changed data section pointer should refer to a free memory area. The application should use the old data section for further processing by forwarding the data section pointer.

(6) Transfer Function of FIFO Message Buffers

To use this buffer transfer function the output transfer has to be activated (see **Section 25.3.16.2(1), Activation and Deactivation**). The activation process requires the set up of the FIFO pointer table (see **Section 25.3.16.2(4), FIFO Output Pointer Table**) in order to specify a location in the local RAM/global RAM reserved for the storage of the required output data structures.

A FIFO data transfer is initiated when a valid FlexRay data frame has been stored in the FlexRay internal FIFO.

After transfers from the internal FIFO to the output data structure, the FIFO interrupt status flag (FLXAnFROTS.FIS) and FLXAnFROTS.FDA are set to 1. The bit FLXAnFROTS.FIS can be used as an interrupt source. The bit FLXAnFROTS.FDA indicates that the FIFO is not empty.

Up to FLXAnFROTS.FTM output structures configured in the FLXAnFROTC register can be queued.

The transfer to the extended FIFO buffer structure is controlled by index pointers. This put index is controlled by the FIFO transfer handler and is incremented after transferring a message to the output data structure.

The FIFO reception handler also maintains a get index which is flagged in FLXAnFROTS.FGIDX. The value of this get index is known by the application by either reading the status or maintaining a software variable. The get index (the value after a reset is zero) is incremented by one when the application releases the oldest entry of the FIFO queue by writing 1 to FLXAnFROTS.FDA. By comparing the put index and the get index the FIFO handler knows about the current fill level of the queued buffer structure.

The current FIFO fill level is flagged in FLXAnFROTS.FFL. When FLXAnFROTS.FDA is 1, there is at least one entry in the FIFO queue.

In case the queued buffer structure in the local RAM/global RAM is full (FLXAnFROTS.FFL = FLXAnFROTC.FTM + 1), no further transfers are initialized, new messages remain in the FlexRay internal FIFO and the FIFO transfer warning interrupt status flag (FLXAnFROTS.FWIS) is set to 1.

In case the FlexRay internal FIFO structure becomes full, messages in the FlexRay internal FIFO structure may get overwritten. The related status flags and configuration registers of the FlexRay core module can be used to generate desired warning notifications.

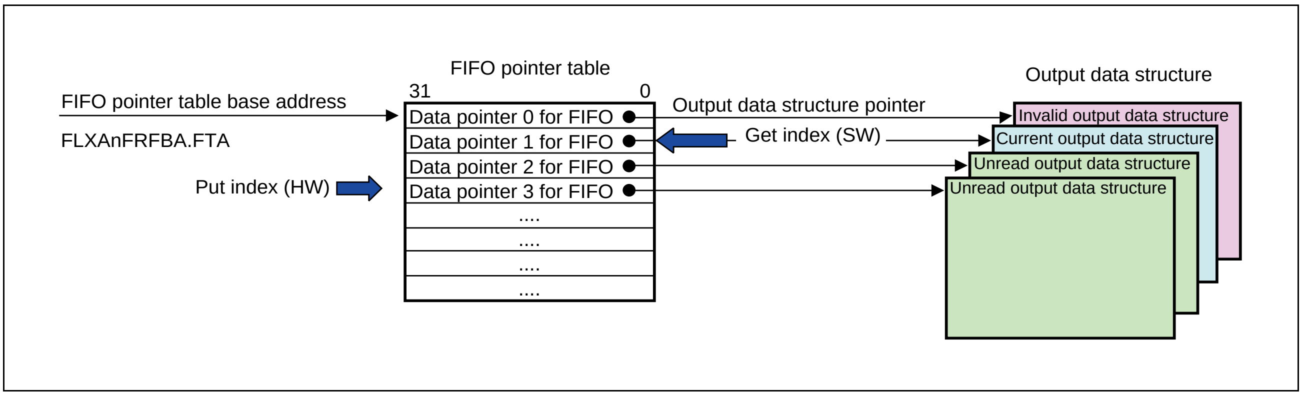


Figure 25.30 FIFO Pointer Table

(7) Transfer Function of User Output Transfer Requests

To use this transfer function the output transfer has to be activated (see **Section 25.3.16.2(1), Activation and Deactivation**). The activation process requires to setup the output pointer table (see **Section 25.3.16.2(3), Output Pointer Table**) in order to specify the location in the local RAM/global RAM reserved for the transfer of the data (output data structures).

The application is capable, by using FLXAnFRUOR.UMBNR, to request a transfer of dedicated message buffer to an output data structure. Except in CONFIG state, message buffers which are part of the FlexRay internal FIFO should not be requested.

The header section is always transferred to the output data structure. The transfer of the data section can be enabled by setting FLXAnFRUOR.URDS to 1. The selected message buffer content is stored in the output data structure location determined by the pointers in the output pointer table.

The data available status and transfer blocking by the DA bits in the FLXAnFRDAi register is also used for the user requested transfers. Therefore the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) should be released before making the transfer request.

After writing to FLXAnFRUOR.UMBNR, the bit FLXAnFROTS.UORP is set to 1 to indicate that there is a pending user transfer request. When the transfer has been processed the bit FLXAnFROTS.UORP is set to 0, the bit FLXAnFROTS.OTIS is set to 1 and the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) is set to 1.

User output transfer requests cannot be queued. The application should check the bit FLXAnFROTS.UORP before writing to FLXAnFRUOR.UMBNR.

User output transfer requests should not be made for message buffers which are pending in the input transfer queue.

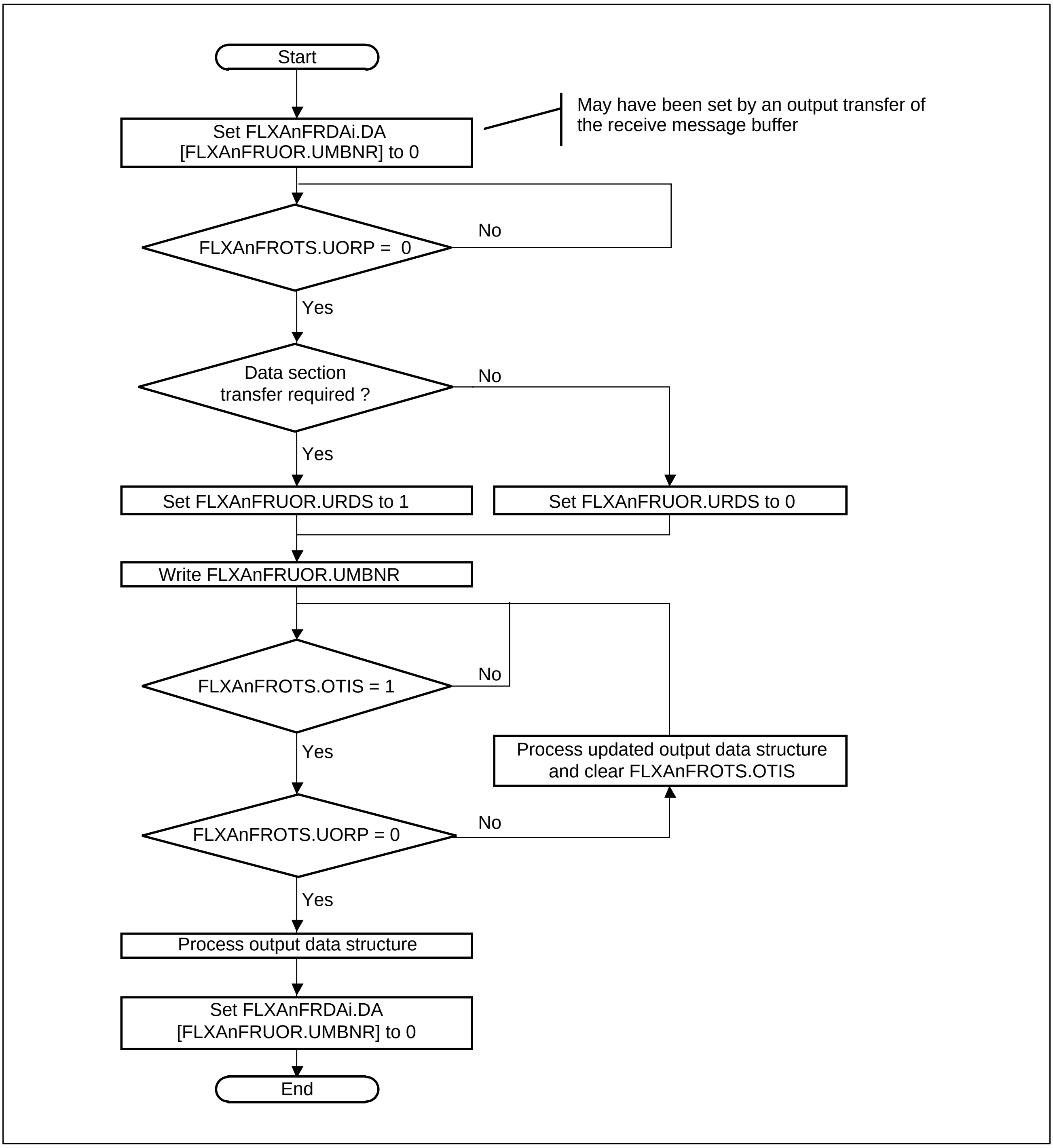


Figure 25.31 User Output Transfer Request Flow

Note that it may be possible that the data structure addressed by a user request is being updated due to a receive message buffer update (which causes the bit DA in the FLXAnFRDAi register being set).

This set DA flags inhibits the user output transfer request. Therefore polling FLXAnFROTS.UORP is not a secure method to identify when the transfer of a requested message buffer has been completed. The bits FLXAnFROTS.OTIS or the DA bit in the FLXAnFRDAi register can be used instead. The exact flow depends on the software architecture.

25.3.16.3 Data Structure Transfer Scheduling

Cyclically the different types of transfer requests are checked. In order to guarantee a certain transfer time the different types of transfers have different priorities.

Use requested input transfers have highest priority followed by the transfer of data structures committed into the active input transfer queue. No new output transfer will be started as long as there is a pending input transfer request.

The three output transfer request types are checked in a specific order:

(1) All Dedicated Message Buffers in Ascending Order

When FLXAnFROTC.OTCS is set to 0, set flags in the FLXAnFRNDATi register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is 0).

When FLXAnFROTC.OTCS is set to 1, set flags in the FLXAnFRNDATi register or set flags in the FLXAnFRMBSCi register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is 0).

(2) FlexRay Internal FIFO

When the FlexRay internal FIFO is not empty and there is a free destination area, one FIFO message is transferred into the output data structure addressed by the FIFO pointer table.

(3) User Output Request

If there is a pending user output transfer request, one message buffer is transferred into the corresponding output data structure.

The check sequence is suspended when an input transfer occurs.

25.3.16.4 Behavior in Case of Data Transfer Access Error

The memory areas accessed by the data transfer function may be protected by a memory protection unit (MPU). When the MPU flags an access to a protected address caused by an input or output transfer, an access error event is generated and the related bit in the FLXAnFRAES register is set.

The ongoing transfer is immediately terminated but succeeding transfers are processed and may generate further access errors. Any following access errors are only flagged in FLXAnFRAES.MAE. The other status flags are not updated.

(1) Access Error during Input Transfer

When an access error occurs during an input transfer:

- The ongoing transfer is immediately terminated. The FlexRay internal message RAM will not be updated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.IAE is set to 1
- The input pointer table index is flagged in FLXAnFRAES.EIDX
- In case of an normal input transfer, the related DA flag in the FLXAnFRDAi register is set to 0
- In case of an user input transfer, request FLXAnFRITS.UIRP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. In addition the application needs to clear the input access error flag (FLXAnFRAES.IAE).

(2) Access Error during Output Transfer

When an access error occurs during an output transfer:

- The ongoing transfer is immediately terminated but the update of the data structure may have started.
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.OAE is set to 1
- The output pointer table index is flagged in FLXAnFRAES.EIDX
- In case of an normal output transfer, the related DA flag in the FLXAnFRDAi register remains 0 and no output transfer, interrupt is generated
- In case of an user output transfer request FLXAnFROTS.UORP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. The data structure in the local RAM/global RAM cannot be treated as valid.

In addition the application needs to clear the output access error flag (FLXAnFRAES.OAE).

The FlexRay module internal transfer of the message buffer is completed before the local RAM/global RAM access error is detected. The output transfer will not be re-initiated. To avoid loss of data, the application can perform a user output transfer request of this message buffer to a correct local RAM/global RAM location.

(3) Access Error during FIFO Transfer

When an access error occurs during an FIFO transfer:

- The ongoing transfer is immediately terminated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.FAE is set to 1
- The FIFO pointer table index is flagged in FLXAnFRAES.EIDX
- The FIFO index pointer are not changed and hence the FIFO status flags are unchanged

With the given status information the application is able to identify and correct the faulty data structure. In addition the application needs to clear the FIFO access error flag (FLXAnFRAES.FAE).

The data in the local RAM/global RAM cannot be treated as valid and is not released to the application. The message cannot be recovered.

25.3.16.5 Behaviors in Case of RAM Read Errors

The FlexRay internal message RAM has an ECC checking mechanism. In case an uncorrectable RAM read error occurs, the application has to analyze the status in the FLXAnFRMHDS register and react as described in **Section 25.3.16.3, Data Structure Transfer Scheduling**. The input and output transfer handler reacts also on these errors detected in the message RAM when the error is related to an active transfer.

In addition, the Temporary Buffer RAM A and Temporary Buffer RAM B have an ECC checking mechanism as well. An uncorrectable RAM read errors does not impact the data transfer functionality but have to be handled as described in **Section 25.3.13.1(4), Message Buffer Status FLXAnFRMBS (Word 3)**.

In all cases, data causing a read error is never transferred to the local RAM/global RAM. If there is no recovery available in the application, the message is lost.

(1) Read Error during Transfer from TBF to MBF

This internal transfer is done for each valid FlexRay message received.

A read error can only occur when reading the header section in the FlexRay Message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

For dedicated receive message buffers, the related ND flag in the FLXAnFRNDATi register will not get set. Consequently the affected message buffer will not be transferred to the output data structure.

For the FlexRay internal FIFO buffers, the ND flag in the FLXAnFRNDATi register is not set but the FlexRay internal FIFO put index is incremented. Due to this, a transfer procedure from the FlexRay internal FIFO buffer to the output buffer is started. However, if the read error is still present in the header section, updating of the output data structure will not start (see **Section 25.3.16.5(2), Read Error during Transfer from MBF to OBF**); thus the data in the local RAM/global RAM remains correct.

Note that the correction or any other reconfiguration of FIFO related to message buffers while there are pending FIFO transfers may result in incorrect data in the local RAM/global RAM. It is strongly recommended to deactivate the output data transfer before starting the reconfiguration and flush the FlexRay internal FIFO before reactivation of the output data transfer.

(2) Read Error during Transfer from MBF to OBF

This internal transfer is done for every output data transfer (dedicated reception, FIFO, user requested).

A read error can occur in the header and data section (see read error flags in FLXAnFRMHDS). In both cases the message gets lost. If the error is located in the header section, the message buffer needs to be re-configured. If the error is located in the data section, the error is corrected with the next data section update.

When a read error occurs during the transfer from the message RAM to the output buffer, the output data structure will not be updated and the data available will not be set to 1. The FIFO put index and the FIFO fill level are not changed also.

In case of user output transfer request, FLXAnFROTS.UORP is set to 0 even if there was no update of the output data structure.

(3) Read Error during Transfer from IBF to MBF

This internal transfer is done for every input data transfer.

A read error can occur only when there is no update of the header section requested (the bit LHS in FLXAnFRWRHS4 is set to 0) due to the reading of the header section from the message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

When a read error occurs during the input data transfer, the actually transferred message in the input queue gets lost.

(4) Message RAM Read Errors

Read errors when reading the header section are flagged in the FLXAnFRMHDS register.

Depending on the buffer type and set buffer protection, a reconfiguration of the message buffer may not be possible.

The input transfer function cannot be used to reconfigure a locked message buffer using the method described in **Section 25.3.13.4(3), Temporary Unlocking of Header Section**.

Before reconfiguring a locked buffer, the user should disable the input transfer function and the output transfer function.

25.3.16.6 Data Transfer Timings

The transfer timing between the local RAM/global RAM and the FlexRay internal message buffers is composed of two elements. The first element is the FlexRay core internal transfer time between the interface buffer and the message RAM. The second element is the transfer time required between the local RAM/global RAM and the interface buffer.

The concept of the input- and output handler allows only one active transfer between the local RAM/global RAM and input buffer as well as only one active transfer between the input buffer and the message RAM. In case two requests are made at the same time, the input transfer task is handled at first.

Due to this concept constrains the calculation of transfer times does not consider a parallel transfer but the worst case scenario whereby an output transfer is started only after all input transfers are completed.

(1) FlexRay Core Internal Transfer Time

The number of clock cycles required for a transfer is given in FlexRay core module clock cycles (bus clock). The transfer time differs between the input and output transfer.

The time required for a transfer between the interface buffer and the FlexRay internal message RAM is given by [ERAY_ADD]

$$cycles_{req\_MHD[PL]} = (numberOfConcurrentTasks) \times \left(setupTime + \left(DataWords_{req[PL]}\right) + 75\right)$$

Equation 3

The number of data words (32 bit) can be calculated by

$$DataWords_{req[PL]} = 4 + floor\left(\frac{PLC + 1}{2}\right)$$

Equation 4

Whereby 4 is the number of header section words (32 bit) which needs to be always accessed even if the header section is not updated.

As there can be in total three parallel tasks in the FlexRay internal message handler but the number of concurrent tasks is unknown, not only for a worst case consideration the number of concurrent tasks must be set to 3. Beside the setup time and the number of words to be transferred a message handler internal delay time of 75 bus clock cycles must be added. The maximum number of payload to be transferred is in worst case 64.

The worst case for an input transfer is different from the worst case for an output transfer. For an input transfer the setup time required is 6 bus clock cycles and therefore

$$cycles_{req\_MHD\ input[64]} = 3 \times (6 + 68 + 75) = 447$$

Equation 5 Message Hander Internal Input Transfer Time (Worst Case)

For an output transfer the setup time required is 2 bus clock cycles and therefore

$$cycles_{req\_MHD\ output[64]} = 3 \times (2 + 68 + 75) = 435$$

Equation 6 Message Handler Internal Output Transfer Time (Worst Case)

(2) System Bus Transfer Time

The number of bus clock cycles required for a transfer is given in bus clock cycles and is identical for all kind of transfers between the local RAM/global RAM and the FlexRay module.

The time required by the transfer handler for a transfer between the local RAM/global RAM and the FlexRay interface buffer is given by

$$cycles_{req\_SB[PL]} = throughput \times \left(systemTime + \left(DataWords_{req[PL]}\right) + setupTime\right)$$

Equation 7 General System Bus Transfer Time Formula

The number of payload words (32 bit) can be calculated by using Equation 4.

The throughput depends on the utilization of the system bus and depends on the application. However, the data transfer handler has the highest priority for accessing the local RAM/global RAM and the data throughput factor can be assumed as XXXX even in worst case scenarios. The system time (read of the data pointer from local RAM/global RAM) can be set to 1 in all cases. The transfer setup time can be set to 3 in all cases.

Therefore the system bus transfer time (given in bus clock cycles) is given by

$$cycles_{req\_SB[64]} = throughput \times (2 + 68) + 3 = XXXX \times (2 + 68) + 3$$

Equation 8 System Bus Transfer Time (Worst Case)

In addition to the system bus transfer time a FlexRay module internal processing time needs to be taken into account. The number of clock cycles required for the internal processing time required is given in FlexRay core module clock cycles (bus clock) and is different between input transfers and output transfers but can be treated as constants.

$$cycles_{req\_THD\ input} = 2 + 4 = 6$$

Equation 9

Whereby 2 cycles are required to initiate the FlexRay module internal transfer from the input buffer to the message RAM. In addition 4 internal processing cycles are required.

$$cycles_{req\_THD\ output} = 3 + 4 = 7$$

Equation 10

Whereby 3 cycles are required to initiate the FlexRay module internal transfer from the input buffer to the message RAM. In addition 4 internal processing cycles are required.

(3) Summary Required Input Transfer Time

The worst case time (as a function of the transferred payload) required to transfer data between the local RAM/global RAM to the FlexRay internal message buffer can be calculated by

$$t\_input_{[PL]} = t\_chi \times \left(cycles_{req\_THD\_input} + cycles_{req\_MHD\_input[PL]}\right) + t\_sys \times \left(cycles_{req\_SB[\mathrm{PL}]}\right)$$

$$t\_input_{[PL]} = t\_chi \times \left(6 + 3 \times \left[81 + DataWords_{[PL]}\right]\right) + t\_sys \times \left(throughput \times \left[2 + DataWords_{[PL]}\right] + 3\right)$$

Equation 11 Input transfer time

With

$$t\_sys = \left(f(sysclk)\right)^{-1}$$

and.

$$t\_chi = \left(f(clkp2\_flx)\right)^{-1}$$

(4) Summary Required Output Transfer Time

The worst case time (as a function of the transferred payload) required to transfer data from the FlexRay internal message buffer to the local RAM/global RAM can be calculated by

$$t\_output_{[PL]} = t\_chi \times \left(cycles_{req\_THD\_output} + cycles_{req\_MHD\_output[PL]}\right) + t\_sys \times \left(cycles_{req\_SB[\mathrm{PL}]}\right)$$

$$t\_output_{[PL]} = t\_chi \times \left(7 + 3 \times \left[77 + DataWords_{[PL]}\right]\right) + t\_sys \times \left(throughput \times \left[2 + DataWords_{[PL]}\right] + 3\right)$$

Equation 12 Output Transfer Time

With

$$t\_sys = \left(f(sysclk)\right)^{-1}$$

and.

$$t\_chi = \left(f(clkp2\_flx)\right)^{-1}$$

25.3.17 Byte Alignment

The alignment of the bytes received by the FlexRay protocol and the alignment of the bytes required by the application may be different. The FlexRay module provides with FLXAnFROC.BEC a byte alignment function to support different byte ordering styles.

Figure 25.32, Byte Alignment on the FlexRay Bus shows the payload byte alignment in a FlexRay frame.

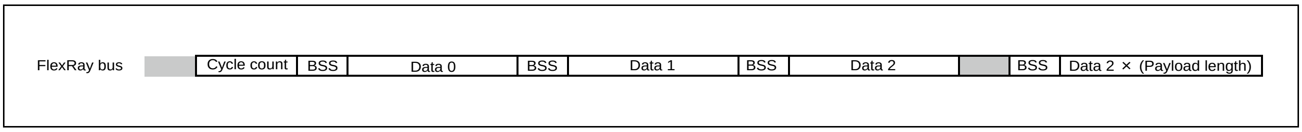


Figure 25.32 Byte Alignment on the FlexRay Bus

25.3.17.1 Little Endian Alignment

When FLXAnFROC.BEC is 0, the byte alignment is set to Little Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as below.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXAnFRNMV1	Data 3								Data 2								Data 1								Data 0							
FLXAnFRNMV2	Data 7								Data 6								Data 5								Data 4							
FLXAnFRNMV3	Data 11								Data 10								Data 9								Data 8							

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as below.

FLXAnFRWRDSx.MD[7:0] = Data_{4x-4}
FLXAnFRWRDSx.MD[15:8] = Data_{4x-3}
FLXAnFRWRDSx.MD[23:16] =Data_{4x-2}
FLXAnFRWRDSx.MD[31:24] = Data_{4x-1}

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[7:0], FLXAnFRWRDSx.MD [15:8], FLXAnFRWRDSx.MD [23:16], FLXAnFRWRDSx.MD [31:24] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure is as below.

FLXAnFRRDDSx.MD[7:0] = Data_{4x-4}
FLXAnFRRDDSx.MD[15:8] = Data_{4x-3}
FLXAnFRRDDSx.MD[23:16] =Data_{4x-2}
FLXAnFRRDDSx.MD[31:24] = Data_{4x-1}

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[7:0], FLXAnFRRDDSx.MD [15:8], FLXAnFRRDDSx.MD [23:16], FLXAnFRRDDSx.MD [31:24] with the most significant bit (MSB) received first.

25.3.17.2 Big Endian Alignment

When FLXAnFROC.BEC is 1, the byte alignment is set to Big Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as below.

Bit	31 to 24	23 to 16	15 to 8	7 to 0
FLXAnFRNMV1	Data 0	Data 1	Data 2	Data 3
FLXAnFRNMV2	Data 4	Data 5	Data 6	Data 7
FLXAnFRNMV3	Data 8	Data 9	Data 10	Data 11

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as below.

FLXAnFRWRDSx.MD[7:0] = Data_{4x-1}
FLXAnFRWRDSx.MD[15:8] = Data_{4x-2}
FLXAnFRWRDSx.MD[23:16] =Data_{4x-3}
FLXAnFRWRDSx.MD[31:24] = Data_{4x-4}

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[31:24], FLXAnFRWRDSx.MD[23:16], FLXAnFRWRDSx.MD[15:8], FLXAnFRWRDSx.MD[7:0] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure is as below.

FLXAnFRRDDSx.MD[7:0] = Data_{4x-1}
FLXAnFRRDDSx.MD[15:8] = Data_{4x-2}
FLXAnFRRDDSx.MD[23:16] =Data_{4x-3}
FLXAnFRRDDSx.MD[31:24] = Data_{4x-4}

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[31:24], FLXAnFRRDDSx.MD[23:16], FLXAnFRRDDSx.MD[15:8], FLXAnFRRDDSx.MD[7:0] with the most significant bit (MSB) received first.

25.4 Detection and Correction of Errors in FlexRay RAM

For details of ECC, see **Section 40A.2.6, ECC for Peripheral RAM**, **Section 40B.2.6, ECC for Peripheral RAM**, **Section 40C.2.5, ECC for Peripheral RAM**.

Section 26 Ethernet AVB (ETNB)

This section contains a generic description of the Ethernet AVB (ETNB).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of ETNB.

CAUTION

Read URAM in this manual as local RAM/global RAM/retention RAM.
Read AXI BUS in this manual as H-Bus.

26.1 Features of RH850/F1KH, RH850/F1KM ETNB

26.1.1 Number of Units

This microcontroller has the following number of ETNB units.

Each ETNB unit has single channel interface.

Table 26.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	2
Name	ETNBn (n = 0)	ETNBn (n = 0)	ETNBn (n = 0,1)

Table 26.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	—	—	1	1	1
Name	—	—	ETNBn (n = 0)	ETNBn (n = 0)	ETNBn (n = 0)

Table 26.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	—	—
Name	—	—	—	—

Table 26.4 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual ETNB units are identified by the index “n”; for example, ETNBnCCC (n = 0, 1) is the AVB-DMAC mode register.

Table 26.5 Index (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual ETNB units are identified by the index “n”; for example, ETNBnCCC (n = 0) is the AVB-DMAC mode register.

26.1.2 Register Base Addresses

ETNB base addresses are listed in the following table.

ETNB register addresses are given as an offset from the base addresses.

Table 26.6 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<ETNB0_base>	FFD6 E000$_H$
<ETNB1_base>	FFD6 E800$_H$

Table 26.7 Register Base Address (RH850/F1KM-S4)

Base Address Name	Base Address
<ETNB0_base>	FFD6 E000$_H$

26.1.3 Clock Supply

The ETNB clock supply is shown in the following table.

Table 26.8 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name
ETNBn	clk_chi*1,*2	CPUCLK_L
	Register access clock*1,*2	CPUCLK_L

Note 1. ETNB clocks can be stoped by OPBT0.ETNBnEN, refer to **Section 44.9.2, OPBT0 — Option Byte 0**.

Note 2. ETNB clocks is more than 15 MHz.

Table 26.9 Clock Supply (RH850/F1KM-S4)

Unit Name	Unit Clock Name	Supply Clock Name
ETNBn	clk_chi*1,*2	CPUCLK_L
	Register access clock*1,*2	CPUCLK_L

Note 1. ETNB clocks can be stoped by OPBT0.ETNBnEN, refer to **Section 44.9.2, OPBT0 — Option Byte 0**.

Note 2. ETNB clocks is more than 15 MHz.

26.1.4 Interrupt Requests

ETNB interrupt requests are listed in the following table.

Table 26.10 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
ETNB0			
INTETNB0DATA	Data related interrupt	315	—
INTETNB0ERR	Error related interrupt	316	—
INTETNB0MNG	Management related interrupt	317	—
INTETNB0MAC	MAC interrupt	318	—
ETNB1			
INTETNB1DATA	Data related interrupt	377	—
INTETNB1ERR	Error related interrupt	378	—
INTETNB1MNG	Management related interrupt	379	—
INTETNB1MAC	MAC interrupt	380	—

Table 26.11 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
ETNB0			
INTETNB0DATA	Data related interrupt	315	—
INTETNB0ERR	Error related interrupt	316	—
INTETNB0MNG	Management related interrupt	317	—
INTETNB0MAC	MAC interrupt	318	—

26.1.5 Reset Sources

ETNB reset sources are listed in the following table. ETNB is initialized by these reset sources.

Table 26.12 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
ETNBn	All reset sources (ISORES)

Table 26.13 Reset Sources (RH850/F1KM-S4)

Unit Name	Reset Source
ETNBn	All reset sources (ISORES)

26.1.6 External Input/Output Signals

External input/output signals of ETNB are listed below.

Table 26.14 ETNBn Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
ETNB0		
AVB_TX_CLK	MII transmit clock signal	ETNB0TXCLK
AVB_RX_CLK	MII receive clock signal	ETNB0RXCLK
AVB_TX_EN	MII transmit data enable signal	ETNB0TXEN
AVB_TXD[3:0]	MII transmit data signal	ETNB0TXD[3:0]
AVB_RX_DV	MII receive data valid signal	ETNB0RXDV
AVB_RXD[3:0]	MII receive data signal	ETNB0RXD[3:0]
AVB_RX_ER	MII receive error signal	ETNB0RXERR
AVB_MDC	PHY management clock signal	ETNB0MDC
AVB_MDIO	PHY management transfer data signal	ETNB0MDIO
AVB_LINK	PHY link status signal	ETNB0LINK
ETNB1		
AVB_TX_CLK	MII transmit clock signal	ETNB1TXCLK
AVB_RX_CLK	MII receive clock signal	ETNB1RXCLK
AVB_TX_EN	MII transmit data enable signal	ETNB1TXEN
AVB_TXD[3:0]	MII transmit data signal	ETNB1TXD[3:0]
AVB_RX_DV	MII receive data valid signal	ETNB1RXDV
AVB_RXD[3:0]	MII receive data signal	ETNB1RXD[3:0]
AVB_RX_ER	MII receive error signal	ETNB1RXERR
AVB_MDC	PHY management clock signal	ETNB1MDC
AVB_MDIO	PHY management transfer data signal	ETNB1MDIO
AVB_LINK	PHY link status signal	ETNB1LINK

Note: In RH850/F1KH, there is no TXERR pin. It is recommended to connect to EVSS or BVSS via a resistor for TXERR of PHY.

Table 26.15 ETNBn Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
ETNB0		
AVB_TX_CLK	MII transmit clock signal	ETNB0TXCLK
AVB_RX_CLK	MII receive clock signal	ETNB0RXCLK
AVB_TX_EN	MII transmit data enable signal	ETNB0TXEN
AVB_TXD[3:0]	MII transmit data signal	ETNB0TXD[3:0]
AVB_RX_DV	MII receive data valid signal	ETNB0RXDV
AVB_RXD[3:0]	MII receive data signal	ETNB0RXD[3:0]
AVB_RX_ER	MII receive error signal	ETNB0RXERR
AVB_MDC	PHY management clock signal	ETNB0MDC
AVB_MDIO	PHY management transfer data signal	ETNB0MDIO
AVB_LINK	PHY link status signal	ETNB0LINK

Note: In RH850/F1KM, there is no TXERR pin. It is recommended to connect to EVSS or BVSS via a resistor for TXERR of PHY.

26.2 Overview

26.2.1 Functional Overview

Table 26.16, Specifications of ETNB lists the specifications of the ETNB.

Table 26.16 Specifications of ETNB

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
Communication interface	MII
Transfer speed	● 100 Mbps ● 10 Mbps*1
Transfer mode	● Full-duplex mode
AVB function	● Conforming with the following standards stipulated for IEEE 802.1BA. – IEEE 802.1AS (time synchronization protocol) – IEEE 802.1Qav (real-time transfer) * IEEE 802.1Qat should be supported by software. ● Descriptor management system ● IEEE 1722 (AVTP presentation timestamp)

Note 1. Not available for the AVB function.

26.2.2 Block Diagram

Figure 26.1, Block Diagram of ETNB is a block diagram of the ETNB.

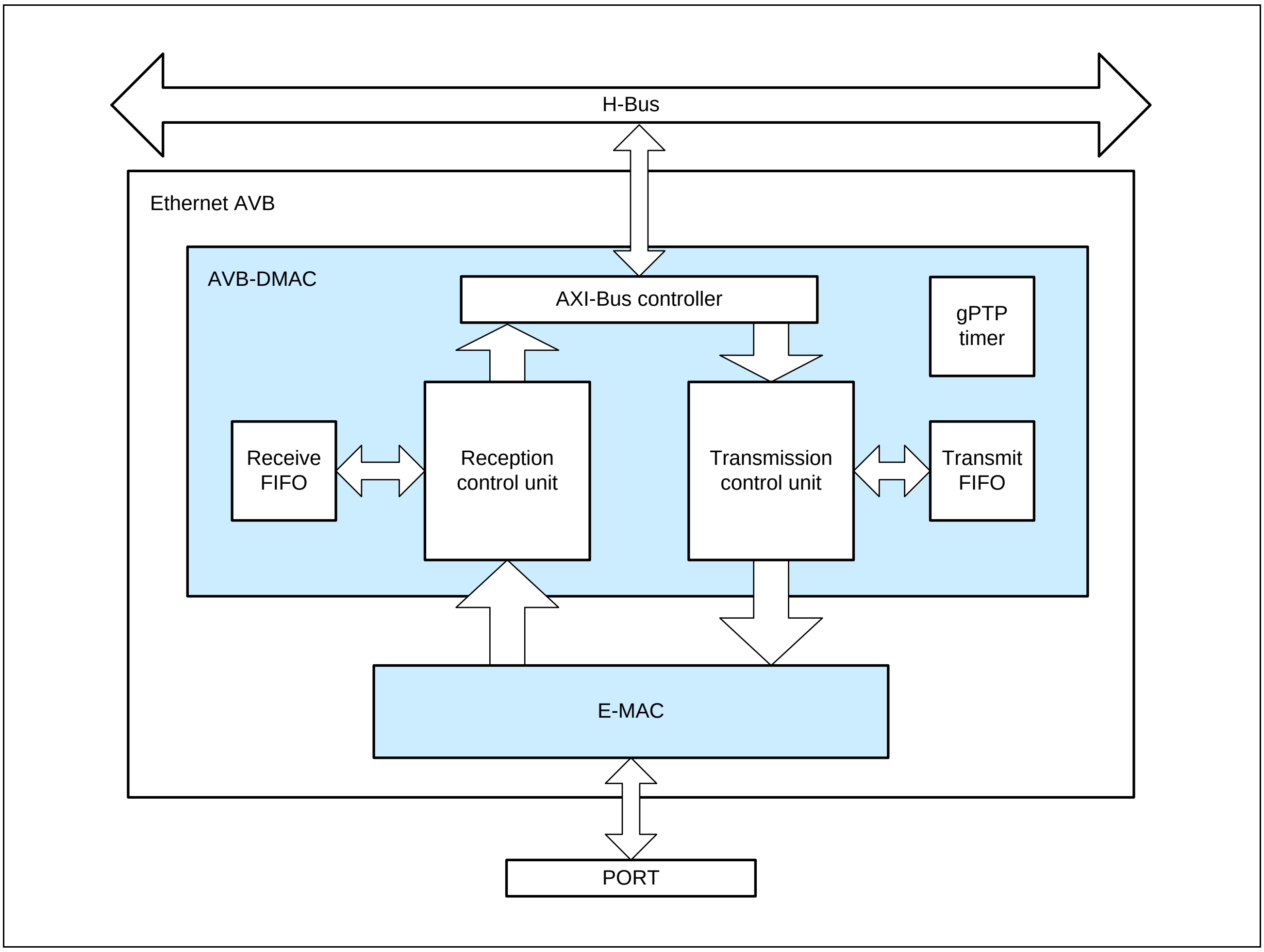


Figure 26.1 Block Diagram of ETNB

26.3 Registers

26.3.1 List of Registers

The following table lists the ETNB registers.

For details about <ETNBn_base>, see **Section 26.1.2, Register Base Address**.

Table 26.17 List of Registers

Module Name	Register Name	Abbreviation	Address
ETNBn	AVB-DMAC mode register	ETNBnCCC	<ETNBn_base> + 0000_H
	Descriptor base address table register	ETNBnDBAT	<ETNBn_base> + 0004_H
	Descriptor base address load request register	ETNBnDLR	<ETNBn_base> + 0008_H
	AVB-DMAC status register	ETNBnCSR	<ETNBn_base> + $000C_H$
	Current descriptor address register q (q = 0 to 21)	ETNBnCDARq	<ETNBn_base> + 0010_H + q × 4_H
	Error status register	ETNBnESR	<ETNBn_base> + 0088_H
	Receive configuration register	ETNBnRCR	<ETNBn_base> + 0090_H
	Receive queue configuration register i (i = 0 to 4)	ETNBnRQCi	<ETNBn_base> + 0094_H + i × 4_H
	Receive padding configuration register	ETNBnRPC	<ETNBn_base> + $00B0_H$
	Unread frame counter stop level configuration register	ETNBnUFCS	<ETNBn_base> + $00C0_H$
	Unread frame counter register i (i = 0 to 4)	ETNBnUFCVi	<ETNBn_base> + $00C4_H$ + i × 4_H
	Unread frame counter decrement register i (i = 0 to 4)	ETNBnUFCDi	<ETNBn_base> + $00E0_H$ + i × 4_H
	Separation filter offset register	ETNBnSFO	<ETNBn_base> + $00FC_H$
	Separation filter pattern register i (i = 0 to 31)	ETNBnSFPi	<ETNBn_base> + 0100_H + i × 4_H
	Separation filter mask register i (i = 0, 1)	ETNBnSFMi	<ETNBn_base> + $01C0_H$ + i × 4_H
	Transmit configuration register	ETNBnTGC	<ETNBn_base> + 0300_H
	Transmit configuration control register	ETNBnTCCR	<ETNBn_base> + 0304_H
	Transmit status register	ETNBnTSR	<ETNBn_base> + 0308_H
	Timestamp FIFO access register 0	ETNBnTFA0	<ETNBn_base> + 0310_H
	Timestamp FIFO access register 1	ETNBnTFA1	<ETNBn_base> + 0314_H
	Timestamp FIFO access register 2	ETNBnTFA2	<ETNBn_base> + 0318_H
	CBS increment value register c (c = 0, 1)	ETNBnCIVRc	<ETNBn_base> + 0320_H + c × 4_H
	CBS decrement value register c (c = 0, 1)	ETNBnCDVRc	<ETNBn_base> + 0328_H + c × 4_H
	CBS upper limit register c (c = 0, 1)	ETNBnCULc	<ETNBn_base> + 0330_H + c × 4_H
	CBS lower limit register c (c = 0, 1)	ETNBnCLLc	<ETNBn_base> + 0338_H + c × 4_H
	Descriptor interrupt control register	ETNBnDIC	<ETNBn_base> + 0350_H
	Descriptor interrupt status register	ETNBnDIS	<ETNBn_base> + 0354_H
	Error interrupt control register	ETNBnEIC	<ETNBn_base> + 0358_H
	Error interrupt status register	ETNBnEIS	<ETNBn_base> + $035C_H$
	Receive interrupt control register 0	ETNBnRIC0	<ETNBn_base> + 0360_H
	Receive interrupt status register 0	ETNBnRIS0	<ETNBn_base> + 0364_H
	Receive interrupt control register 1	ETNBnRIC1	<ETNBn_base> + 0368_H
	Receive interrupt status register 1	ETNBnRIS1	<ETNBn_base> + $036C_H$
	Receive interrupt control register 2	ETNBnRIC2	<ETNBn_base> + 0370_H
	Receive interrupt status register 2	ETNBnRIS2	<ETNBn_base> + 0374_H
	Transmit interrupt control register	ETNBnTIC	<ETNBn_base> + 0378_H
	Transmit interrupt status register	ETNBnTIS	<ETNBn_base> + $037C_H$
	Interrupt summary status register	ETNBnISS	<ETNBn_base> + 0380_H
	gPTP configuration control register	ETNBnGCCR	<ETNBn_base> + 0390_H

Table 26.17 List of Registers

Module Name	Register Name	Abbreviation	Address
ETNBn	gPTP maximum transit time register	ETNBnGMTT	<ETNBn_base> + 0394_H
	gPTP presentation time comparison register	ETNBnGPTC	<ETNBn_base> + 0398_H
	gPTP timer increment register	ETNBnGTI	<ETNBn_base> + $039C_H$
	gPTP timer offset register i (i = 0 to 2)	ETNBnGTOi	<ETNBn_base> + $03A0_H$ + i × 4_H
	gPTP interrupt control register	ETNBnGIC	<ETNBn_base> + $03AC_H$
	gPTP interrupt status register	ETNBnGIS	<ETNBn_base> + $03B0_H$
	gPTP timer capture register i (i = 0 to 2)	ETNBnGCTi	<ETNBn_base> + $03B8_H$ + i × 4_H
	E-MAC mode register	ETNBnECMR	<ETNBn_base> + 0500_H
	Receive frame length register	ETNBnRFLR	<ETNBn_base> + 0508_H
	E-MAC status register	ETNBnECSR	<ETNBn_base> + 0510_H
	E-MAC interrupt permission register	ETNBnECSIPR	<ETNBn_base> + 0518_H
	PHY interface register	ETNBnPIR	<ETNBn_base> + 0520_H
	PHY LINK status register	ETNBnPLSR	<ETNBn_base> + 0528_H
	Auto PAUSE frame time parameter register	ETNBnAPFTP	<ETNBn_base> + 0554_H
	Manual PAUSE frame register	ETNBnMPR	<ETNBn_base> + 0558_H
	PAUSE frame transmit counter	ETNBnPFTCR	<ETNBn_base> + $055C_H$
	PAUSE frame receive counter	ETNBnPFRCR	<ETNBn_base> + 0560_H
	MAC address high register	ETNBnMAHR	<ETNBn_base> + $05C0_H$
	MAC address low register	ETNBnMALR	<ETNBn_base> + $05C8_H$
	Transmit retry over counter register	ETNBnTROCR	<ETNBn_base> + 0700_H
	Late collision detect counter register	ETNBnCDCR	<ETNBn_base> + 0708_H
	Lost carrier counter register	ETNBnLCCR	<ETNBn_base> + 0710_H
	CRC error frame receive counter register	ETNBnCEFCR	<ETNBn_base> + 0740_H
	Frame receive error counter register	ETNBnFRECR	<ETNBn_base> + 0748_H
	Too-short frame receive counter register	ETNBnTSFRCR	<ETNBn_base> + 0750_H
	Too-long frame receive counter register	ETNBnTLFRCR	<ETNBn_base> + 0758_H
	Residual-bit frame receive counter register	ETNBnRFCR	<ETNBn_base> + 0760_H
	Multicast address frame receive counter register	ETNBnMAFCR	<ETNBn_base> + 0778_H
	Communication interface control register	ETNBnIFCTL	<ETNBn_base> + 1000_H

26.3.2 ETNBnCCC — AVB-DMAC Mode Register

The ETNBnCCC register specifies the operating mode of the AVB-DMAC.

Access: ETNBnCCC can be read or written in 32-bit units.
ETNBnCCCL and ETNBnCCCH can be read or written in 16-bit units.
ETNBnCCCLL, ETNBnCCCLH, ETNBnCCCHL, ETNBnCCCHH can be read or written in 8-bit units.

Address: ETNBnCCC: <ETNBn_base> + 0000$_H$
ETNBnCCCL: <ETNBn_base> + 0000$_H$
ETNBnCCCH: <ETNBn_base> + 0002$_H$
ETNBnCCCLL: <ETNBn_base> + 0000$_H$
ETNBnCCCLH: <ETNBn_base> + 0001$_H$
ETNBnCCCHL: <ETNBn_base> + 0002$_H$
ETNBnCCCHH: <ETNBn_base> + 0003$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	FCE	LBME	—	—	—	BOC	—	—	CSEL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTSR	—	—	—	—	—	—	OPC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 26.18 ETNBnCCC Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	FCE	Flow Control Enable 0: Flow control disabled 1: Flow control enabled
24	LBME	Loopback Mode Enable 0: Normal operation 1: Loopback mode is enabled.
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	BOC	Byte Order Configuration 0: Big endian The first byte is the 8 lower-order bits (URAM[7:0]) 1: Little endian The first byte is the 8 higher-order bits (URAM[31:24])
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	CSEL[1:0]	gPTP Clock Select 00$_B$: gPTP is not in use. 01$_B$: Peripheral bus clock (clk_chi) 10$_B$: Ethernet transmission clock (ETNBnTXCLK) 11$_B$: Setting prohibited
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 26.18 ETNBnCCC Register Contents

Bit Position	Bit Name	Function
8	DTSR	Data Transmission Suspend Request 0: Normal operation 1: Requests suspension
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	OPC[1:0]	Operating Mode Configuration 00_B: Reset mode 01_B: Configuration mode 10_B: Operation mode 11_B: Standby mode

FCE (Flow Control Enable) Bit

This bit enables the flow control support of MAC.

When flow control is enabled, the MAC gets informed about the Rx-FIFO level (Rx-FIFO fill level reached ETNBnRCR.RFCL).

LBME (Loopback Mode Enable) Bit

This bit enables loopback mode.

In loopback mode, the transmission lines are internally connected to the reception lines. When loopback mode is to be used, the Ethernet transmission clock must be supplied to the MII interface. A received clock signal is not required. Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

Data for transmission are still output normally. To eliminate effects on external modules, pin control should be applied to block the output of data. For details about the pin control, see **Section 2A, Pin Function of RH850/F1KH-D8**, **Section 2B, Pin Function of RH850/F1KM-S4**.

BOC (Byte Order Configuration) Bit

This bit specifies the assignment of the first byte of received Ethernet frames when it is allocated to the URAM.

This setting of this register does not affect the descriptor format and filter parameters of the URAM. Writing to this bit is only possible when the current operating mode is configuration mode.

Figure 26.2, Data of Ethernet Frame Received to **Figure 26.4, When ETNBnCCC.BOC = 1** show how data from frames received via the Ethernet connection are stored in the URAM.

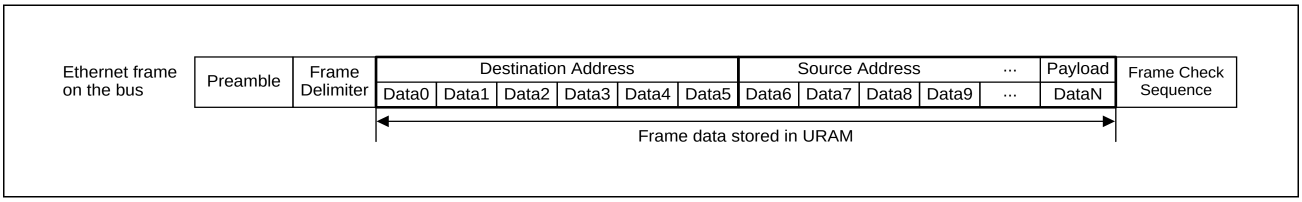


Figure 26.2 Data of Ethernet Frame Received

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Data3								Data2								Data1								Data0							
DPTR+4	Data7								Data6								Data5								Data4							
DPTR+8	Data11								Data10								Data9								Data8							

Figure 26.3 When ETNBnCCC.BOC = 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Data0								Data1								Data2								Data3							
DPTR+4	Data4								Data5								Data6								Data7							
DPTR+8	Data8								Data9								Data10								Data11							

Figure 26.4 When ETNBnCCC.BOC = 1

CSEL[1:0] (gPTP Clock Select) Bits

These bits select the clock source for the gPTP timer.

Writing to these bits is only possible when the current operating mode is configuration mode.

DTSR (Data Transmission Suspend Request) Bit

This bit can suspend access to the URAM.

The access is suspended on completion of the transfer of the frame currently being transferred.

This function disables access to the URAM without affecting normal operation of the AVB-DMAC. Use this bit when exclusive control over the contents of the URAM is necessary, for example, in checking its integrity.

CAUTION

The transmission and reception queues are not processed while access is suspended. Change neither the AVB-DMAC settings nor the mode until the suspend request is given by ETNBnCCC.DTSR and later the value of ETNBnCSR.DTS is updated.

OPC[1:0] (Operating Mode Configuration) Bits

These bits specify the operating mode.

For the operating modes, see **Section 26.4.1.1, Operating Modes**.

Writing to this bit is possible in any of the operating modes, but should not be done after the application system has issued a Power Off request.

26.3.3 ETNBnDBAT — Descriptor Base Address Table Register

The ETNBnDBAT register is used to set the base address of the descriptor table.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0004_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.19 ETNBnDBAT Register Contents

Bit Position	Bit Name	Function
31 to 0	TA[31:0]	Descriptor Base Table Address Base address of the descriptor table in the URAM

CAUTION

The setting of this bit must be a multiple of four (Bit 0 and bit 1 must be set to “0”).

TA[31:0] (Descriptor Base Table Address) Bits

These bits specify the base address of the descriptor table in the URAM.

For the structure of the descriptor base address table, see **Section 26.4.3, Descriptors**.

Writing to this bit is only possible when the current operating mode is configuration mode.

26.3.4 ETNBnDLR — Descriptor Base Address Load Request Register

The ETNBnDLR register is used to issue a request to load the values from the descriptor base address table register (ETNBnDBAT) for each queue to the current descriptor address register q (ETNBnCDARq).

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0008_H
Value after reset:	$003F\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	LBA21	LBA20	LBA19	LBA18	LBA17	LBA16
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBA15	LBA14	LBA13	LBA12	LBA11	LBA10	LBA9	LBA8	LBA7	LBA6	LBA5	LBA4	LBA3	LBA2	LBA1	LBA0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.20 ETNBnDLR Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	LBA21	Base Address Load Request (Rx17: Stream 15) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
20	LBA20	Base Address Load Request (Rx16: Stream 14) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
19	LBA19	Base Address Load Request (Rx15: Stream 13) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
18	LBA18	Base Address Load Request (Rx14: Stream 12) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
17	LBA17	Base Address Load Request (Rx13: Stream 11) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
16	LBA16	Base Address Load Request (Rx12: Stream 10) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
15	LBA15	Base Address Load Request (Rx11: Stream 9) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 26.20 ETNBnDLR Register Contents

Bit Position	Bit Name	Function
14	LBA14	Base Address Load Request (Rx10: Stream 8) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
13	LBA13	Base Address Load Request (Rx9: Stream 7) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
12	LBA12	Base Address Load Request (Rx8: Stream 6) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
11	LBA11	Base Address Load Request (Rx7: Stream 5) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
10	LBA10	Base Address Load Request (Rx6: Stream 4) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
9	LBA9	Base Address Load Request (Rx5: Stream 3) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
8	LBA8	Base Address Load Request (Rx4: Stream 2) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
7	LBA7	Base Address Load Request (Rx3: Stream 1) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
6	LBA6	Base Address Load Request (Rx2: Stream 0) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
5	LBA5	Base Address Load Request (Rx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
4	LBA4	Base Address Load Request (Rx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
3	LBA3	Base Address Load Request (Tx3: Stream Class A) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
2	LBA2	Base Address Load Request (Tx2: Stream Class B) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 26.20 ETNBnDLR Register Contents

Bit Position	Bit Name	Function
1	LBA1	Base Address Load Request (Tx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
0	LBA0	Base Address Load Request (Tx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

LBAq (q = 0 to 21) (Base Address Load Request) Bits

Each bit is used to issue requests to load base addresses and to indicate that a base address is currently being loaded. Setting this bit to 1 issues a request for loading the descriptor base address for the queue q.
If transfer is currently in progress, loading is executed on completion of transfer for the current frame. Completion of loading leads to automatic setting of the corresponding bit to 0.

For transmission queues, base address load requests are executed even while fetching is in progress (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is 1). Therefore, be sure to check that fetching is not in progress before issuing a request.

Writing to a bit of this register is only possible when the current operating mode is operation mode. Only 1 can be written to this bit.

26.3.5 ETNBnCSR — AVB-DMAC Status Register

The ETNBnCSR register is used to indicate the operating mode in which the AVB-DMAC is running and communications states.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<ETNBn_base> + 000C$_{\text{H}}$
Value after reset:	0000 0001$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RPO	TPO3	TPO2	TPO1	TPO0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTS	—	—	—	—	OPS[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.21 ETNBnCSR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20	RPO	Receive Process Status 0: Normal operation 1: Reception is in progress.
19	TPO3	Transmit Process Status 3 (Stream Class A) 0: Normal operation 1: Transmission is in progress.
18	TPO2	Transmit Process Status 2 (Stream Class B) 0: Normal operation 1: Transmission is in progress.
17	TPO1	Transmit Process Status 1 (Network Control) 0: Normal operation 1: Transmission is in progress.
16	TPO0	Transmit Process Status 0 (Best Effort) 0: Normal operation 1: Transmission is in progress.
15 to 9	Reserved	When read, the value after reset is returned.
8	DTS	Data Transmission Suspended Status 0: Normal operation 1: Transmission is suspended.
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	OPS[3:0]	Operating Mode Status 0001$_{\text{B}}$: Reset 0010$_{\text{B}}$: Configuration mode 0100$_{\text{B}}$: Operation mode 1000$_{\text{B}}$: Standby mode Other settings are prohibited.

RPO (Receive Process Status) Bit

This bit indicates whether a reception queue contains an unread received frame.

This bit being set to 1 indicates that a received frame is yet to be stored in the URAM.

[Clearing conditions]

- The current operating mode changes from operation mode.
- Received frames in the reception FIFO all being stored in the URAM.

[Setting condition]

- A received frame being stored in the reception FIFO (but not yet in the URAM)

TPO3 (Transmit Process Status 3) Bit

This bit indicates whether a class A stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

[Clearing conditions]

- The current operating mode changes from operation mode.
- Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ3) is set to 0)

[Setting condition]

- Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ3))

TPO2 (Transmit Process Status 2) Bit

This bit indicates whether a class B stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

[Clearing conditions]

- The current operating mode changes from operation mode.
- Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ2) is set to 0)

[Setting condition]

- Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ2))

TPO1 (Transmit Process Status 1) Bit

This bit indicates whether a network control is being transmitted.

When this bit is set to 1, the AVB-DMAC is fetching data in the URAM or E-MAC is transmitting data.

[Clearing conditions]

- The current operating mode changes from operation mode.
- Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ1) is set to 0)

[Setting condition]

- Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ1))

TPO0 (Transmit Process Status 0) Bit

This bit indicates whether a best effort is being transmitted.

When this bit is set to 1, the AVB-DMAC is fetching data in the URAM or E-MAC is transmitting data.

[Clearing conditions]

- The current operating mode changes from operation mode.
- Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ0) is set to 0)

[Setting condition]

- Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ0))

DTS (Data Transmission Suspend Status) Bit

This bit indicates whether access to the URAM is enabled.

[Clearing conditions]

- The current operating mode is not operating mode.
- The data transmission suspend request bit in the AVB-DMAC mode register (ETNBnCCC.DTSR) being 0.

[Setting condition]

- Access to the URAM not proceeding while the data transmission suspend request bit (ETNBnCCC.DTSR) in the AVB-DMAC mode register is 1 (if the URAM is being accessed, this bit is set to 1 on completion of access).

OPS[3:0] (Operating Mode Status) Bits

These bits indicate the current operating mode.

For the operating modes, see **Section 26.4.1.1, Operating Modes**.

26.3.6 ETNBnCDARq — Current Descriptor Address Register q (q = 0 to 21)

The ETNBnCDARq register indicates the current descriptor address.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0010_H + q × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.22 ETNBnCDARq Register Contents

Bit Position	Bit Name	Function
31 to 0	CDA[31:0]	Current Descriptor Address The address of the current descriptors for the transmission queues

CDA[31:0] (Current Descriptor Address) Bits

ETNBnCDAR0 to ETNBnCDAR3 indicate the addresses of the current descriptors for the corresponding transmission queues while ETNBnCDAR4 to ETNBnCDAR21 indicate the addresses of the current descriptors for the corresponding reception queues.

If the operating mode is changed to operation mode, contents of this bits are changed to (ETNBnDBAT + q × 8).

Also, when the descriptor base address load request register (ETNBnDLR) issues a load request, the contents of the corresponding register are changed to the descriptor base address table register (ETNBnDBAT).

[Conditions for Updating]

- These bits are set to 0 when the operating mode is not operation mode.
- This register is updated when a descriptor to a queue is processed.

26.3.7 ETNBnESR — Error Status Register

The ETNBnESR register indicates the state of error by the AVB-DMAC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0088_{H}

Value after reset: $0000\ 0000_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	EIL	ET[3:0]				—	—	—	EQN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.23 ETNBnESR Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12	EIL	Error Information Lost 0: No loss of error information 1: Lost of error information detected
11 to 8	ET[3:0]	Error Type 0000_{B}: Read descriptor from URAM 0001_{B}: Write descriptor to URAM 0010_{B}: Interpret read descriptor 0011_{B}: Tx-Buffer is corrupted 0100_{B}: Read data from URAM 0101_{B}: Write data or timestamp to URAM 0110_{B}: Reading from received FIFO 0111_{B}: Received FIFO is corrupted 1000_{B}: Frame size error during reception detected 1001_{B}: Frame size error during transmission detected 1010_{B}: Tx-Buffer overflow
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	EQN[4:0]	Error Queue Number

EIL Error Information Lost Bit

This bit indicates that error information detected by Ethernet AVB is lost because the previous reported error has not been processed by CPU.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 0 when CPU writes 0 to the queue error flag (ETNBnEIS.QEF) of the error interrupt status register.
- This bit is set to 1 when the set condition of the queue error flag (ETNBnEIS.QEF) is fulfilled while the queue error flag (ETNBnEIS.QEF) is 1.

ET[3:0] Error Type Bits

These bits indicate details about the transfer stage which was handled when Ethernet AVB has detected an error.

When a fault relates to the read descriptor (ETNBnESR.ET = 0000_B or 0010_B), the CPU needs to correct the faulty descriptor so that the related queues can continue processing. If a queue halts at a faulty descriptor, ETNBnCDARq.CDA (q = ETNBnESR.EQN) directly checks the faulty descriptor.

When the fault is related to descriptor writing (ETNBnESR.ET = 0001_B), CPU needs recognize the not-updated or incorrectly updated descriptor in queue ETNBnESR.EQN. The write problem is not influencing how Ethernet AVB processes the descriptor chain.

When the fault is related to the Tx-Buffer (ETNBnESR.ET = 0011_B) CPU needs to clean-up the Tx-Buffer to correct the buffer control configuration.

All other errors are transient in nature and may be corrected by continuation of HW or SW operation; so there is no strong demand on CPU interaction. For the details of the error processing, **Section 26.4.2.3, Checking Integrity**.

The CPU can only evaluate these bits when the queue error flag (ETNBnEIS.QEF) is 1.

[Changing condition]

These bits are updated when the queue error flag (ETNBnEIS.QEF) setting conditions are satisfied and ETNBnEIS.QEF is 0.

EQN[4:0] Error Queue Number Bits

These bits indicate the queue number which was handled when Ethernet AVB has detected an error. A fault reported for ETNBnESR.EQN = 0 to 3 is related to transmit queue t = 0 to 3.

From ETNBnESR.EQN = 4 the fault is related to receive queue r = ETNBnESR.EQN – 4. The CPU can only evaluate these bits when the queue error flag (ETNBnEIS.QEF) is 1.

The CPU can not evaluate these bits when the error type bit (ETNBnESR.ET) of the error status register is 0011_B or 0111_B.

[Changing condition]

These bits are updated when the set condition of the queue error flag (ETNBnEIS.QEF) is fulfilled and the queue error flag (ETNBnEIS.QEF) is 0

26.3.8 ETNBnRCR — Receive Configuration Register

The ETNBnRCR register is used to configure receive-relating settings of the AVB-DMAC.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0090_H

Value after reset: $1800\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RFCL[12:0]												
Value after reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ETS2	ETS0	ESF[1:0]		ENCF	EFFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.24 ETNBnRCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	RFCL[12:0]	Receive FIFO Warning Level Recommended value: 1800_H
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	ETS2	Timestamp Enable (Stream) 0: Timestamp is disabled. 1: Timestamp is enabled. Recommended value: 0
4	ETS0	Timestamp Enable (Best Effort) 0: Timestamp is disabled. 1: Timestamp is enabled. Recommended value: 0
3, 2	ESF[1:0]	Stream Filtering Select Settings for reception queues 2 to 17 00_B: Filtering is disabled. Frames are processed in queue 0 (best effort). 01_B: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching stream frames are processed in queue 0 (best effort). 10_B: The separation filter is enabled for AVB stream frames. Non-matching stream frames are discarded. 11_B: The separation filter is enabled for AVB stream frames. Non-matching stream frames are handled with queue 0 (best effort). Recommended value: 10_B or 11_B
1	ENCF	Network Control Filtering Enable Setting for reception queue 1 (network control) 0: Network control is disabled. 1: Network control is enabled.
0	EFFS	Error Frame Enable 0: Error frames are disabled. 1: Error frames are enabled. Recommended value: 0

RFCL[12:0] (Receive FIFO Warning Level) Bits

These bits set the caution level for the reception FIFO and are used to maintain the priority order of the storage of received data and the fetching of data for transmission.

If the reception FIFO contains less data than this level, processing of both transmission and reception queues becomes pending.

If the reception FIFO contains more data than this level, only data in the reception queue are transferred, and processing of the transmission queue becomes pending.

Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

In the case of this LSI chip, set these bits to 1800_H.

ETS2 (Timestamp Enable (Stream)) Bit

This bit enables timestamp information included in reception queues 2 to 17.

Writing to this bit is only possible when the current operating mode is configuration mode. Set this bit to 1 when use the extended descriptor.

ETS0 (Timestamp Enable (Best Effort)) Bit

This bit enables timestamp information included in reception queue 0.

Writing to this bit is only possible when the current operating mode is configuration mode. Set this bit to 1 when use the extended descriptor.

ESF[1:0] (Stream Filtering Select) Bits

These bits select separation filtering to reception queues 2 to 17.

The queue-dependent separation filter can be used in combination with the identification of AVB stream frames.

When the value is 00_B, filtering is disabled and frames from streams are processed in reception queue 0 (best effort).

When the value is 01_B, the separation filter is enabled for both AVB stream frames and non-AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

When the value is 10_B, the separation filter is enabled for AVB stream frames; frames from non- matching streams are discarded.

When the value is 11_B, the separation filter is enabled for AVB stream frames; frames from non- matching streams are processed in reception queue 0 (best effort).

For separation filtering, see **Section 26.4.4.1(1), Separation Filtering**.

Writing to this bit is only possible when the current operating mode is configuration mode.

ENCF (Enable Network Control Filtering) Bit

Enables the AVB network control frame for reception queue 1.

When reception queue 1 is disabled, a received frame is stored in reception queue 0 (best effort). Writing to this bit is only possible when the current operating mode is configuration mode.

EFFS (Enable Error Frame) Bit

Enables or disables the reception of frames that have been classified as error frames by the E-MAC. Received error frames are stored in reception queue 0 (best effort).

An indicator of error detection by the E-MAC during reception is stored in the descriptor (DESCR.MSC).

Writing to this bit is only possible when the current operating mode is configuration mode.

26.3.9 ETNBnRQCi — Receive Queue Configuration Register i (i = 0 to 4)

The ETNBnRQCi register configures the settings of receive queue (r = 0 + i × 4 to 3 + i × 4).*1

Access:	ETNBnRQCi can be read or written in 32-bit units. ETNBnRQCiL and ETNBnRQCiH can be read or written in 16-bit units.*2 ETNBnRQCiLL, ETNBnRQCiLH, ETNBnRQCiHL, ETNBnRQCiHH can be read or written in 8-bit units.*2
Address:	ETNBnRQCi: <ETNBn_base> + 0094_H + i × 4_H ETNBnRQCiL: <ETNBn_base> + 0094_H + i × 4_H ETNBnRQCiH: <ETNBn_base> + 0094_H + i × 4_H + 2_H ETNBnRQCiLL: <ETNBn_base> + 0094_H + i × 4_H ETNBnRQCiLH: <ETNBn_base> + 0094_H + i × 4_H + 1_H ETNBnRQCiHL: <ETNBn_base> + 0094_H + i × 4_H + 2_H ETNBnRQCiHH: <ETNBn_base> + 0094H + i × 4_H + 3_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	UFCCr[1:0] (r = 3 + i × 4)*3		—	—	RSMr[1:0] (r = 3 + i × 4)*3		—	—	UFCCr[1:0] (r = 2 + i × 4)*3		—	—	RSMr[1:0] (r = 2 + i × 4)*3	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	UFCCr[1:0] (r = 1 + i × 4)*3		—	—	RSMr[1:0] (r = 1 + i × 4)*3		—	—	UFCCr[1:0] (r = 0 + i × 4)*3		—	—	RSMr[1:0] (r = 0 + i × 4)*3	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Note 1. The ETNBnRQC4 register corresponds to reception queues from 16 and 17.

Note 2. The ETNBnRQC4 register corresponds to ETNBnRQC4, ETNBnRQC4L, ETNBnRQC4LL, and ETNBnRQC4LH only.

Note 3. The ETNBnRQC4 register corresponds to RSM16[1:0], UFCC16[1:0], RSM17[1:0], and UFCC17[1:0] only and bits from 16 to 31 are reserved bits.

Table 26.25 ETNBnRQCi Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 3 + i × 4) These bits set the unread frame counter used in reception queue 3 + i × 4.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 3 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 2 + i × 4) These bits set the unread frame counter used in reception queue 2 + i × 4.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 2 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 1 + i × 4) These bits set the unread frame counter used in reception queue 1 + i × 4.

Table 26.25 ETNBnRQCi Register Contents

Bit Position	Bit Name	Function
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 1 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 0 + i × 4) These bits set the unread frame counter used in reception queue 0 + i × 4.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 0 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited

UFCCr[1:0] (r = 0 to 17) Unread Frame Counter Configuration Bits

These bits set the unread frame counter for reception queue r.

With the AVB-DMAC, four patterns of settings are available for the unread frame counter. Use the unread frame counter stop level configuration register (ETNBnUFCS) to set the warning level and stop level of the unread frame counter.

Set the pattern number (0 to 3) set in the unread frame counter stop level configuration register (ETNBnUFCS) in these bits.

When the value is 00_B, the stop function is disabled.

Writing to the bits is only possible when the current operating mode is configuration mode.

RSMr[1:0] (r = 0 to 17) Receive Synchronous Mode Bits

These bits set receive synchronous mode. Set 00_B in these bits.

For receive synchronous mode, see **Section 26.4.4.3(3), Mode with Write-Back**.

Writing to the bits is only possible when the current operating mode is configuration mode.

26.3.10 ETNBnRPC — Receive Padding Configuration Register

The ETNBnRPC register is used to set padding for received frames.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $00B0_H$

Value after reset: $0000\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DCNT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 26.26 ETNBnRPC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	DCNT[7:0]	Stored Data Counter These bits specify the amount of data to be stored with the descriptor. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	PCNT[2:0]	Stored Padding Counter These bits indicate the amount of padding to be stored in data areas for descriptors. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

Padding can be used to extend frame lengths, but frame lengths should not exceed 4 Kbytes.

DCNT[7:0] Stored Data Counter Bits

These bits specify the amount of the frame data (1 to 255) to be stored following the padding. Counting by one indicates one word (4 bytes). For example, when these bits are set to 47, the amount of data is 47 words (= 188 bytes).

When these bits are 0, all received data is stored following the initial padding.

Writing to the bits is only possible when the current operating mode is configuration mode. For details on padding, see **Section 26.4.4.3(2)(c), Padding**.

PCNT[2:0] Stored Padding Counter Bits

These bits specify the amount of padding (1 to 7) to be appended to the URAM. Counting by one indicates one word (4 bytes). For example, when these bits are set to 1, the amount of padding is one word (= 4 bytes).

Writing to the bits is only possible when the current operating mode is configuration mode. For details on padding, see **Section 26.4.4.3(2)(c), Padding**.

26.3.11 ETNBnUFCS — Unread Frame Counter Stop Level Configuration Register

The ETNBnUFCS register sets the stop levels for unread frames.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 00C0$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SL3[5:0]						—	—	SL2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SL1[5:0]						—	—	SL0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 26.27 ETNBnUFCS Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	SL3[5:0]	Stop Level 3 Unread frame count stop level 3
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	SL2[5:0]	Stop Level 2 Unread frame count stop level 2
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	SL1[5:0]	Stop Level 1 Unread frame count stop level 1
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	SL0[5:0]	Stop Level 0 Unread frame count stop level 0 The write value should be 0.

SL0 to SL3[5:0] Stop Level 0 to 3 Bits

These bits set the stop levels for unread frames.

One of the four stop levels from 0 to 3 can be set for each reception queue. When these bits are set to 0, the stop function is disabled. The level to be used is specified by the receive queue configuration register i (ETNBnRQCi) (i = 0 to 4).

Writing to the bits is only possible when the current operating mode is configuration mode.

26.3.12 ETNBnUFCVi — Unread Frame Counter Register i (i = 0 to 4)

The ETNBnUFCVi register indicates the number of unread frames in reception queues r (r = 0 + i × 4 to 3 + i × 4).*1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $00C4_H$ + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CVr[5:0](r = 3 + i × 4)*2						—	—	CVr[5:0](r = 2 + i × 4)*2					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CVr[5:0](r = 1 + i × 4)*2						—	—	CVr[5:0](r = 0 + i × 4)*2					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The ETNBnUFCV4 register corresponds to reception queues from 16 to 17.

Note 2. The ETNBnUFCV4 register corresponds to CV16[5:0] and CV17[5:0] only and bits from 16 to 31 are reserved bits.

Table 26.28 ETNBnUFCVi Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 24	CVr[5:0]	Unread Frame Count 3 + 4 × i Number of unread frames in reception queue 3 + 4 × i
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	CVr[5:0]	Unread Frame Count 2 + 4 × i Number of unread frames in reception queue 2 + 4 × i
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	CVr[5:0]	Unread Frame Count 1 + 4 × i Number of unread frames in reception queue 1 + 4 × i
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	CVr[5:0]	Unread Frame Count 0 + 4 × i Number of unread frames in reception queue 0 + 4 × i

CVr[5:0] (r = 0 to 17) Unread Frame Count r Bits

These bits indicate the number of unread frames in reception queue r.

The number of unread frames is decremented by the value that is written to the unread frame counter decrement register i (ETNBnUFCDi).

For a description of how to use unread frames, refer to **Section 26.4.4.4, Unread Frame Counters**.

[Conditions for Updating]

- The bits are set to 0 when the operating mode is not operation mode and when the descriptor base address load request register (ETNBnDLR) issues a base address load request.
- The number is incremented when data received in reception queue r are stored normally. The maximum increment is $3F_H$. If the value exceeds $3F_H$, incrementation will not proceed.)
- The number is decremented by the value written to the unread frame counter decrement register i (ETNBnUFCDi).

26.3.13 ETNBnUFCDi — Unread Frame Counter Decrement Register i (i = 0 to 4)

The ETNBnUFCDi register is used to decrement unread counters of reception queues r (r = 0 + i × 4 to 3 + i × 4).*1

Access: ETNBnUFCDi can be read or written in 32-bit units.

ETNBnUFCDiL and ETNBnUFCDiH can be read or written in 16-bit units.*2

ETNBnUFCDiLL, ETNBnUFCDiLH, ETNBnUFCDiHL, ETNBnUFCDiHH can be read or written in 8-bit units.*2

Address: ETNBnUFCDi: <ETNBn_base> + $00E0_H$ + i × 4_H

ETNBnUFCDiL: <ETNBn_base> + $00E0_H$ + i × 4_H

ETNBnUFCDiH: <ETNBn_base> + $00E0_H$ + i × 4_H + 2_H

ETNBnUFCDiLL: <ETNBn_base> + $00E0_H$ + i × 4_H

ETNBnUFCDiLH: <ETNBn_base> + $00E0_H$ + i × 4_H + 1_H

ETNBnUFCDiHL: <ETNBn_base> + $00E0_H$ + i × 4_H + 2_H

ETNBnUFCDiHH: <ETNBn_base> + $00E0_H$ + i × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	DVr[5:0] (r = 3 + i × 4) *3						—	—	DVr[5:0] (r = 2 + i × 4)*3					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DVr[5:0] (r = 1 + i × 4)*3						—	—	DVr[5:0] (r = 0 + i × 4)*3					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The ETNBnUFCD4 register corresponds to reception queues from 16 to 17.

Note 2. The ETNBnUFCD4 register corresponds to ETNBnUFCD4, ETNBnUFCD4L, ETNBnUFCD4LL, and ETNBnUFCD4LH only.

Note 3. The ETNBnUFCD4 register corresponds to DV16[5:0] and DV17[5:0] only and bits from 16 to 31 are reserved bits.

Table 26.29 ETNBnUFCDi Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	DVr[5:0]	Unread Frame Decrement Value 3 + 4 × i Unread frame decrement value for reception queue 3 + 4 × i
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	DVr[5:0]	Unread Frame Decrement Value 2 + 4 × i Unread frame decrement value for reception queue 2 + 4 × i
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	DVr[5:0]	Unread Frame Decrement Value 1 + 4 × i Unread frame decrement value for reception queue 1 + 4 × i
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	DVr[5:0]	Unread Frame Decrement Value 0 + 4 × i Unread frame decrement value for reception queue 0 + 4 × i

DVr[5:0] (r = 0 to 17) Unread Frame Decrement Value r Bits

These bits set the decrement value for unread frames in reception queue r. The value of an unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) is decremented by the value set in the corresponding bits of this register.

Write $3F_H$ to these bits to reset the unread counters in reception queue r. These bits are always read as 0.

26.3.14 ETNBnSFO — Separation Filter Offset Register

The ETNBnSFO register sets an offset into frames for use by the separation filter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 00FC$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FBP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.30 ETNBnSFO Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FBP[5:0]	First Byte Position Position of the first byte of Ethernet frames in the separation filter

FBP[5:0] First Byte Position Bits

These bits set the position of the first byte of Ethernet frames in the separation filter.

When these bits are 0, the separation filter starts from the top of each Ethernet frame (first byte of the destination address). For bytes in Ethernet frames, see **Figure 26.2, Data of Ethernet Frame Received**, in **Section 26.3.2, ETNBnCCC — AVB-DMAC Mode Register**.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 26.4.4.1(1), Separation Filtering**.

CAUTION

Received frames having fewer bytes than the setting of these bits + 8 bytes do not match the separation filter. In this case, the data will either be sorted into a reception queue or discarded in accord with the setting of the separation filtering select bits in the receive configuration register (ETNBnRCR.ESF).

26.3.15 ETNBnSFPi — Separation Filter Pattern Register i (i = 0 to 31)

The ETNBnSFPi register configures the pattern of the separation filter used for reception queues 2 to 17.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0100_H + i × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FPs[31:16]*2 (s = reception queue r – 2)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPs[15:0]*1 (s = reception queue r – 2)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. In this case, i is an even number. When i is an odd number, FPs[47:32]

Note 2. In this case, i is an even number. When i is an odd number, FPs[63:48]

Table 26.31 ETNBnSFPi Register Contents

Bit Position	Bit Name	Function
31 to 0	FPs[63:0]	Separation Filter Pattern These bits set the pattern of the separation filter. The 64-bit filter pattern is set for each queue.

FPs[63:0] (s = 0 to 15) Separation Filter Pattern Bits

These bits set the pattern for a separation filter to be used with reception queues 2 to 17 (for streams 0 to 15).

Each queue has space of 64-bit. Reception queue 2 (Stream 0) uses ETNBnSFP0 and ETNBnSFP1. Reception queue 17 (Stream 15) uses ETNBnSFP30 and ETNBnSFP31.

The separation filter passes a frame when, after masking by the mask value set in the separation filter mask register (ETNBnSFMi), data from received frames match the value defined in these bits.

ETNBnSFPi.FPs[7:0] (where i is an even number) are used for the byte of Ethernet frame data specified by the separation filter offset register, while ETNBnSFPi.FPs[63:56] (where i is the odd number) are used for the byte at the address specified by the separation filter offset register (ETNBnSFO) + 7.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 26.4.4.1(1), Separation Filtering**.

26.3.16 ETNBnSFMi — Separation Filter Mask Register i (i = 0 or 1)

The ETNBnSFMi register sets the mask value of the separation filter used for reception queue 2 to 17.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $01C0_H$ + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFM[31 + 32 × i: 16 + 32 × i]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFM[15 + 32 × i: 0 + 32 × i]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.32 ETNBnSFMi Register Contents

Bit Position	Bit Name	Function
31 to 0	CFM[63:0]	Separation Filter Mask These bits set the mask value for the separation filter.

CFM[63:0] Separation Filter Mask Bits

These bits set the mask value for the separation filter for use with the corresponding reception queue 2 to 17 (stream 0 to 15).

ETNBnSFM0.CFM[7:0] are used for bytes of Ethernet frame data specified by the separation filter offset register, while ETNBnSFM1.CFM[63:56] are used for the separation filter offset register (ETNBnSFO) + 7.

Frame data at the positions of mask bits that are set to 0 are masked; that is, they do not affect pattern- matching by the separation filter.

Writing to the bits is only possible when the current operating mode is configuration mode. For separation filtering, see **Section 26.4.4.1(1), Separation Filtering**.

26.3.17 ETNBnTGC — Transmit Configuration Register

The ETNBnTGC register configures the transmission-relating settings of the AVB-DMAC.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0300_H

Value after reset: $0022\ 2200_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	TBD3[1:0]		—	—	TBD2[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TBD1[1:0]		—	—	TBD0[1:0]		—	—	TQP[1:0]		TSM3	TSM2	TSM1	TSM0
Value after reset	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.33 ETNBnTGC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	TBD3[1:0]	Transmit FIFO Size (Stream Class A) Number of frames to be fetched from transmission queue 3 (for stream class A) **CAUTION:** Write 2 to these bits.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	TBD2[1:0]	Transmit FIFO Size (Stream Class B) Number of frames to be fetched from transmission queue 2 (for stream class B) **CAUTION:** Write 2 to these bits.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	TBD1[1:0]	Transmit FIFO Size (Network Control) Number of frames to be fetched from transmission queue 1 (for network control) **CAUTION:** Write 2 to these bits.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TBD0[1:0]	Transmit FIFO Size (Best Effort) Number of frames to be fetched from transmission queue 0 (for best effort) **CAUTION:** Write 2 to these bits.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	TQP[1:0]	Transmit Queue Priority 00: Non-AVB mode 01: AVB mode 1 10: Setting prohibited 11: AVB mode 2
3	TSM3	Transmit Synchronous Mode (Stream Class A) 0: With write-back 1: Setting prohibited
2	TSM2	Transmit Synchronous Mode (Stream Class B) 0: With write-back 1: Setting prohibited

Table 26.33 ETNBnTGC Register Contents

Bit Position	Bit Name	Function
1	TSM1	Transmit Synchronous Mode (Network Control) 0: With write-back 1: Setting prohibited
0	TSM0	Transmit Synchronous Mode (Best Effort) 0: With write-back 1: Setting prohibited

TBDt[1:0] (t = 0 to 3) Transmit FIFO Size (Stream Class A/ Stream Class B/Network Control/Best Effort) Bits

These bits set the sizes of the transmission FIFO buffers for use with each of the transmission queues. Writing to these bits is only possible when the current operating mode is configuration mode.

Set these bits to 2 (“10_B”).

TQP[1:0] Transmit Queue Priority Bits

These bits set the priority of the transmission queues.

00_B: Non-AVB mode: Q3 → Q2 → Q1 → Q0

01_B: AVB mode 1: Q3 (CBS) → Q2 (CBS) → Q1 → Q0

10_B: Setting prohibited

11_B: AVB mode 2: Q1 → Q3 (CBS) → Q2 (CBS) → Q0

For the credit-based shaping (CBS) algorithm, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

The CBS algorithm is invalidated in non-AVB mode (i.e. when the value is “00_B”).

Writing to the bits is only possible when the current operating mode is configuration mode.

TSM0 to TSM3 Transmit Synchronous Mode Bits

These bits set Transmit synchronous mode.

Writing to these bits is only possible when the current operating mode is configuration mode. Set these bits to 0.

26.3.18 ETNBnTCCR — Transmit Configuration Control Register

The ETNBnTCCR register controls transmission by the AVB-DMAC and is used to make related settings.

Access:	ETNBnTCCR can be read or written in 32-bit units. ETNBnTCCRL can be read or written in 16-bit units. ETNBnTCCRLL, ETNBnTCCRLH can be read or written in 8-bit units.
Address:	ETNBnTCCR: <ETNBn_base> + 0304_H ETNBnTCCRL: <ETNBn_base> + 0304_H ETNBnTCCRLL: <ETNBn_base> + 0304_H ETNBnTCCRLH: <ETNBn_base> + 0304_H + 1_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFR	TFEN	—	—	—	—	TSRQ3	TSRQ2	TSRQ1	TSRQ0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.34 ETNBnTCCR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TFR	Timestamp FIFO Release 0: (Not operating) 1: Releases the oldest entry in the timestamp FIFO.
8	TFEN	Timestamp FIFO Enable 0: Recording of transmission timestamps in the timestamp FIFO is disabled. 1: Recording of transmission timestamps in the timestamp FIFO is enabled.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	TSRQ3	Transmit Start Request (Queue 3 (Stream Class A)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.
2	TSRQ2	Transmit Start Request (Queue 2 (Stream Class B)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.
1	TSRQ1	Transmit Start Request (Queue 1 (Network Control)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.
0	TSRQ0	Transmit Start Request (Queue 0 (Best Effort)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.

TFR Timestamp FIFO Release Bit

This bit makes the timestamp FIFO release the oldest entry.

For a description of how to use the timestamp FIFO, see **Section 26.4.5.4, Timestamping in Transmission**.

TFEN Timestamp FIFO Enable Bit

This bit enables storage in the timestamp FIFO.

When it is set, timestamp information is stored for descriptors with DESCR.TSR set to 1 (for DESCR.TSR, see **Section 26.4.5.2(2), Configuration of Transmission Frame Data Descriptors**).

When 0 is set in this bit, all timestamp FIFO entries are invalidated.

For a description of how to use the timestamp FIFO, see **Section 26.4.5.4, Timestamping in Transmission**.

TSRQt (t = 0 to 3) Transmit Start Request (Queue t) Bit

This bit issues a request to start transmission for transmission queue t.

When read, this bit being set to 1 indicates that transmission queue t has a frame that has not yet been fetched to the transmission FIFO.

Frame transmission by the E-MAC is processed independently from fetching to the transmission FIFO. The timing of transmission from a queue depends on the priority order of transmission.

For the scheduling of transmission queues, see **Section 26.4.5.1, Transmission Modes**.

Writing to this bit is only possible when the current operating mode is configuration mode.

Only 1 can be written to the bit. Writing 0 to the bit has no effect.

[Conditions for Updating]

> The bit is set to 0 when the operating mode is not operation mode, when a descriptor of type EEMPTY, FEMPTY or LEMPTY (no usable data) is processed, when an EOS descriptor is processed, or when a descriptor with defective data is processed.

26.3.19 ETNBnTSR — Transmit Status Register

The ETNBnTSR register indicates the state of transmission by the AVB-DMAC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0308_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFFL[2:0]			—	—	—	—	CCS1[1:0]		CCS0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.35 ETNBnTSR Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10 to 8	TFFL[2:0]	Timestamp FIFO Count Number of timestamp FIFOs
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	CCS1[1:0]	CBS Counter Status 1 (Class A) 00_B: The current credit value is within the range. 01_B: The current credit value is less than or equal to the lower limit. 10_B: The current credit value is greater than or equal to the upper limit. 11_B: (Reserved)
1, 0	CCS0[1:0]	CBS Counter Status 0 (Class B) 00_B: The current credit value is within the range. 01_B: The current credit value is less than or equal to the lower limit. 10_B: The current credit value is greater than or equal to the upper limit. 11_B: (Reserved)

TFFL[2:0] Timestamp FIFO Count Bits

These bits indicate the number of timestamps in the timestamp FIFO.

The value 0 indicates it is empty and the value 2 indicates it is full (values 3 to 7 are reserved).

[Conditions for Updating]

- The bits are set to 0 when the operating mode is not operation mode or when the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) = 0.
- When the timestamp FIFO enable bit (ETNBnTCCR.TFEN) is 1 and these bits are not 2, the value of these bits is incremented after a frame with DESCR.TSR set has been transmitted by the E- MAC (for DESCR.TSR, see **Section 26.4.5.2(2), Configuration of Transmission Frame Data Descriptors**).

The value of these bits is decremented if it is not 0 when 1 is written to the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR).

CCS0 and CCS1[1:0] CBS Counter Status 0 and 1 Bits

These bits indicate the CBS (credit-based shaping) state of stream data transmission queues 0 and 1. If the calculated credit value is outside the range specified by CBS upper limit register c (ETNBnCULc) and CBS lower limit register c (ETNBnCLLc), it falls outside the range for CBS.

[Conditions for Updating]

- The bits are set to 00_B when the operating mode is not operation mode.
- When the determined credit value is within the range, 00_B is set.
- If the determined credit value is lower than the CBS lower limit register c (ETNBnCLLc), 01_B is set.
- If the determined credit value is higher than the upper limit register c (ETNBnCULc), 10_B is set.

26.3.20 ETNBnTFA0 — Timestamp FIFO Access Register 0

ETNBnTFA0 indicates the nano seconds portion of the timestamp value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0310_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.36 ETNBnTFA0 Register Contents

Bit Position	Bit Name	Function
31 to 0	TSV[31:0]	Timestamp Value

TSV[79:0] Timestamp Value Bits

These 80 bits consist of ETNBnTFA0.TSV[31:0], ETNBnTFA1.TSV[63:32], and ETNBnTFA2.TSV[79:64], which together indicate the oldest timestamp value stored in the timestamp FIFO.

Once the timestamp FIFO is full, no further timestamp values are stored.

[Conditions for Updating]

- The bits are set to $0000\ 0000_H$ when the operating mode is not operation mode.
- The register is updated whenever a value is stored in the timestamp FIFO (when the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) changes from 0 to 1).
- The register is updated when the oldest entry is released (when the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR) is set to 1).

26.3.21 ETNBnTFA1 — Timestamp FIFO Access Register 1

The ETNBnTFA1 register indicates the lower seconds portion of the timestamp value.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<ETNBn_base> + 0314$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSV[63:48]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.37 ETNBnTFA1 Register Contents

Bit Position	Bit Name	Function
31 to 0	TSV[63:32]	Timestamp Value

TSV[63:32] Timestamp Value Bits

For details, see **Section 26.3.20, ETNBnTFA0 — Timestamp FIFO Access Register 0**.

26.3.22 ETNBnTFA2 — Timestamp FIFO Access Register 2

The ETNBnTFA2 register indicates the timestamp tag and the higher seconds portion of the timestamp value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0318_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	TST[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[79:64]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.38 ETNBnTFA2 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25 to 16	TST[9:0]	Timestamp Tag
15 to 0	TSV[79:64]	Timestamp Value

TST[9:0] Timestamp Tag Bits

These bits indicate the contents of the DESCR.TAG bit within the descriptor for frame transmission. These values are used to check the correlation between frames within the transmission queue and the timestamp values (timestamp FIFO access register 0 to 2 (ETNBnTFA0 to 2)) which can be placed in the FIFO.

For transmit frame tagging, **Section 26.4.5.4, Timestamping in Transmission**.

[Conditions for Updating]

- The bits are set to 000_H when the operating mode is not operation mode.
- The bits are updated when a value is stored in the timestamp FIFO (when the value of the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) changes from 0 to 1).
- The bits are updated when the oldest entry has been released (1 is set in the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR)).

TSV[79:64] Timestamp Value Bits

For details, see **Section 26.3.20, ETNBnTFA0 — Timestamp FIFO Access Register 0**.

26.3.23 ETNBnCIVRc — CBS Increment Value Register c (c = 0 or 1)

The ETNBnCIVR0 register sets the increment in the CBS algorithm for transmission queue 2 (for stream class B) and the ETNBnCIVR1 register sets the increment in the CBS algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0320_H + c × 4_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CIV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.39 ETNBnCIVRc Register Contents

Bit Position	Bit Name	Function
31 to 0	CIV[31:0]	CBS Increment Value Setting value: 1 to 65535 ($0000\ 0001_H$ to $0000\ FFFF_H$)

CIV[31:0] CBS Increment Value Bits

These bits set the increment for the CBS algorithm.
Set a value in the range from 1 to 65535 ($0000\ 0001_H$ to $0000\ FFFF_H$).
The value to be written to these bits depends on the Ethernet bit rate and clk_chi (peripheral bus clock).

For details, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

26.3.24 ETNBnCDVRc — CBS Decrement Value Register c (c = 0 or 1)

The ETNBnCDVR0 register sets the decrement in the CBS algorithm for transmission queue 2 (for stream class B).

The ETNBnCDVR1 register sets the decrement in the CBS algorithm for transmission queue 3 (for stream class A).

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0328$_H$ + c × 4$_H$
Value after reset:	FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDV[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.40 ETNBnCDVRc Register Contents

Bit Position	Bit Name	Function
31 to 0	CDV[31:0]	CBS Decrement Value Setting value: –1 to –65536 (FFFF FFFF$_H$ to FFFF 0000$_H$)

CDV[31:0] CBS Decrement Value Bits

These bits set the decrement for the CBS algorithm.
Set a negative value from –1 to –65536 (FFFF FFFF$_H$ to FFFF 0000$_H$).
The value to be written to these bits depends on the Ethernet bit rate and clk_chi (peripheral bus clock).

For details, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

26.3.25 ETNBnCULc — CBS Upper Limit Register c (c = 0 or 1)

The ETNBnCUL0 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The ETNBnCUL1 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0330_H + c × 4_H

Value after reset: $7FFF\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULV[31:16]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ULV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.41 ETNBnCULc Register Contents

Bit Position	Bit Name	Function
31 to 0	ULV[31:0]	CBS Upper Limit Upper limit on CBS values

ULV[31:0] CBS Upper Limit Bits

These bits set the upper limit for credit values calculated by using the CBS algorithm.
The setting is a limiting value for error detection and does not normally affect operation of the algorithm.
Write a positive value to these bits.

For details, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

26.3.26 ETNBnCLLc — CBS Lower Limit Register c (c = 0 or 1)

The ETNBnCLL0 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The ETNBnCLL1 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0338_H + c × 4_H

Value after reset: 8000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LLV[31:16]															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.42 ETNBnCLLc Register Contents

Bit Position	Bit Name	Function
31 to 0	LLV[31:0]	CBS Lower Limit Lower limit on CBS values

LLV[31:0] CBS Lower Limit Bits

These bits set the lower limit for credit values calculated by using the CBS algorithm.
The setting is a limiting value for error detection and does not normally affect operation of the algorithm.
Write a negative value to these bits.

For details, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

26.3.27 ETNBnDIC — Descriptor Interrupt Control Register

The ETNBnDIC register is used to control descriptor interrupts 1 to 15.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0350$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPE15	DPE14	DPE13	DPE12	DPE11	DPE10	DPE9	DPE8	DPE7	DPE6	DPE5	DPE4	DPE3	DPE2	DPE1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 26.43 ETNBnDIC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	DPE15	Descriptor Interrupt Enable 15 0: Disabled 1: Enabled
14	DPE14	Descriptor Interrupt Enable 14 0: Disabled 1: Enabled
13	DPE13	Descriptor Interrupt Enable 13 0: Disabled 1: Enabled
12	DPE12	Descriptor Interrupt Enable 12 0: Disabled 1: Enabled
11	DPE11	Descriptor Interrupt Enable 11 0: Disabled 1: Enabled
10	DPE10	Descriptor Interrupt Enable 10 0: Disabled 1: Enabled
9	DPE9	Descriptor Interrupt Enable 9 0: Disabled 1: Enabled
8	DPE8	Descriptor Interrupt Enable 8 0: Disabled 1: Enabled
7	DPE7	Descriptor Interrupt Enable 7 0: Disabled 1: Enabled
6	DPE6	Descriptor Interrupt Enable 6 0: Disabled 1: Enabled

Table 26.43 ETNBnDIC Register Contents

Bit Position	Bit Name	Function
5	DPE5	Descriptor Interrupt Enable 5 0: Disabled 1: Enabled
4	DPE4	Descriptor Interrupt Enable 4 0: Disabled 1: Enabled
3	DPE3	Descriptor Interrupt Enable 3 0: Disabled 1: Enabled
2	DPE2	Descriptor Interrupt Enable 2 0: Disabled 1: Enabled
1	DPE1	Descriptor Interrupt Enable 1 0: Disabled 1: Enabled
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

DPEi (i = 1 to 15) Descriptor Interrupt Enable Bits

When interrupts are allowed, and if an interrupt source occurs (descriptor interrupt status bit of the descriptor interrupt status register (ETNBnDIS 1 to 15) = 1), an interrupt is generated.

26.3.28 ETNBnDIS — Descriptor Interrupt Status Register

The ETNBnDIS register indicates the state of descriptor interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0354_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPF15	DPF14	DPF13	DPF12	DPF11	DPF10	DPF9	DPF8	DPF7	DPF6	DPF5	DPF4	DPF3	DPF2	DPF1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 26.44 ETNBnDIS Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	DPF15	Descriptor Interrupt Status 15 0: The interrupt is not pending. 1: The interrupt is pending.
14	DPF14	Descriptor Interrupt Status 14 0: The interrupt is not pending. 1: The interrupt is pending.
13	DPF13	Descriptor Interrupt Status 13 0: The interrupt is not pending. 1: The interrupt is pending.
12	DPF12	Descriptor Interrupt Status 12 0: The interrupt is not pending. 1: The interrupt is pending.
11	DPF11	Descriptor Interrupt Status 11 0: The interrupt is not pending. 1: The interrupt is pending.
10	DPF10	Descriptor Interrupt Status 10 0: The interrupt is not pending. 1: The interrupt is pending.
9	DPF9	Descriptor Interrupt Status 9 0: The interrupt is not pending. 1: The interrupt is pending.
8	DPF8	Descriptor Interrupt Status 8 0: The interrupt is not pending. 1: The interrupt is pending.
7	DPF7	Descriptor Interrupt Status 7 0: The interrupt is not pending. 1: The interrupt is pending.
6	DPF6	Descriptor Interrupt Status 6 0: The interrupt is not pending. 1: The interrupt is pending.

Table 26.44 ETNBnDIS Register Contents

Bit Position	Bit Name	Function
5	DPF5	Descriptor Interrupt Status 5 0: The interrupt is not pending. 1: The interrupt is pending.
4	DPF4	Descriptor Interrupt Status 4 0: The interrupt is not pending. 1: The interrupt is pending.
3	DPF3	Descriptor Interrupt Status 3 0: The interrupt is not pending. 1: The interrupt is pending.
2	DPF2	Descriptor Interrupt Status 2 0: The interrupt is not pending. 1: The interrupt is pending.
1	DPF1	Descriptor Interrupt Status 1 0: The interrupt is not pending. 1: The interrupt is pending.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

DPFi (i = 1 to 15) Descriptor Interrupt Status Bits

When DESCR.DIE is 1 to 15 (0001_B to 1111_B), the corresponding bit indicates completion of the processing of a descriptor within the reception or transmission queue.

When DESCR.DIE is 0, the descriptor interrupt is not generated. Only 0 can be written to these bits.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 after a descriptor with DESCR.DIE set to the corresponding number from 1 to 15 (0001_B to 1111_B) is processed.

26.3.29 ETNBnEIC — Error Interrupt Control Register

The ETNBnEIC register controls the AVB-DMAC-related error interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0358$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TFFE	CULE1	CULE0	CLLE1	CLLE0	SEE	QEE	MTEE	MREE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.45 ETNBnEIC Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TFFE	Timestamp FIFO Full-Error Interrupt Enable 0: Disabled 1: Enabled
7	CULE1	CBS Upper Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
6	CULE0	CBS Upper Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
5	CLLE1	CBS Lower Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
4	CLLE0	CBS Lower Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
3	SEE	Separation Error interrupt Enable 0: Disabled 1: Enabled
2	QEE	Queue Error interrupt Enable 0: Disabled 1: Enabled
1	MTEE	MAC Transmission Error interrupt Enable 0: Disabled 1: Enabled
0	MREE	MAC Reception Error interrupt Enable 0: Disabled 1: Enabled

TFFE Timestamp FIFO Full-Error Interrupt Enable Bits

When the timestamp FIFO is full (the timestamp FIFO full error interrupt status bits of the error interrupt status register (ETNBnEIS.TFFF) = 1) and the interrupt is enabled, the interrupt is issued.

CULE1 CBS Upper Limit Error Interrupt Enable Bit (Class A)

When the CBS of Class A reached the upper limit value (CBS upper limit error interrupt status bit (Class A) (ETNBnEIS.CULF1) = 1) and the interrupt is enabled, an interrupt occurs.

CULE0 CBS Upper Limit Error Interrupt Enable Bit (Class B)

When the CBS of Class B reached the upper limit value (CBS upper limit error interrupt status bit (Class B) (ETNBnEIS.CULF0) = 1) and the interrupt is enabled, an interrupt occurs.

CLLE1 CBS Lower Limit Error Interrupt Enable Bit (Class A)

When the CBS of Class A reached the lower limit value (CBS lower limit error interrupt status bit (Class A) (ETNBnEIS.CLLF1) = 1) and the interrupt is enabled, an interrupt occurs.

CLLE0 CBS Lower Limit Error Interrupt Enable Bit (Class B)

When the CBS of Class B reached the lower limit value (CBS lower limit error interrupt status bit (Class B) (ETNBnEIS.CLLF0)=1) and the interrupt is enabled, an interrupt occurs.

SEE Separation Error interrupt Enable Bit

When the separation error flag (ETNBnEIS.SEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

QEE Queue Error interrupt Enable Bit

When the queue error flag (ETNBnEIS.QEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

MTEE MAC Transmission Error interrupt Enable Bit

When the MAC transmission error flag (ETNBnEIS.MTEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

MREE MAC Reception Error interrupt Enable Bit

When the MAC reception error flag (ETNBnEIS.MREF) is set to 1 and the interrupt is enabled, an interrupt occurs.

26.3.30 ETNBnEIS — Error Interrupt Status Register

The ETNBnEIS register indicates the states of AVB-DMAC-related error interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $035C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TFFF	CULF1	CULF0	CLLF1	CLLF0	SEF	QEF	MTEF	MREF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.46 ETNBnEIS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	QFS	Queue Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TFFF	Timestamp FIFO Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
7	CULF1	CBS Upper Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
6	CULF0	CBS Upper Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
5	CLLF1	CBS Lower Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
4	CLLF0	CBS Lower Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
3	SEF	Separation Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
2	QEF	Queue Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
1	MTEF	MAC Transmission Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.

Table 26.46 ETNBnEIS Register Contents

Bit Position	Bit Name	Function
0	MREF	MAC Reception Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.

QFS Queue Full Error Status Bit

With the interrupts enabled, this bit indicates that a queue is full (the receive queue r full interrupt status bit (QFFr) or the receive FIFO full interrupt status bit (RFFF) in receive interrupt status register 2 (ETNBnRIS2) = 1).

[Conditions for Changing]

- If the receive queue r full interrupt status bit (ETNBnRIS2.QFFr) and the receive queue r full interrupt enable bit in the receive interrupt control register 2 (ETNBnRIC2.QFEr) are updated, this bit is also updated.
- If the receive FIFO full interrupt status bit (ETNBnRIS2.RFFF) and the receive FIFO full interrupt enable bit (ETNBnRIC2.RFFE) are updated, this bit is also updated.

TFFF Timestamp FIFO Full-Error Interrupt Status Bit

This bit indicates that a new transmission timestamp has been discarded due to the timestamp FIFO being full (i.e. has reached the overrun state).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when a frame with DESCR.TSR set is transmitted while the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) is set to 1 and the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) is set to 2.

CULF1 CBS Upper Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has exceeded the set upper limit (ETNBnCUL1.ULV in the CBS upper limit register c (ETNBnCULc)).
Only 0 can be written to the bit.

[Conditions for Changing]

- This bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 1 when the value of the CBS counter status 1 (Class A) bits in the transmit status register (ETNBnTSR.CCS1) change from 00_B (within the range) to 10_B (indicating a value equal to or higher than the upper limit).

CULF0 CBS Upper Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has exceeded the set upper limit (ETNBnCUL0.ULV in the CBS upper limit register c (ETNBnCULc)).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 0 (Class B) (ETNBnTSR.CCS0) bit in the transmit status register changes from 00_B (within the range) to 10_B (indicating a value over the upper limit).

CLLF1 CBS Lower Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has fallen below the set lower limit (ETNBnCLL1.LLV in CBS lower limit register c (ETNBnCLLc)).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 1 (Class A) bit in the transmit status register (ETNBnTSR.CCS1) changes from 00_B (within the range) to 01_B (indicating a value less than the lower limit).

CLLF0 CBS Lower Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has fallen below the set lower limit (ETNBnCLL0.LLV in the CBS lower limit register c (ETNBnCLLc)).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register (ETNBnTSR.CCS0) changes from 00_B (within the range) to 01_B (indicating a value less than the lower limit).

SEF Separation Error Flag

This bit indicates that a received frame was discarded because it has not matched any configured separation filter for AVB stream data frames.
The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 1 when a valid AVB stream data frame was received by MAC but discarded because the separation filter select bits in the receive configuration register (ETNBnRCR.ESF) are 10_B and no separation filter has matched.

QEF Queue Error Flag

This bit indicates that an error has been detected while processing reception or transmit queue. Detail about the detected error is indicated by the error status register (ETNBnESR).

For the details of the error processing, **Section 26.4.2.3, Checking Integrity**.
The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when the operating mode changes from operation mode.
- This bit is set to 1 when a state of error is detected.

MTEF MAC Transmission Error Flag

This bit indicates that the MAC has detected a fault during transmission. For detail the MAC registers have to be checked.
The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when the operating mode changes from operation mode.
- This bit is set to 1 when MAC detects an error during frame transmission.

MREF MAC Reception Error Flag

This bit indicates that the MAC has detected a fault during reception.

For detail the MAC registers have to be checked.

NOTE

When the storage of faulty received frames (ETNBnRCR.EFFS) is enabled, the MAC error code (DESCR.MSC) is stored in the descriptor. By evaluating this information CPU can identify corrupted frames in URAM.

The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 1 when E-MAC detects an error during frame reception.

26.3.31 ETNBnRIC0 — Receive Interrupt Control Register 0

The ETNBnRIC0 register controls the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0360$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRE17	FRE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRE15	FRE14	FRE13	FRE12	FRE11	FRE10	FRE9	FRE8	FRE7	FRE6	FRE5	FRE4	FRE3	FRE2	FRE1	FRE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.47 ETNBnRIC0 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	FRE17	Receive Frame Interrupt Enable 17 (Stream 15) 0: Disabled 1: Enabled
16	FRE16	Receive Frame Interrupt Enable 16 (Stream 14) 0: Disabled 1: Enabled
15	FRE15	Receive Frame Interrupt Enable 15 (Stream 13) 0: Disabled 1: Enabled
14	FRE14	Receive Frame Interrupt Enable 14 (Stream 12) 0: Disabled 1: Enabled
13	FRE13	Receive Frame Interrupt Enable 13 (Stream 11) 0: Disabled 1: Enabled
12	FRE12	Receive Frame Interrupt Enable 12 (Stream 10) 0: Disabled 1: Enabled
11	FRE11	Receive Frame Interrupt Enable 11 (Stream 9) 0: Disabled 1: Enabled
10	FRE10	Receive Frame Interrupt Enable 10 (Stream 8) 0: Disabled 1: Enabled
9	FRE9	Receive Frame Interrupt Enable 9 (Stream 7) 0: Disabled 1: Enabled
8	FRE8	Receive Frame Interrupt Enable 8 (Stream 6) 0: Disabled 1: Enabled

Table 26.47 ETNBnRIC0 Register Contents

Bit Position	Bit Name	Function
7	FRE7	Receive Frame Interrupt Enable 7 (Stream 5) 0: Disabled 1: Enabled
6	FRE6	Receive Frame Interrupt Enable 6 (Stream 4) 0: Disabled 1: Enabled
5	FRE5	Receive Frame Interrupt Enable 5 (Stream 3) 0: Disabled 1: Enabled
4	FRE4	Receive Frame Interrupt Enable 4 (Stream 2) 0: Disabled 1: Enabled
3	FRE3	Receive Frame Interrupt Enable 3 (Stream 1) 0: Disabled 1: Enabled
2	FRE2	Receive Frame Interrupt Enable 2 (Stream 0) 0: Disabled 1: Enabled
1	FRE1	Receive Frame Interrupt Enable 1 (Network Control) 0: Disabled 1: Enabled
0	FRE0	Receive Frame Interrupt Enable 0 (Best Effort) 0: Disabled 1: Enabled

FREr (r = 0 to 17) Receive Frame Interrupt Enable Bits

When interrupts are allowed, and if an interrupt source occurs (receive interrupt status bit in the receive interrupt status register (ETNBnRIS0.FRF0 to 17) = 1), an interrupt is generated.

26.3.32 ETNBnRIS0 — Receive Interrupt Status Register 0

The ETNBnRIS0 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0364$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRF17	FRF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRF15	FRF14	FRF13	FRF12	FRF11	FRF10	FRF9	FRF8	FRF7	FRF6	FRF5	FRF4	FRF3	FRF2	FRF1	FRF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.48 ETNBnRIS0 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	FRF17	Receive Frame Interrupt Status 17 (Stream 15) 0: The interrupt is not pending. 1: The interrupt is pending.
16	FRF16	Receive Frame Interrupt Status 16 (Stream 14) 0: The interrupt is not pending. 1: The interrupt is pending.
15	FRF15	Receive Frame Interrupt Status 15 (Stream 13) 0: The interrupt is not pending. 1: The interrupt is pending.
14	FRF14	Receive Frame Interrupt Status 14 (Stream 12) 0: The interrupt is not pending. 1: The interrupt is pending.
13	FRF13	Receive Frame Interrupt Status 13 (Stream 11) 0: The interrupt is not pending. 1: The interrupt is pending.
12	FRF12	Receive Frame Interrupt Status 12 (Stream 10) 0: The interrupt is not pending. 1: The interrupt is pending.
11	FRF11	Receive Frame Interrupt Status 11 (Stream 9) 0: The interrupt is not pending. 1: The interrupt is pending.
10	FRF10	Receive Frame Interrupt Status 10 (Stream 8) 0: The interrupt is not pending. 1: The interrupt is pending.
9	FRF9	Receive Frame Interrupt Status 9 (Stream 7) 0: The interrupt is not pending. 1: The interrupt is pending.
8	FRF8	Receive Frame Interrupt Status 8 (Stream 6) 0: The interrupt is not pending. 1: The interrupt is pending.

Table 26.48 ETNBnRIS0 Register Contents

Bit Position	Bit Name	Function
7	FRF7	Receive Frame Interrupt Status 7 (Stream 5) 0: The interrupt is not pending. 1: The interrupt is pending.
6	FRF6	Receive Frame Interrupt Status 6 (Stream 4) 0: The interrupt is not pending. 1: The interrupt is pending.
5	FRF5	Receive Frame Interrupt Status 5 (Stream 3) 0: The interrupt is not pending. 1: The interrupt is pending.
4	FRF4	Receive Frame Interrupt Status 4 (Stream 2) 0: The interrupt is not pending. 1: The interrupt is pending.
3	FRF3	Receive Frame Interrupt Status 3 (Stream 1) 0: The interrupt is not pending. 1: The interrupt is pending.
2	FRF2	Receive Frame Interrupt Status 2 (Stream 0) 0: The interrupt is not pending. 1: The interrupt is pending.
1	FRF1	Receive Frame Interrupt Status 1 (Network Control) 0: The interrupt is not pending. 1: The interrupt is pending.
0	FRF0	Receive Frame Interrupt Status 0 (Best Effort) 0: The interrupt is not pending. 1: The interrupt is pending.

FRFr (r = 0 to 17) Receive Frame Interrupt Status Bits

Each bit indicates that a corresponding frame has been stored normally in reception queues 0 to 17 and that data is queued for CPU processing.
Only 0 can be written to the bit.

[Conditions for Changing]

- A bit is set to 0 when the operating mode is not operation mode.
- A bit is set to 0 when a value is written to the unread frame counter decrement register i (ETNBnUFCDi) (i = 0 to 4), and this decrements the value of unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) to 0.
- When a frame is stored normally in a reception queue, the corresponding bit is set to 1.

26.3.33 ETNBnRIC1 — Receive Interrupt Control Register 1

The ETNBnRIC1 register controls AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0368_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.49 ETNBnRIC1 Register Contents

Bit Position	Bit Name	Function
31	RFWE	Receive FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
30 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RFWE Receive FIFO Warning Interrupt Enable Bit

If the reception FIFO reaches the warning level (the value set in the receive FIFO warning level bits in the receive configuration register (ETNBnRCR.RFCL)) with the corresponding interrupt enabled, an interrupt is issued.

26.3.34 ETNBnRIS1 — Receive Interrupt Status Register 1

The ETNBnRIS1 register indicates the states of AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $036C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.50 ETNBnRIS1 Register Contents

Bit Position	Bit Name	Function
31	RFWF	Receive FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
30 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RFWF Receive FIFO Warning Interrupt Status Bit

This bit indicates that the reception FIFO has reached the set caution level (the value set in the receive FIFO warning level bits in the receive configuration register (ETNBnRCR.RFCL)).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the reception FIFO reaches the set caution level (the value set in the receive FIFO warning level bits in the receive configuration register (ETNBnRCR.RFCL)).

26.3.35 ETNBnRIC2 — Receive Interrupt Control Register 2

The ETNBnRIC2 register controls AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0370_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFE	—	—	—	—	—	—	—	—	—	—	—	—	—	QFE17	QFE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFE15	QFE14	QFE13	QFE12	QFE11	QFE10	QFE9	QFE8	QFE7	QFE6	QFE5	QFE4	QFE3	QFE2	QFE1	QFE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.51 ETNBnRIC2 Register Contents

Bit Position	Bit Name	Function
31	RFFE	Receive FIFO Full Interrupt Enable 0: Disabled 1: Enabled
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	QFE17	Receive Queue 17 (Stream 15) Full Interrupt Enable 0: Disabled 1: Enabled
16	QFE16	Receive Queue 16 (Stream 14) Full Interrupt Enable 0: Disabled 1: Enabled
15	QFE15	Receive Queue 15 (Stream 13) Full Interrupt Enable 0: Disabled 1: Enabled
14	QFE14	Receive Queue 14 (Stream 12) Full Interrupt Enable 0: Disabled 1: Enabled
13	QFE13	Receive Queue 13 (Stream 11) Full Interrupt Enable 0: Disabled 1: Enabled
12	QFE12	Receive Queue 12 (Stream 10) Full Interrupt Enable 0: Disabled 1: Enabled
11	QFE11	Receive Queue 11 (Stream 9) Full Interrupt Enable 0: Disabled 1: Enabled
10	QFE10	Receive Queue 10 (Stream 8) Full Interrupt Enable 0: Disabled 1: Enabled
9	QFE9	Receive Queue 9 (Stream 7) Full Interrupt Enable 0: Disabled 1: Enabled

Table 26.51 ETNBnRIC2 Register Contents

Bit Position	Bit Name	Function
8	QFE8	Receive Queue 8 (Stream 6) Full Interrupt Enable 0: Disabled 1: Enabled
7	QFE7	Receive Queue 7 (Stream 5) Full Interrupt Enable 0: Disabled 1: Enabled
6	QFE6	Receive Queue 6 (Stream 4) Full Interrupt Enable 0: Disabled 1: Enabled
5	QFE5	Receive Queue 5 (Stream 3) Full Interrupt Enable 0: Disabled 1: Enabled
4	QFE4	Receive Queue 4 (Stream 2) Full Interrupt Enable 0: Disabled 1: Enabled
3	QFE3	Receive Queue 3 (Stream 1) Full Interrupt Enable 0: Disabled 1: Enabled
2	QFE2	Receive Queue 2 (Stream 0) Full Interrupt Enable 0: Disabled 1: Enabled
1	QFE1	Receive Queue 1 (Network Control) Full Interrupt Enable 0: Disabled 1: Enabled
0	QFE0	Receive Queue 0 (Best Effort) Full Interrupt Enable 0: Disabled 1: Enabled

RFFE Receive FIFO Full Interrupt Enable Bit

When interrupts are allowed, and if the reception FIFO becomes full (the reception FIFO full interrupt status bit (ETNBnRIS2.RFFF) = 1 in receive interrupt status register 2), an interrupt is generated.

QFEr (r = 0 to 17) Receive Queue r Full Interrupt Enable Bits

When interrupts are allowed, and if a receive queue (0 to 17) becomes full (the receive queue r full interrupt status bit (ETNBnRIS2.QFF0 to 17) = 1 in receive interrupt status register 2), an interrupt is generated.

26.3.36 ETNBnRIS2 — Receive Interrupt Status Register 2

The ETNBnRIS2 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0374_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFF	—	—	—	—	—	—	—	—	—	—	—	—	—	QFF17	QFF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFF15	QFF14	QFF13	QFF12	QFF11	QFF10	QFF9	QFF8	QFF7	QFF6	QFF5	QFF4	QFF3	QFF2	QFF1	QFF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.52 ETNBnRIS2 Register Contents

Bit Position	Bit Name	Function
31	RFFF	Receive FIFO Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	QFF17	Receive Queue 17 (Stream 15) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
16	QFF16	Receive Queue 16 (Stream 14) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
15	QFF15	Receive Queue 15 (Stream 13) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
14	QFF14	Receive Queue 14 (Stream 12) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
13	QFF13	Receive Queue 13 (Stream 11) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
12	QFF12	Receive Queue 12 (Stream 10) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
11	QFF11	Receive Queue 11 (Stream 9) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
10	QFF10	Receive Queue 10 (Stream 8) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
9	QFF9	Receive Queue 9 (Stream 7) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

Table 26.52 ETNBnRIS2 Register Contents

Bit Position	Bit Name	Function
8	QFF8	Receive Queue 8 (Stream 6) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
7	QFF7	Receive Queue 7 (Stream 5) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
6	QFF6	Receive Queue 6 (Stream 4) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
5	QFF5	Receive Queue 5 (Stream 3) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
4	QFF4	Receive Queue 4 (Stream 2) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
3	QFF3	Receive Queue 3 (Stream 1) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
2	QFF2	Receive Queue 2 (Stream 0) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
1	QFF1	Receive Queue 1 (Network Control) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
0	QFF0	Receive Queue 0 (Best Effort) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

RFFF Receive FIFO Full Interrupt Status Bit

This bit indicates that a frame was received but storing it was not possible due to the reception FIFO being full.

When receiving a frame is not possible, the frame will be discarded.

Other information regarding discarded frames is not retained. Even if the frame is not discarded, this bit may also be set to 1 if the E-MAC determines that the frame is an error frame
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the reception FIFO cannot hold received frame data.

QFFr (r = 0 to 17) Receive r Full Interrupt Status Bits

When a received frame is dropped due to non-availability of empty descriptor, ETNBnRIS2.QFFr is correctly set to 1.

Additionally, when ETNBnUFCVi.CVr reached the configured stop level (ETNBnUFCS.SLj), the queue full flag (ETNBnRIS2.QFFr) is set to 1 even before any further received frame is dropped.

CAUTION

If no FEMPTY descriptors or no empty space for descriptors remains in the queue during storing of a divided frame (see **Section 26.4.4.3(1)(b), Storing Frame Data as Divided Frames**) an error frame is stored in the queue. Such error frames are treated as descriptor sequence errors.

[Conditions for Changing]

- A bit is set to 0 when the operating mode is not operation mode.
- A bit is set to 1 when reception queue r has no space available for storage.
- A bit is set to 1 when the unread frame counter (unread frame counter register i (ETNBnUFCVi) (i = 0 to 4)) reaches the set level for stopping.

26.3.37 ETNBnTIC — Transmit Interrupt Control Register

The ETNBnTIC register controls the AVB-DMAC transmit interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0378$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFWE	TFUE	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 26.53 ETNBnTIC Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TFWE	Timestamp FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
8	TFUE	Timestamp FIFO Update Interrupt Enable 0: Disabled 1: Enabled
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

TFWE Timestamp FIFO Warning Interrupt Enable Bit

When the timestamp FIFO reaches the warning level while the interrupt is enabled, the interrupt is issued.

TFUE Timestamp FIFO Update Interrupt Enable Bit

When the timestamp FIFO is updated while the interrupt is enabled, the interrupt is issued.

26.3.38 ETNBnTIS — Transmit Interrupt Status Register

The ETNBnTIS register indicates the states of the AVB-DMAC transmit interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 037C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFWF	TFUF	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 26.54 ETNBnTIS Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11, 10	Reserved	When read, an undefined value is returned. When writing, write the value after reset.
9	TFWF	Timestamp FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The timestamp FIFO has reached the warning level.
8	TFUF	Timestamp FIFO Update Interrupt Status 0: The interrupt is not pending. 1: The timestamp FIFO has been updated.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	Reserved	When read, an undefined value is returned. When writing, write the value after reset.

TFWF Timestamp FIFO Warning Interrupt Status Bit

This bit indicates that the transmission timestamp FIFO has reached the warning level.
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode, when the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) is 0, or when 1 is written to the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR).
- The bits are set to “1” after a frame including DESCR.TSR set has been transmitted and one entry has already been stored (the transmit status register timestamp FIFO count bit (ETNBnTSR.TFFL) is set to “1”) in the timestamp FIFO.

TFUF Timestamp FIFO Update Interrupt Status Bit

This bit indicates that the transmission timestamp FIFO has been updated.
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode, when the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) is 0, or when 1 is written to the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR).
- The bit is set to 1 when the timestamp FIFO enable bit (ETNBnTCCR.TFEN) is 1 after a frame including DESCR.TSR set has been transmitted.

26.3.39 ETNBnISS — Interrupt Summary Status Register

The ETNBnISS register gives a summary of the states of AVB-DMAC-related interrupts.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<ETNBn_base> + 0380$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DPS15	DPS14	DPS13	DPS12	DPS11	DPS10	DPS9	DPS8	DPS7	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CGIS	RFWS	—	—	TFWS	TFUS	MS	ES	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.55 ETNBnISS Register Contents

Bit Position	Bit Name	Function
31	DPS15	Descriptor Interrupt 15 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
30	DPS14	Descriptor Interrupt 14 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
29	DPS13	Descriptor Interrupt 13 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
28	DPS12	Descriptor Interrupt 12 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
27	DPS11	Descriptor Interrupt 11 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
26	DPS10	Descriptor Interrupt 10 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
25	DPS9	Descriptor Interrupt 9 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
24	DPS8	Descriptor Interrupt 8 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
23	DPS7	Descriptor Interrupt 7 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
22	DPS6	Descriptor Interrupt 6 Summary 0: The interrupt is not pending. 1: The interrupt is pending.

Table 26.55 ETNBnISS Register Contents

Bit Position	Bit Name	Function
21	DPS5	Descriptor Interrupt 5 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
20	DPS4	Descriptor Interrupt 4 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
19	DPS3	Descriptor Interrupt 3 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
18	DPS2	Descriptor Interrupt 2 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
17	DPS1	Descriptor Interrupt 1 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
16 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	CGIS	gPTP Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
12	RFWS	Receive FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
11, 10	Reserved	When read, an undefined value is returned. When writing, write the value after reset.
9	TFWS	Timestamp FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
8	TFUS	Timestamp FIFO Update Interrupt 0: The interrupt is not pending. 1: The interrupt is pending.
7	MS	E-MAC Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
6	ES	Error Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	Reserved	When read, an undefined value is returned. When writing, write the value after reset.

DPSi (i = 1 to 15) Descriptor Interrupt 1 to 15 Summary Bits

These bits are set to 1 when the given descriptor interrupt enable bit (ETNBnDIC.DPE1 to ETNBnDIC.DPE15) and descriptor interrupt status flag (ETNBnDIS.DPF1 to ETNBnDIS.DPF15) are both 1.

CGIS gPTP Interrupt Summary Bit

This bit is set to 1 when either interrupt-related bit in the two gPTP-related interrupt registers (ETNBnGIC and ETNBnGIS) is 1.

RFWS Receive FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the receive FIFO warning interrupt enable bit (ETNBnRIC1.RFWE) and receive FIFO warning interrupt status flag (ETNBnRIS1.RFWF) are both 1.

TFWS Timestamp FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the timestamp FIFO warning interrupt enable bit (ETNBnTIC.TFWE) and timestamp FIFO warning interrupt status flag (ETNBnTIS.TFWF) are both 1.

TFUS Timestamp FIFO Update Interrupt Summary Bit

This bit is set to 1 when the timestamp FIFO update interrupt enable bit (ETNBnTIC.TFUE) and timestamp FIFO update interrupt status flag (ETNBnTIS.TFUF) are both 1.

MS E-MAC Interrupt Summary Bit

This bit is set to 1 when an E-MAC interrupt is issued.

ES Error Interrupt Summary Bit

This bit is set to 1 when any of the valid bits in the error interrupt status register (ETNBnEIS) is 1 or the queue full error interrupt status bit (ETNBnEIS.QFS) in the error interrupt status register (ETNBnEIS) is 1.

26.3.40 ETNBnGCCR — gPTP Configuration Control Register

The ETNBnGCCR register is used to set and control the gPTP (generalized precision time protocol).

Access:	ETNBnGCCR can be read or written in 32-bit units. ETNBnGCCRL can be read or written in16-bit units. ETNBnGCCRLL, ETNBnGCCRLH can be read or written in 8-bit units.
Address:	ETNBnGCCR: <ETNBn_base> + 0390_H ETNBnGCCRL: <ETNBn_base> + 0390_H ETNBnGCCRLL: <ETNBn_base> + 0390_H ETNBnGCCRLH: <ETNBn_base> + 0390_H + 1_H
Value after reset:	0000 $003C_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TCSS[1:0]		—	—	LMTT	LPTC	LTI	LTO	TCR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.56 ETNBnGCCR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TCSS[1:0]	Timer Capture Source Select 00_B: gPTP timer value 01_B: Adjusted gPTP timer value 10_B: AVTP presentation time 11_B: Setting prohibited
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LMTT	Maximum Transit Time Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
4	LPTC	Presentation Time Compare Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
3	LTI	Timer Increment Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
2	LTO	Timer Offset Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.

Table 26.56 ETNBnGCCR Register Contents

Bit Position	Bit Name	Function
1, 0	TCR[1:0]	Timer Control Request 00_B: Timer control is not requested. 01_B: gPTP/AVTP presentation time reset 10_B: Setting prohibited 11_B: Captures the value set in the TCSS bit.

TCSS[1:0] Timer Capture Source Select Bits

These bits select the source used for updating the captured timer register (gPTP timer capture register (ETNBnGCTi.CTV)).

These bits should still be controlled when timer control is not being requested (ETNBnGCCR.TCR = 00_B).

LMTT Maximum Transit Time Configuration Request Bit

This bit issues requests for configuring the gPTP maximum transit time configuration register (ETNBnGMTT).
Only 1 can be written to the bit.

[Conditions for Changing]

- The bit is set to 1 when the operating mode is not operation mode.
- The bit is set to 0 when the value of the gPTP maximum transit time configuration register (ETNBnGMTT) is loaded.

LPTC Presentation Time Compare Value Configuration Request Bit

This bit issues requests for configuring the gPTP presentation time comparison register (ETNBnGPTC).
Only 1 can be written to the bit.

[Conditions for Changing]

- The bit is set to 1 when the operating mode is not operation mode.
- The bit is set to 0 when the value of the gPTP presentation time comparison register (ETNBnGPTC) is loaded.

LTI Timer Increment Value Configuration Request Bit

This bit issues requests for configuring the gPTP timer increment configuration register (ETNBnGTI).
Only 1 can be written to the bit.

[Conditions for Changing]

- The bit is set to 1 when the operating mode is not operation mode.
- The bit is set to 0 when the value of the gPTP timer increment configuration register (ETNBnGTI) is loaded.

LTO Timer Offset Value Configuration Request Bit

This bit issues requests for configuring gPTP timer offset configuration register i (ETNBnGTOi).
Only 1 can be written to the bit.

[Conditions for Changing]

- The bit is set to 1 when the operating mode is not operation mode.
- The bit is set to 0 when the value of gPTP timer offset configuration register i (ETNBnGTOi) is loaded.

TCR[1:0] Timer Control Request Bits

These bits issue requests for controlling the gPTP timer.

Writing to the bits is only possible when the current operating mode is operation mode.

Do not write to the bits when the gPTP timer clock select bit in the AVB-DMAC mode register is 00_B.

[Conditions for Changing]

The bits are set to 00_B when the operating mode is not operation mode and on completion of the requested processing.

26.3.41 ETNBnGMTT — gPTP Maximum Transit Time Configuration Register

The ETNBnGMTT register sets the maximum time for transitions of the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0394_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MTTV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MTTV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.57 ETNBnGMTT Register Contents

Bit Position	Bit Name	Function
31 to 0	MTTV[31:0]	Maximum Transit Time The maximum transition time for addition to the presentation time

MTTV[31:0] Maximum Transit Time Bits

These bits set the maximum transition time for use in calculating AVTP presentation times.

Write the desired setting to the bits, then issue the configuration request by setting the maximum transit time configuration request bit in the gPTP configuration control register (ETNBnGCCR.LMTT) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the maximum transit time configuration request bit (ETNBnGCCR.LMTT) is 1.

26.3.42 ETNBnGPTC — gPTP Presentation Time Comparison Register

The ETNBnGPTC register sets a value for comparison with presentation times in the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0398_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PTCV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.58 ETNBnGPTC Register Contents

Bit Position	Bit Name	Function
31 to 0	PTCV[31:0]	Presentation Time Comparison Value Value for comparison with the gPTP presentation times

PTCV[31:0] Presentation Time Comparison Value Bits

These bits set a value for comparison with AVTP timer values to which a maximum transit time is not appended.

Write the desired setting to the bits, then issue the configuration request by setting the presentation time comparison value configuration request bit in the gPTP configuration control register (ETNBnGCCR.LPTC) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the presentation time comparison value configuration request bit (ETNBnGCCR.LPTC) is 1.
Do not write the range of “x – 1 to x + 1” to the bits. (x is a value of ETNBnGTI.TIV[27:0])

26.3.43 ETNBnGTI — gPTP Timer Increment Configuration Register

The ETNBnGTI register sets the increment for the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $039C_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TIV[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.59 ETNBnGTI Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 0	TIV[27:0]	gPTP Timer Increment Value Increment for the gPTP timer

TIV[27:0] Bits (gPTP Timer Increment Value)

When the gPTP clock select bits in the AVB-DMAV mode register (ETNBnCCC.CSEL) are selecting a clock signal, these bits set the value by which the timer is incremented each time a cycle of that clock signal elapses.

Write the desired setting to the bits, then issue the configuration request by setting the timer increment value configuration request bit in the gPTP configuration control register (ETNBnGCCR.LTI) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer increment value configuration request bit (ETNBnGCCR.LTI) is 1.
Do not write 0 to all bits.

26.3.44 ETNBnGTOi — gPTP Timer Offset Configuration Register i (i = 0 to 2)

The ETNBnGTOi register sets an offset value for the gPTP timer.

The offset value is added to the combination of bits 0 to 31 in ETNBnGTO0, 32 to 63 in ETNBnGTO1, and 64 to 79 in ETNBnGTO2, which together make up the gPTP timer.

Access:	This register can be read or written in 32-bit units.
Address:	ETNBnGTO0: <ETNBn_base> + 03A0$_\text{H}$ ETNBnGTO1: <ETNBn_base> + 03A4$_\text{H}$ ETNBnGTO2: <ETNBn_base> + 03A8$_\text{H}$
Value after reset:	0000 0000$_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOV[31 + 32 × i:16 + 32 × i][*1]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOV[15 + 32 × i:0 + 32 × i][*1]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ETNBnGTO2 register corresponds to TOV[79:64] only and bits from 16 to 31 are reserved bits.

Table 26.60 ETNBnGTOi Register Contents

Bit Position	Bit Name	Function
31 to 0	TOV[95:0]	Timer Offset Value Offset value for the gPTP timer

TOV[79:0] Timer Offset Value Bits

This is an 80-bit value consisting of the settings in ETNBnGTO0.TOV[31:0], ETNBnGTO1.TOV[63:32], and ETNBnGTO2.TOV[79:64], and is used to set an offset for adding to the value of the gPTP timer.

Write the desired setting to the bits, then issue the configuration request by setting the timer offset value configuration request bit in the gPTP configuration control register (GCCR.LTO) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer offset value configuration request bit (ETNBnGCCR.LTO) is 1.
Write 0000$_\text{H}$ to ETNBnGTO2.TOV[95:80].
Set a value in the range from 0 to 10^9-1 (0000 0000$_\text{H}$ to 3B9A C9FF$_\text{H}$) in ETNBnGTO0.TOV[31:0].

26.3.45 ETNBnGIC — gPTP Interrupt Control Register

The ETNBnGIC register is used to control gPTP-related interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 03AC$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTME	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 26.61 ETNBnGIC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PTME	Presentation Time Match Interrupt Enable 0: Disabled 1: Enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

PTME Presentation Time Match Interrupt Enable Bit

When this bit is 1, setting of the presentation time match interrupt flag in the gPTP interrupt status register (ETNBnGIS.PTMF) to 1 leads to generation of that interrupt.

26.3.46 ETNBnGIS — gPTP Interrupt Status Register

The ETNBnGIS register indicates the state of the gPTP-related interrupt.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + $03B0_H$
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTMF	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 26.62 ETNBnGIS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PTMF	Presentation Time Match Interrupt Flag 0: The interrupt is not pending. 1: The interrupt is pending.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

PTMF Presentation Time Match Interrupt Flag Bit

This bit indicates that the value of the AVTP timer exceeds the value of the gPTP presentation time comparison register (ETNBnGPTC).
Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the AVTP timer value is greater than or equal to the value of the gPTP presentation time comparison register (ETNBnGPTC).

26.3.47 ETNBnGCTi — gPTP Timer Capture Register i (i = 0 to 2)

The ETNBnGCTi registers form an 80-bit register that captures the gPTP timer value.

Access: This register can be read or written in 32-bit units.

Address: ETNBnGCT0: <ETNBn_base> + 03B8$_H$
ETNBnGCT1: <ETNBn_base> + 03BC$_H$
ETNBnGCT2: <ETNBn_base> + 03C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CTV[31 + 32 × i:16 + 32 × i][*1]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTV[15 + 32 × i:0 + 32 × i][*1]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ETNBnGCT2 register corresponds to CTV[79:64] only and bits from 16 to 31 are reserved bits.

Table 26.63 ETNBnGCTi Register Contents

Bit Position	Bit Name	Function
31 to 0	CTV[95:0]	gPTP Timer Capture Value Captured timer value

CTV[79:0] gPTP Timer Capture Value Bits

These 80 bits consist of ETNBnGCT0.CTV[31:0], ETNBnGCT1.CTV[63:32] and ETNBnGCT2.CTV[79:64], which together indicate captured timer values.

When 00$_B$ (value of the gPTP timer) or 01$_B$ (adjusted gPTP timer value) is selected by the timer capture source select bits in the gPTP configuration control register (ETNBnGCCR.TCSS), the corresponding 80-bit values are stored in these bits.

When 10$_B$ (AVTP presentation time) is selected by the timer capture source select bits (ETNBnGCCR.TCSS), the corresponding 32-bit values are stored in these bits (ETNBnGCT0.CTV[31:0]).

Actual writing of the timer value specified by the timer capture source select bits (ETNBnGCCR.TCSS) proceeds when 11$_B$ (timer capture request) is written to the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR).

Do not read the value while the value of the timer control request bits (ETNBnGCCR.TCR) is 11$_B$ because this indicates that storage is still in progress.

CAUTION

Write 0000$_H$ to ETNBnGCT2.CTV[95:80].

26.3.48 ETNBnECMR — E-MAC Mode Register

ETNBnECMR is used to specify the operating mode of the E-MAC. The settings in this register are normally made in the initialization process following a reset.

The operating mode settings must not be changed while the transmission and reception functions are enabled (i.e. while the RE or TE bit in this register is 1).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0500_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	RE	TE	—	—	—	DM	PRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W

Table 26.64 ETNBnECMR Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TRCCM	Counter Clear Mode 0: Writing to the counter register leads to the register being cleared to 0. 1: Reading from the counter register leads to the register being cleared to 0.
25, 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	RCSC	Checksum Calculation 0: Checksums are not automatically calculated. 1: Checksums are automatically calculated.
22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	DPAD	Data Padding 0: Padding to make up 60 bytes is inserted in data for transmission when fewer than 60 bytes are to be transmitted. 1: Padding is not inserted in data for transmission when fewer than 60 bytes are to be transmitted and the data are transmitted without being changed.
20	RZPF	PAUSE Frame Reception with Time = 0 0: Reception of PAUSE frames with the TIME parameter value 0 is disabled. 1: Reception of PAUSE frames with the TIME parameter value 0 is enabled.
19	ZPF	PAUSE Frame Usage with TIME = 0 Enable ● PAUSE frame usage with TIME = 0 enable (in full-duplex mode) 0: Control in response to and for the sending of PAUSE frames with the TIME parameter value 0 is disabled. 1: Control in response to and for the sending of PAUSE frames with the TIME parameter value is 0 is enabled.

Table 26.64 ETNBnECMR Register Contents

Bit Position	Bit Name	Function
18	PFR	PAUSE Frame Receive Mode 0: PAUSE frames are not transferred to the AVB-DMAC. 1: PAUSE frames are transferred to the AVB-DMAC.
17	RXF	Reception Flow Control Operation Mode 0: Flow control for the receiving port (reception of PAUSE frame) is disabled. 1: Flow control for the receiving port (reception of PAUSE frame) is enabled.
16	TXF	Transmission Flow Control Operation Mode 0: Flow control for the transmitting port is disabled (PAUSE frames are not automatically transmitted). 1: Flow control for the transmitting port is enabled (PAUSE frames are automatically transmitted as required).
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	RE	Reception Enable 0: Reception is disabled. 1: Reception is enabled.
5	TE	Transmission Enable 0: Transmission is disabled. 1: Transmission is enabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DM	When Ethernet AVB is used, this bit must be set to 1 (the value after reset is 0).
0	PRM	Promiscuous Mode 0: Normal operation 1: Promiscuous mode operation

TRCCM Counter Clear Mode Bit

This bit sets the method for clearing the counter register. Refer to the descriptions of the counter registers.

RCSC Checksum Calculation Bit

Setting this bit to 1 enables automatic calculation of checksums for the data field in received frames.

Only the data field of an Ethernet frame is in the scope of checksum calculation. Specifically, the checksum is calculated from the data field, which follows the length/type field and is followed by the CRC field. Calculation only involves 16-bit addition; it does not involve bit inversion.

DPAD Data Padding Control Bit

This bit specifies padding or non-padding of data when less than 60 bytes are to be transmitted.

When this bit is set to 1, data are transmitted without padding; when it is set to 0, data are padded to make up 60-byte units for transmission.

RZPF (PAUSE Frame Reception with Time = 0) Bit

This bit is set to 0, received PAUSE frames with the Timer value 0 are discarded.

This bit is set to 1, release from the transmission wait state follows reception of a PAUSE frame with the Timer value 0.

ZPF (PAUSE Frame Usage with TIME = 0 Enable) Bit

- PAUSE frame usage with TIME = 0 enable (In full-duplex mode)

  When this bit is set to 0, the next frame to be transmitted is not transmitted until the time specified by the Timer value has elapsed.
  When this bit is set to "1", if the amount of data in the reception FIFO becomes less than the setting of the receive FIFO warning level bits in the receive configuration register (ETNBnRCR.RFCL) before the time specified by the Timer value elapses, a PAUSE frame with a Timer value of 0 is automatically transmitted. If the interface is in the transmission wait state, it is released from that state on receiving a PAUSE frame with a Timer value of 0.

PFR PAUSE Frame Receive Mode Bit

This bit specifies whether PAUSE frames are transferred to the AVB-DMAC.

RXF (Operating Mode for Flow Control in Reception) Bit

This bit is set to 1 and a PAUSE frame is received, a next frame to be transmitted is not transmitted until the time indicated by the Timer value in the PAUSE frame has elapsed. However, the transmission of a current frame is continued. The number of received PAUSE frames is also counted. For details, see **Section 26.3.57, ETNBnPFRCR — PAUSE Frame Receive Counter**.

Setting this bit to 0 disables PAUSE frame detection.

TXF (Operating Mode for Flow Control in Transmission) Bit

This bit enables or disables flow control in transmission. Setting this bit to 0 disables PAUSE frame detection.

RE Reception Enable Bit

When this bit is switched from the receive function enabled (RE = 1) to the receive function disabled (RE = 0), and if there is a frame is being received, the receive function remains enabled until the reception of the frame finishes.

TE Transmission Enable Bit

When this bit is switched from the transmit function enabled (TE = 1) to the transmit function disabled (TE = 0), and if there is a frame being transmitted, the transmit function remains enabled until the transmission of the frame finishes.

DM Duplex Mode Bit

This bit selects full-duplex operation.

PRM Promiscuous Mode Bit

This bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.).

26.3.49 ETNBnRFLR — Receive Frame Length Register

The ETNBnRFLR register specifies the maximum length (in bytes) of frames that can be received by this LSI.

This register must not be changed while reception is enabled (while the reception enable bit (ETNBnECMR.RE) in the E-MAC mode register is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0508_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.65 ETNBnRFLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RFL[17:0]	Receive Frame Length Setting value Check value 00000_H: 1,518 bytes : : $005EE_H$: 1,518 bytes $005EF_H$: 1,519 bytes $005F0_H$: 1,520 bytes : : $1FFFF_H$: 131,071 bytes 20000_H: 131,072 bytes : : $3FFFF_H$: 131,072 bytes

RFL[17:0] Receive Frame Length Bits

Frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data are not included in the transfer. When more data than the specified number of bytes are received, the portion of data that exceeds the specified value is discarded.

CAUTION

The prepared descriptor data size is just the specified value (ETNBnRFLR.RFL[17:0]).
Therefore descriptor data size must be more than ETNBnRFLR.RFL[17:0] + 8 if you will receive such the long frame.

26.3.50 ETNBnECSR — E-MAC Status Register

The ETNBnECSR register indicates the state of the E-MAC. The CPU can be notified of the state. For bits that generate an interrupt, the interrupt can be enabled or disabled by the corresponding bit in the E-MAC Interrupt Permission Register (ETNBnECSIPR) described in **Section 26.3.51, ETNBnECSIPR — E-MAC Interrupt Permission Register**.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0510$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LCHNG	—	ICD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	—	0	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 26.66 ETNBnECSR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LCHNG	Link signal change bit 0: Change of Link status signal (AVB_LINK) is not detected. 1: Change of Link status signal (AVB_LINK) is detected.
1	Reserved	When read, an undefined value is read. When writing, write the value after reset.
0	ICD	Illegal Carrier Detection 0: PHY-LSI has not detected an illegal carrier on the line. 1: PHY-LSI has detected an illegal carrier on the line.

LCHNG: Link Signal Change Bit

This bit indicates a transition of the link status signal (AVB_LINK) input from the PHY-LSI from high to low or low to high.

However, signal changes may also be detected at times when the link status signal (AVB_LINK) function is selected.

To check the current link state, refer to the link status pin state bit in the PHY status register (ETNBnPLSR.LINK).

Writing 1 to this bit clears it to 0.

ICD Illegal Carrier Detection Bit

This bit indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used.

Writing 1 to this bit clears it to 0.

26.3.51 ETNBnECSIPR — E-MAC Interrupt Permission Register

The ETNBnECSIPR register instructs permission of interrupt sources reported by the ETNBnECSR register. The bits can permit interrupts corresponding to the ETNBnECSR bits.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0518_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LINKIM	—	ICDIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 26.67 ETNBnECSIPR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LINKIM	LINK Interrupt Mask 0: Interrupts by the LINKI setting prohibited. 1: Interrupts by the LINKI setting permitted.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICDIP	False Carrier Detect Interrupt Enable 0: Interrupts by the ICD bit setting prohibited. 1: Interrupt by the ICD bit setting permitted.

LINKIM: Link Signal Change Interrupt Enable Bit

When this bit is set to “1” and the link signal change bit (ETNBnECSR.LCHNG) in the E-MAC status register is set to “1”, an interrupt occurs.

ICDIP Illegal Carrier Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the illegal carrier detection bit (ETNBnECSR.ICD) in the E-MAC status register to 1.

26.3.52 ETNBnPIR — PHY Interface Register

The ETNBnPIR register provides a means of access to the PHY-LSI internal registers via the MII.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0520$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	—	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.68 ETNBnPIR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MDI	MII Management Data-In Indicates the level of the ETNBnMDIO pin.
2	MDO	MII Management Data-Out Stores data to output from the ETNBnMDIO pin.
1	MMD	MII Management Mode 0: Read direction is specified. 1: Write direction is specified.
0	MDC	MII Management Data Clock The value set in this bit is output from the ETNBnMDC pin, which supplies the management data clock for the MII.

MDI MII Management Data-In Bit

This bit indicates the level of the ETNBnMDIO pin.

MDO MII Management Data-Out Bit

This bit stores data to output from the ETNBnMDIO pin.

The ETNBnMDIO pin outputs data when the MMD bit is set to 1 (to specify writing as the direction). Data are not output while the MMD bit is set to 0 (to specify reading as the direction).

MMD MII Management Mode Bit

This bit specifies the direction for data through MDIO (reading or writing).

MDC MII Management Data Clock Bit

Values set in this bit are output on the ETNBnMDC pin to supply the MII with the management data clock. For the method of access to the MII registers, see **Section 26.4.12, Connection to PHY-LSI**.

26.3.53 ETNBnPLSR — PHY LINK Status Register

The ETNBnPLSR register is used to check the status of PHY LINK pins.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0528$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LINK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.69 ETNBnPLSR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	LINK	When read, this bit returns the state of the ETNBnLINK pin.

26.3.54 ETNBnAPFTP — Auto PAUSE Frame Time Parameter Register

The ETNBnAPFTP register is used to set the value for the TIME parameter of automatically generated PAUSE frame.

When a PAUSE frame is automatically transmitted, the value set in this register is used as its TIME parameter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0554_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APFTP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.70 ETNBnAPFTP Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	APFTP[15:0]	Automatic PAUSE Frame Time Parameter Set a time parameter value of the Automatic PAUSE Frame.*1,*2 0000_H: Setting prohibited 0001_H: 1 × 512 bit-period 0002_H: 2 × 512 bit-period : : $FFFF_H$: 65535 × 512 bit-period

Note 1. The bit-period changes relative to the transfer speed.
100 Mbps: 1 bit-period = 10 ns
10 Mbps: 1 bit-period = 100 ns

Note 2. When setting the Transmission flow control operation mode bit (ETNBnECMR.TXF) in the E-MAC mode register to “1”, set this register value other than $0000\ 0000_H$.

APFTP[15:0] Auto Pause Frame Time Parameter

These bit configure the time parameter value for the transmit of Auto Pause Frame. The unit of the setting value is 512 bit-period.

26.3.55 ETNBnMPR — Manual PAUSE Frame Register

The ETNBnMPR register is used to set the value for the TIME parameter of manually generated PAUSE frames. When a PAUSE frame is manually transmitted, the value set in this register is used as its TIME parameter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0558_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.71 ETNBnMPR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MP[15:0]	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame.*1 0000_H: Setting prohibited 0001_H: 1 × 512 bit-period 0002_H: 2 × 512 bit-period : : $FFFF_H$: 65535 × 512 bit-period

Note 1. The bit-period changes relative to the transfer speed.
100 Mbps: 1 bit-period = 10 ns
10 Mbps: 1 bit-period = 100 ns

MP[15:0] Manual PAUSE Bits

These bits set the value of the TIME parameter in manually generated PAUSE frames. The unit for the setting is 512 bit periods.

26.3.56 ETNBnPFTCR — PAUSE Frame Transmit Counter

The ETNBnPFTCR register is a counter that indicates the number of times PAUSE frames have been transmitted.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 055C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFTXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.72 ETNBnPFTCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	PFTXC[15:0]	PAUSE Frame Transmit Counter These bits indicates the number of transmitted PAUSE frames.

PFTXC[15:0] PAUSE Frame Transmit Counter Bits

These bits indicate the total number of PAUSE frames that have been transmitted (both manually and automatically).

The bits are cleared to 0 when they are read.

If counting up and clearing of the counter coincide, clearing the counter takes priority.

26.3.57 ETNBnPFRCR — PAUSE Frame Receive Counter

The ETNBnPFRCR register is a counter that indicates the number of times PAUSE frames have been received.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0560_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFRXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.73 ETNBnPFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	PFRXC[15:0]	PAUSE Frame Receive Counter Counter for counting the number of received PAUSE frames

PFRXC[15:0] PAUSE Frame Receive Counter Bits

These bits indicate the number of PAUSE frames that have been received when flow control in reception is enabled (the RXF bit in ETNBnECMR = 1).

The bits are cleared to 0 when they are read.

If counting up and clearing the counter coincide, clearing the counter takes priority.

26.3.58 ETNBnMAHR — MAC Address High Register

The ETNBnMAHR register specifies the 32 higher-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

This register must not be changed while transmission or reception is enabled (E-MAC mode register reception enable bit (ETNBnECMR.RE) is set to “1” or transmission enable bit (ETNBnECMR.TE) is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 05C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.74 ETNBnMAHR Register Contents

Bit Position	Bit Name	Function
31 to 0	MA[47:16]	MAC Address Bits 47 to 16 These bits are used to set the 32 higher-order bits of the MAC address.

MA[47:16] MAC Address Bits 47 to 16

These bits are used to set the 32 higher-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 0123 4567$_H$ in the this register.

26.3.59 ETNBnMALR — MAC Address Low Register

The ETNBnMALR register specifies the 16 lower-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

This register must not be changed while transmission or reception is enabled (E-MAC mode register reception enable bit (ETNBnECMR.RE) is set to “1” or transmission enable bit (ETNBnECMR.TE) is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 05C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.75 ETNBnMALR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MA[15:0]	MAC Address Bits 15 to 0 These bits are used to set the 16 lower-order bits of the MAC address.

MA[15:0] MAC Address Bits 15 to 0

These bits are used to set the 16 lower-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 89AB$_H$ in the ETNBnMALR register.

26.3.60 ETNBnTROCR — Transmit Retry Over Counter Register

In this product, this register is not used because a half-duplex mode is not supported.

It shall be retained the value after reset.

The ETNBnTROCR register is a counter that indicates the number of frames the module was unable to transmit in 16 attempts at transmission including the first attempt and retries. When 16 attempts to transmit a frame fail, this register is incremented by 1. Counting up stops when the value in this register reaches 0000 FFFF_H.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0700_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TROC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.76 ETNBnTROCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	TROC[15:0]	Transmit Retry Over Counter These bits indicate the number of frames the module was unable to transmit in 16 attempts at transmission, including the first attempt and retries.

TROC[15:0] Transmit Retry Over Counter Bits

These bits indicate the number of frames the module was unable to transmit in 16 attempts at transmission, including the first attempt and retries.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.61 ETNBnCDCR — Late Collision Detect Counter Register

In this product, this register is not used because a half-duplex mode is not supported.

It shall be retained the value after reset.

The ETNBnCDCR register is a counter that indicates the number of all late collisions that occurred on the line after the start of data transmission. Counting up stops when the value in this register reaches 0000 FFFF_{H}.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0708_{H}
Value after reset:	0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COSDC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.77 ETNBnCDCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	COSDC[15:0]	Late Collision Detect Counter These bits indicate the number of all late collisions after the start of data transmission.

COSDC[15:0] Collision Detect Counter Bits

These bits indicate the number of all late collisions after the start of data transmission.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.62 ETNBnLCCR — Lost Carrier Counter Register

In this product, this register is not used because a half-duplex mode is not supported.

It shall be retained the value after reset.

The ETNBnLCCR register is a counter that indicates the number of times the carrier was lost during data transmission. Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0710_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.78 ETNBnLCCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	LCC[15:0]	Lost Carrier Counter These bits indicate the number of times the carrier was lost during data transmission.

LCC[15:0] Lost Carrier Counter Bits

These bits indicate the number of times the carrier was lost during data transmission.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.63 ETNBnCEFCR — CRC Error Frame Receive Counter Register

The ETNBnCEFCR register is a counter that indicates the number of times frames with CRC errors were received. Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0740_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CEFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.79 ETNBnCEFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CEFC[15:0]	CRC Error Frame Counter These bits indicate the number of CRC error frames received.

CEFC[15:0] CRC Error Frame Counter Bits

These bits indicate the number of received frames having CRC errors.

The bits are cleared to “0” when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to “1”.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.64 ETNBnFRECR — Frame Receive Error Counter Register

The ETNBnFRECR register is a counter that indicates the number of frame receive errors were generated by input on the ETNBnRXERR pin from the PHY-LSI. Counting up stops when the value in this register reaches 0000 FFFF$_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0748$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FREC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.80 ETNBnFRECR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	FREC[15:0]	Frame Receive Error Counter These bits indicate the number of errors during frame reception.

FREC[15:0] Frame Receive Error Counter Bits

These bits indicate the number of errors during frame reception.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.65 ETNBnTSFRCR — Too-Short Frame Receive Counter Register

The ETNBnTSFRCR register is a counter that indicates the number of received frames that were fewer than 64 bytes in length. Counting stops when the value in this register reaches 0000 FFFF_H.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0750_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSFRC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.81 ETNBnTSFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	TSFRC[15:0]	Too-Short Frame Receive Counter These bits indicate the number of frames received with a length of less than 64 bytes.

TSFRCR[15:0] Too-Short Frame Receive Counter Bits

These bits indicate the number of received frames that were fewer than 64 bytes in length.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.66 ETNBnTLFRCR — Too-Long Frame Receive Counter Register

The ETNBnTLFRCR register is a counter that indicates the number of received frames that were longer than the value specified in the receive frame length register (ETNBnRFLR). Counting up stops when the value in this register reaches 0000 FFFF$_{\text{H}}$.

Access:	This register can be read or written in 32-bit units.
Address:	<ETNBn_base> + 0758$_{\text{H}}$
Value after reset:	0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TLFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.82 ETNBnTLFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	TLFC[15:0]	Too-Long Frame Receive Counter These bits indicate the number of frames received with a length exceeding the value in ETNBnRFLR.

TLFRCR[15:0] Too-Long Frame Receive Counter Bits

These bits indicate the number of received frames that were longer than the value in ETNBnRFLR.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.67 ETNBnRFCR — Residual-Bit Frame Receive Counter Register

The ETNBnRFCR register is a counter that indicates the number of received frames containing “residual bits” (trailing bits not making up an 8-bit unit). Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0760_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.83 ETNBnRFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	RFC[15:0]	Residual-Bit Frame Receive Counter These bits indicate the number of received frames containing residual bits.

RFC[15:0] Residual-Bit Frame Receive Counter Bits

These bits indicate the number of received frames containing residual bits.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.68 ETNBnMAFCR — Multicast Address Frame Receive Counter Register

The ETNBnMAFCR register is a counter that indicates the number of received frames for which a multicast address was specified. Counting up stops when the value in this register reaches 0000 FFFF_H.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0778_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAFC [15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.84 ETNBnMAFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MAFC[15:0]	Multicast Address Frame Counter These bits indicate the number of multicast frames that have been received.

MAFC[15:0] Multicast Address Frame Counter Bits

These bits indicate the number of multicast frames that have been received.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

26.3.69 ETNBnIFCTL — Communication Interface Control Register

The ETNBnIFCTL register specifies the communication interface.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 1000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 26.85 ETNBnIFCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IFS	This bit selects the communication interface. 0: MII mode 1: Setting prohibited

26.4 Operation

The Ethernet AVB consists of the following functional units:

- DMA transfer controller (AVB-DMAC): Handles DMA transfer between the data storage areas for reception and transmission in the URAM and the reception and transmission FIFO buffers
- MAC controller (E-MAC): Handles transfer between the reception and transmission FIFO buffers and the MII

Using its direct memory access (DMA) function, the AVB-DMAC handles DMA transfer of frame data between the destinations for storing Ethernet frame data for transmission and reception in the URAM and the FIFO buffers for reception and transmission. Data cannot be directly read from or written to the FIFO buffers.

To handle DMA transfer, the AVB-DMAC requires information that includes the addresses for storage of data for transmission and received data. The information is referred to as descriptors. The AVB- DMAC reads data for transmission from the storage area for data to be transmitted and writes received data to the storage area for received data according to the information described in descriptors. The descriptors are placed in the URAM. Arranging multiple descriptors in descriptor lists allows the continuous reception or transmission of multiple Ethernet frames.

The E-MAC supports a MII, which provides an interface format for the externally connected PHY-LSI. The E-MAC constructs Ethernet frames from data written to the transmission FIFO and transmits these frames to the MII. It also performs CRC checking of Ethernet frames received from the MII and writes the frames to the reception FIFO.

26.4.1 AVB-DMAC Operating Modes

Figure 26.5, Operating Mode of AVB-DMAC illustrates the operating modes of the AVB-DMAC.

Transitions of AVB-DMAC operating mode are under the control of the items listed below.

- CPU operating mode (hardware reset)
- Configuration of the operating mode configuration bits (ETNBnCCC.OPC) in the AVB-DMAC mode register

The current operating mode can be determined by reading the operating mode status bits in the AVB- DMAC status register (ETNBnCSR.OPS).

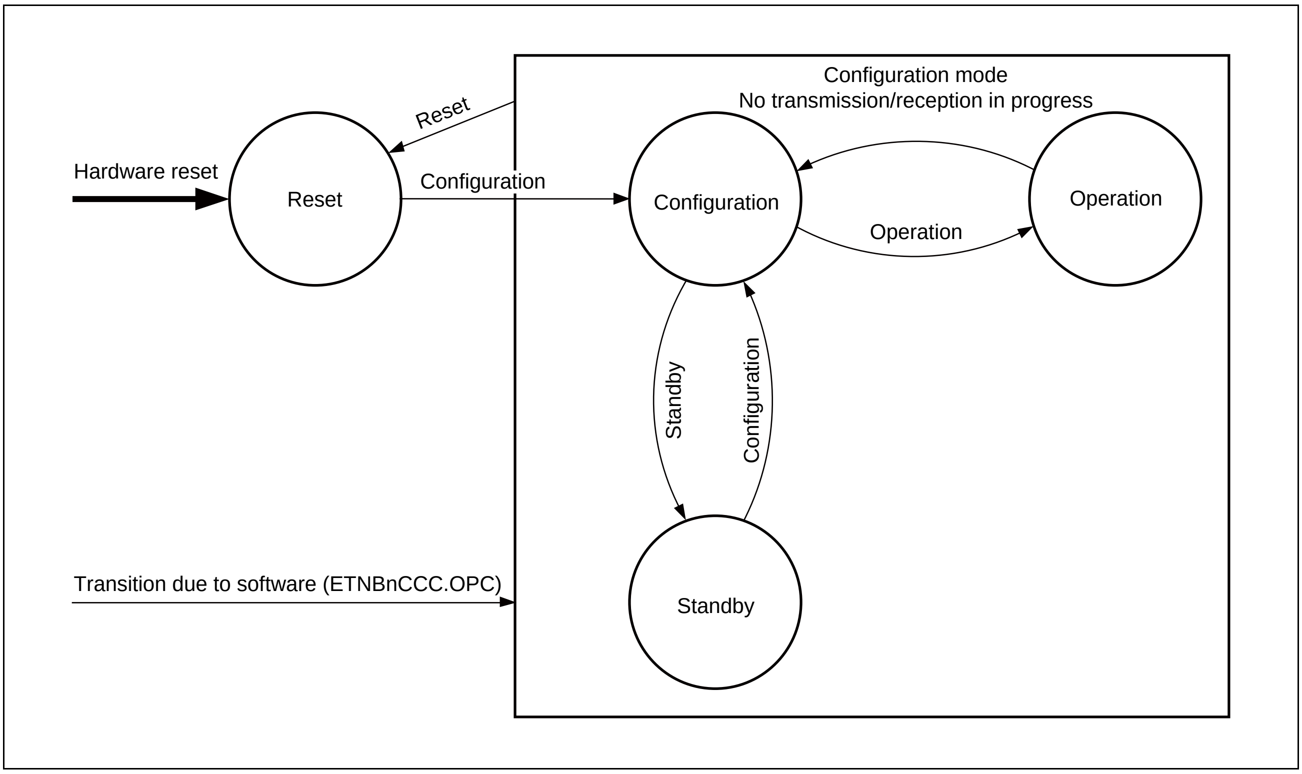


Figure 26.5 Operating Mode of AVB-DMAC

26.4.1.1 Operating Modes

(1) Reset Mode
After a hardware reset, the AVB-DMAC enters reset mode.
In reset mode, only the AVB-DMAC operating mode control function is controllable and other functions are all stopped. This mode is designed for reduced power when the Ethernet function is not necessary.

NOTE

When the operating mode transfers to reset mode all registers, except the ETNBnIFCTL register, are initialized to their reset values.

(2) Configuration Mode
In configuration mode, various settings for the AVB-DMAC can be made.
The operating functions are stopped and all status registers are initialized to their reset values. The E-MAC functions in this mode.

(3) Operation Mode
In operation mode, all functions of the AVB-DMAC can operate. Ethernet communications can only proceed in this mode.
In operation mode, do not set E-MAC again.

(4) Standby Mode
In standby mode, the E-MAC can only be used to control the operating mode. Other functions cannot be used.

26.4.1.2 How to Set the Operating Mode

Set the operating mode configuration bits in the AVB-DMAC mode register (ETNBnCCC.OPC) to select the operating mode. Furthermore, the current operating mode can be checked by reading the operating mode status bits in the AVB-DMAC status register (ETNBnCSR.OPS).

Transitions other than from operation mode to configuration mode are made after the value is written to the operating mode configuration bits (ETNBnCCC.OPC) (**Figure 26.6, Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode)**).

For transitions from operation mode to configuration mode, follow the procedure in **Figure 26.7, Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)** because any transmission and reception in progress will be executed before the transition to configuration mode.

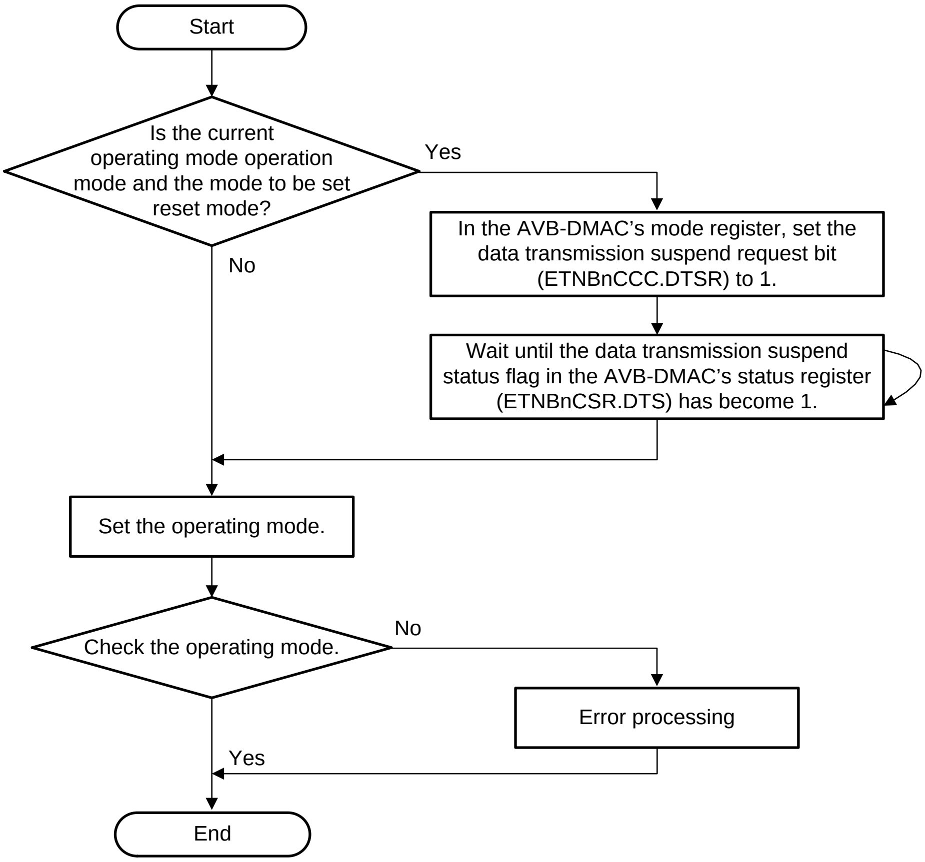


Figure 26.6 Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode)

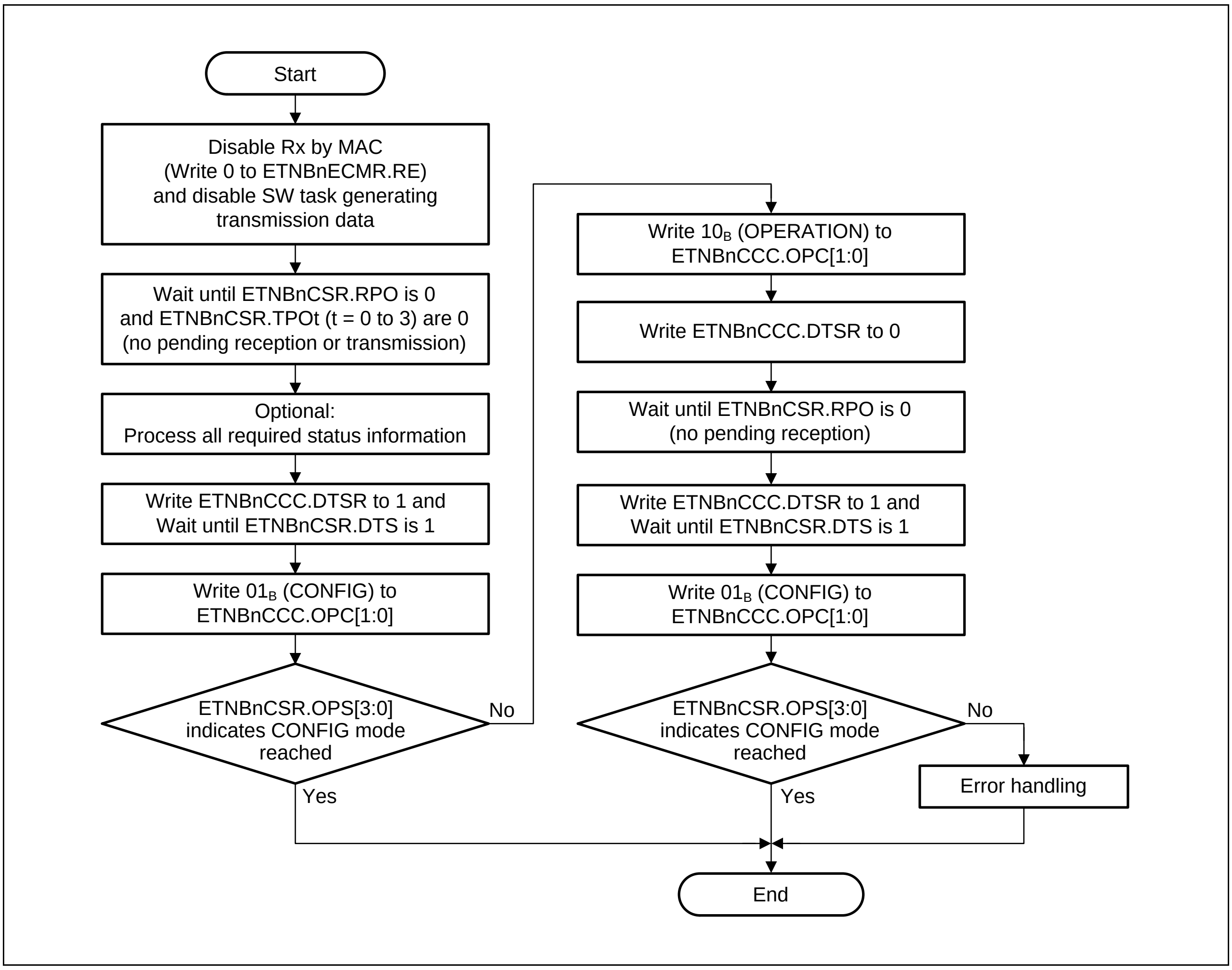


Figure 26.7 Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)

In the transition from operation mode to configuration mode, the AVB-DMAC executes the following operations before the transition is completed. Read the operating mode status bits in the AVB-DMAC status register (ETNBnCSR.OPS) to check that the transition to configuration mode has been completed.

- If the transfer of a frame between the reception FIFO and URAM is in progress, this is completed (other received frames remaining in the FIFO and any frames that are subsequently received by the E-MAC are discarded).
- If the transfer of a frame is in progress between the transmission FIFO and URAM, this is completed (frames for transmission remaining in the URAM will not be transmitted).
- All frames for transmission in the transmission FIFO are transferred to the E-MAC.

NOTES

When the operating mode shifts to configuration mode, all status registers are cleared. We recommend following the procedure below in the case of this transition.

1. Disable reception.
2. Since reception actually stopping after being disabled requires time, wait for an interval equivalent to that for reception of a maximum length packet.
3. Stop the software task that is generating data for transmission.
4. Wait until the receive process status bit (ETNBnCSR.RPO) and the transmit process status bits (ETNBnCSR.TPO0 to 3) in the AVB-DMAC status register are set to 0.

5. Capture all of the required status information.
6. Set the operating mode configuration bits in the AVB-DMAC mode register (ETNBnCCC.OPC) to initiate the transition to configuration mode.

26.4.1.3 Operating Mode Transitions Due to Hardware

The following hardware factors can also initiate transitions of the AVB-DMAC operating mode.

(1) Hardware Reset
Resetting of the LSI chip leads to resetting of the entire EthernetAVB module. The operating mode shifts to reset mode.

26.4.2 Common Control for Transmission and Reception

26.4.2.1 Initialization Procedure

Figure 26.8, Outline of the Initialization Procedure shows the overall initialization procedure in outline.

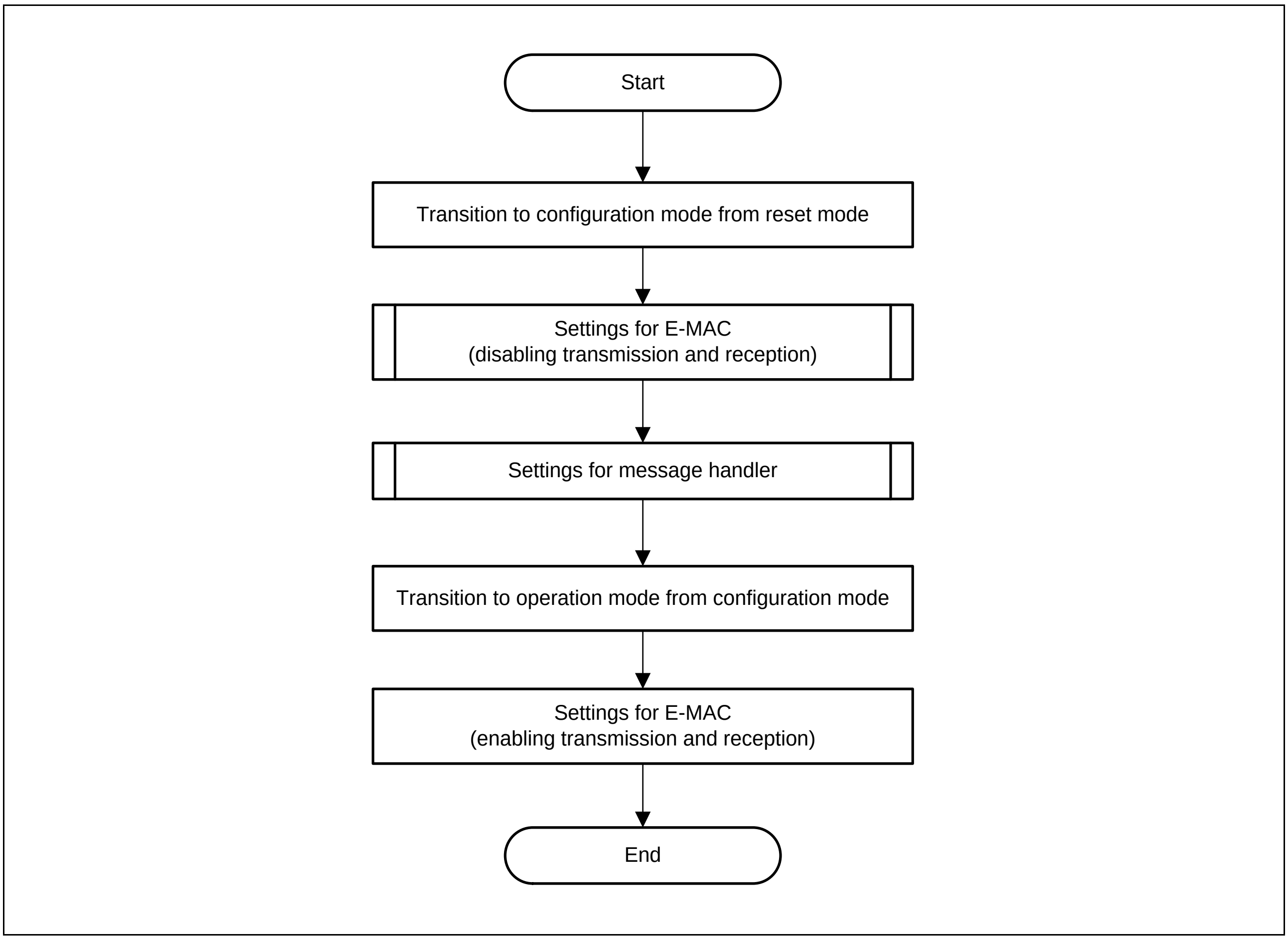


Figure 26.8 Outline of the Initialization Procedure

(1) Initializing the Receiver Section

Before starting reception, follow the procedure below.

Set the operating mode to operation mode or standby mode, and do not enable reception until the settings for the AVB-DMAC are completed.

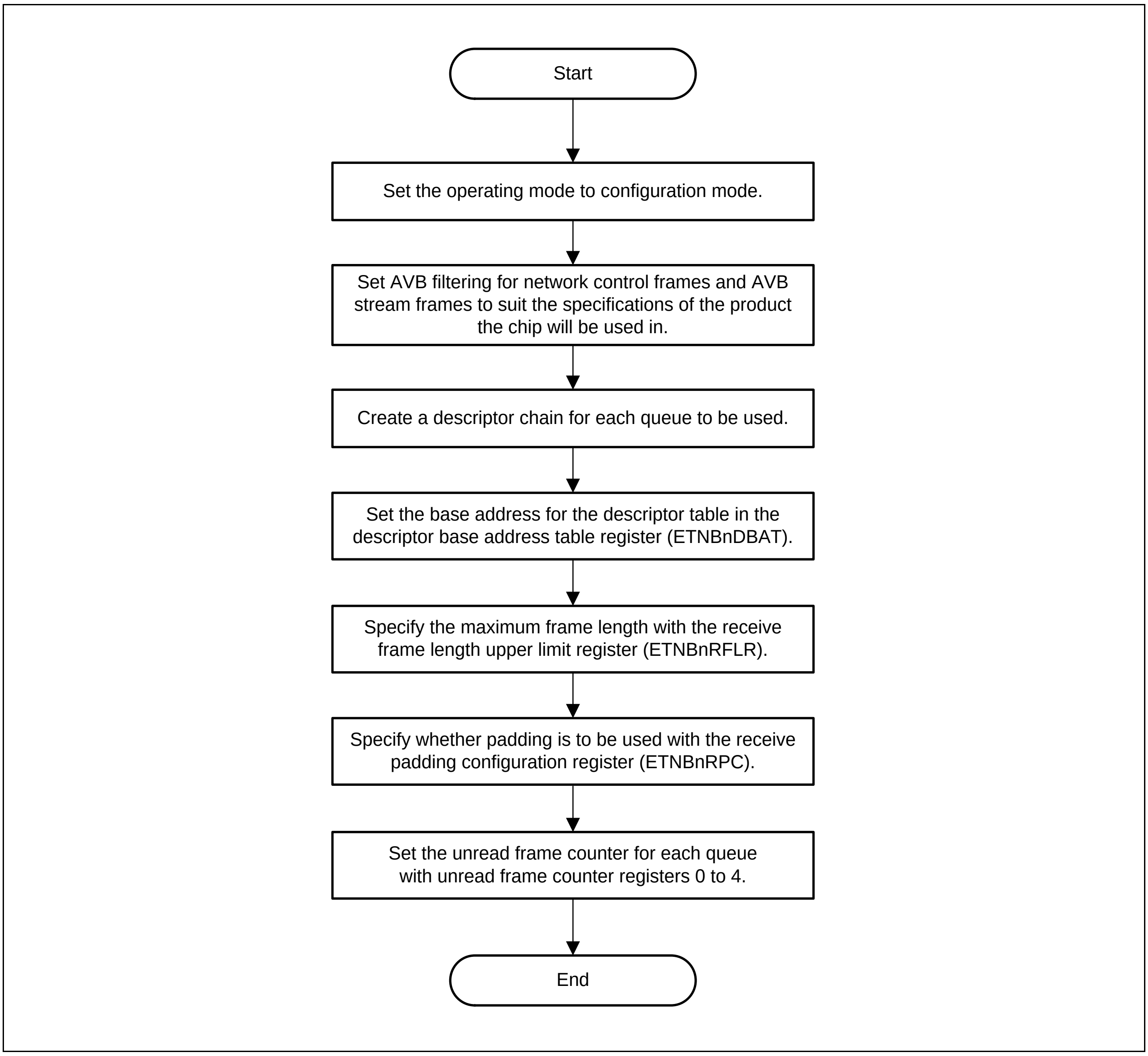


Figure 26.9 Procedure for Initializing the Receiver Section

(2) Initializing the Transmitter Section

Figure 26.10, Procedure for Initializing the Transmitter Section illustrates initialization of the transmitter section.

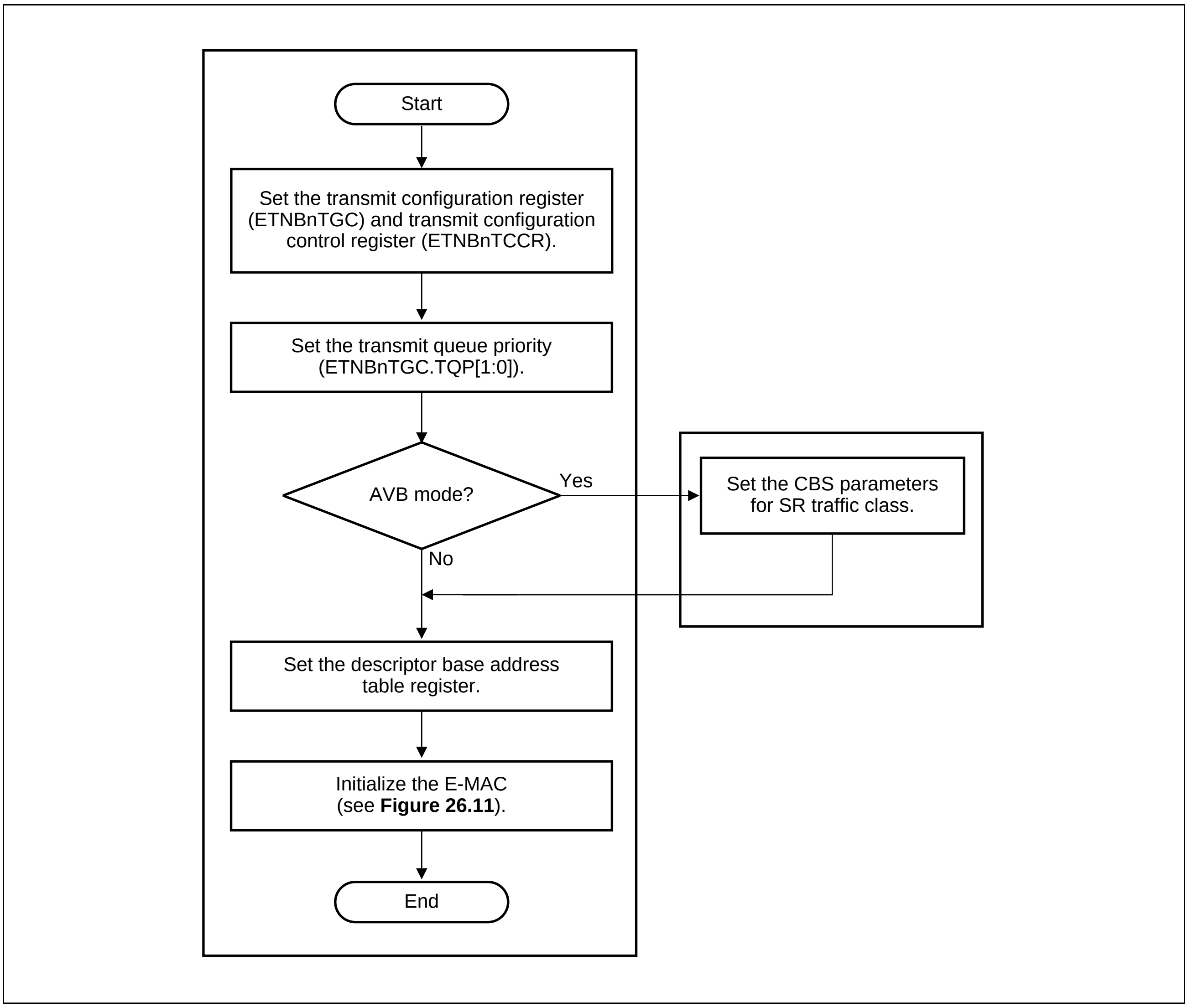


Figure 26.10 Procedure for Initializing the Transmitter Section

(3) Configuration the E-MAC Block

Figure 26.11, Procedure for Configuration the E-MAC Block illustrates configuration of the E-MAC block.

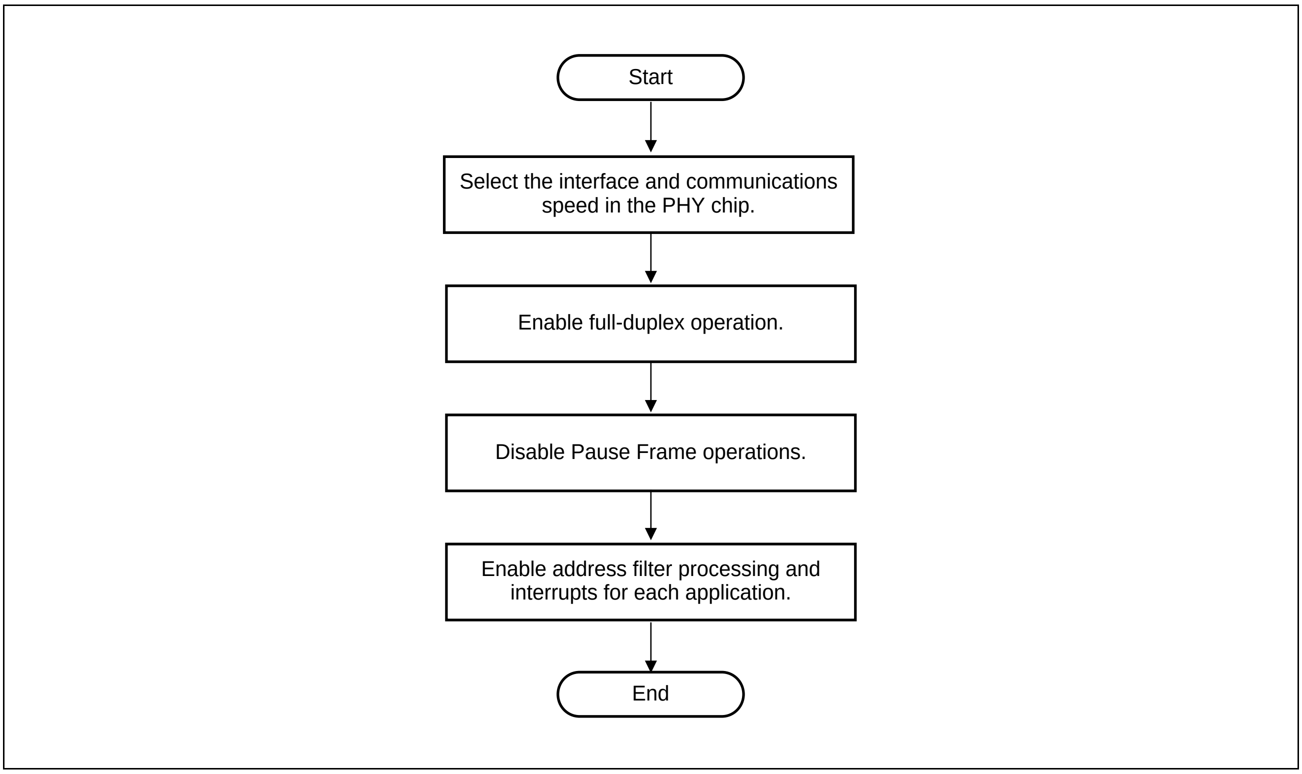


Figure 26.11 Procedure for Configuration the E-MAC Block

(4) Configuration of the Message Handler Section

Figure 26.12, Configuration of the Message Handler Section illustrates configuration the message handler section.

For a description of how to set up the descriptors and the CBS traffic shaping parameters, see **Section 26.4.3, Descriptors**, and **Section 26.4.6, CBS (Credit-Based Shaping)**.

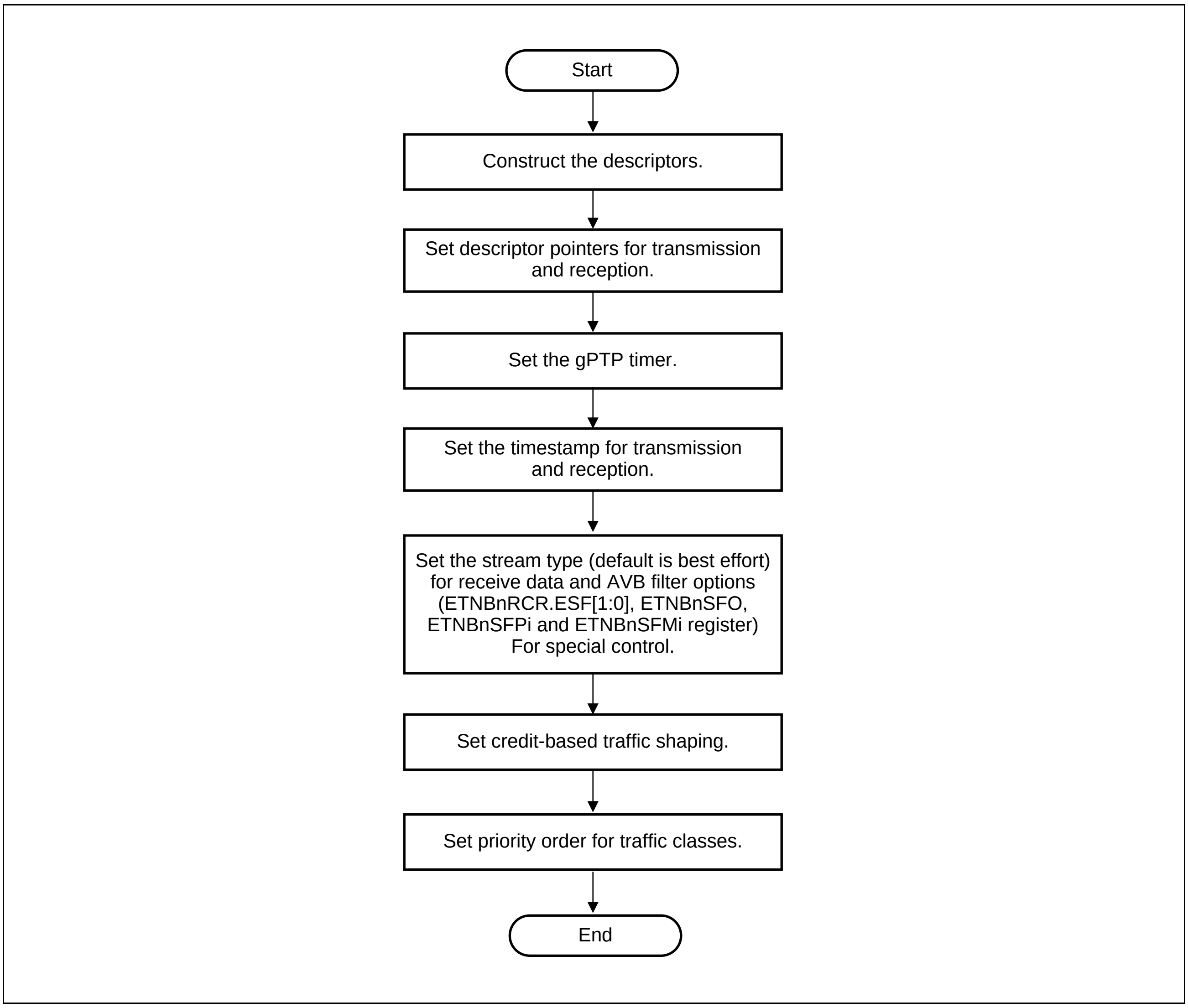


Figure 26.12 Configuration of the Message Handler Section

26.4.2.2 Scheduling Reception and Transmission

The AVB-DMAC normally has independent buses for transmission and reception. Furthermore, the four processes of fetching, storing, transmission and reception are basically independent of one another. Fetching and storing, however, share the same bus master so cannot be executed simultaneously. Access to the bus master is controlled by the scheduler.

Figure 26.13, Schematic View of AVB-DMAC Operations in Transmission and Reception is a schematic view of AVB-DMAC operations in transmission and reception.

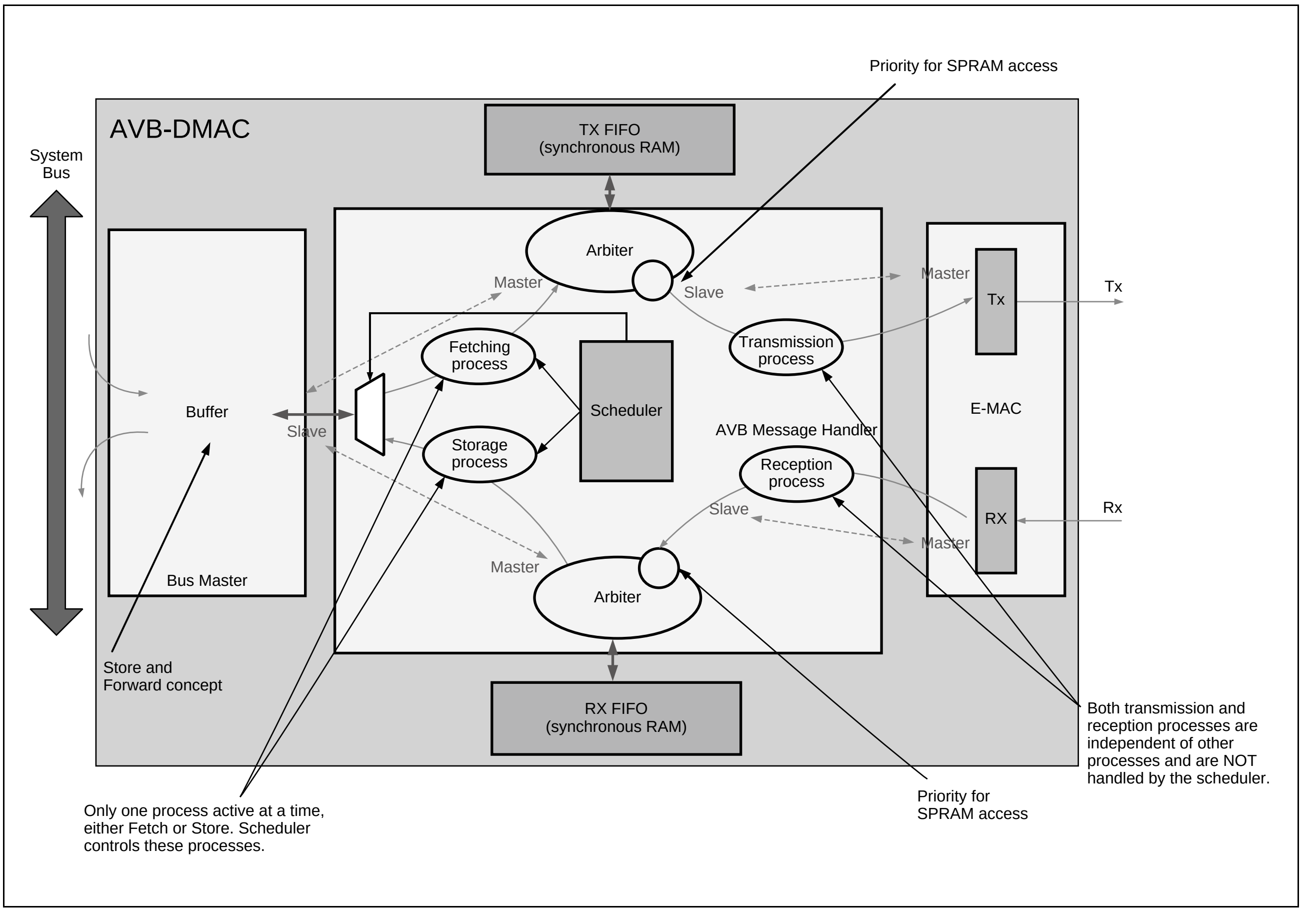


Figure 26.13 Schematic View of AVB-DMAC Operations in Transmission and Reception

Storing and fetching are alternately performed. When the number of frames held by the reception FIFO reaches the warning level, storing takes precedence over fetching.

(1) Relationship between Transmission Queue Numbers and Traffic Classes

In fetching, the relationships between the transmission queues and traffic classes are fixed, so the priority specified by the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) has no effect.

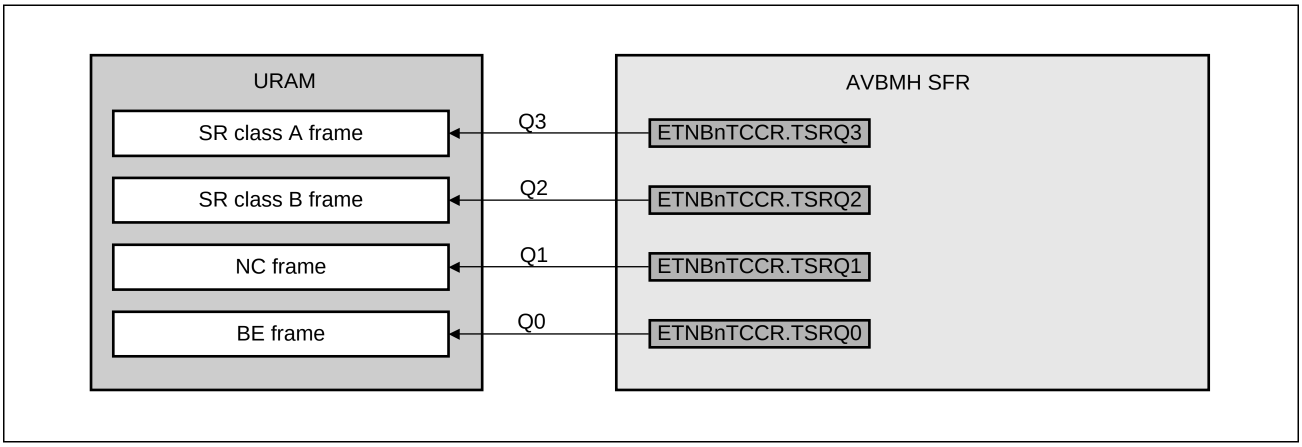


Figure 26.14 Class Associations of Queues for the Scheduler

In fetching, the credit values for stream classes A and B are not taken into account. Behavior depends on the setting of the transfer FIFO size configuration bits in the transfer configuration registers (ETNBnTGC.TBDt) and on the frame size that can be fetched to the transmission FIFO.

When the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) are 00_B or 01_B, the priority order is Q3 → Q2 → Q1 → Q0.

When the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) are 11_B, the priority order is Q1 → Q3 → Q2 → Q0.

26.4.2.3 Checking Integrity

The AVB-DMAC is capable of detecting and identifying errors produced in the processing of Ethernet frames and in the transfer of frame data for transmission and reception.

(1) Concept of Integrity Checking in Reception

The purpose of receive integrity check is to prevent error frames from being stored in the URAM.

If an error frame is stored, information to identify the frame as an error frame is appended to the data from the frame in the URAM.

CAUTION

If a special descriptor chain is to be used for header/data separation on reception, an error that breaks the sequence may lead to storage space for synchronization running out. In such cases, software interaction or re-synchronization via the EOS descriptor is required.

(2) Concept of Integrity Checking in Transmission

The purpose of integrity checking in transmission is to prevent the transmission of broken frames.

Since transmission of a frame by the E-MAC can neither be stopped nor disabled once it has started, this check involves intensive monitoring for problems that can arise during fetching.

(3) Items for Monitoring in Both Reception and Transmission

(a) Errors in Access to the URAM for Reading of Descriptors

The same descriptor may be processed again because the current descriptor address (ETNBnCDARq.CDA) is not changed.

If this problem occurs in a divided frame, the sequence may be broken.

- In reception
  - The received frame will be lost.
  - The same problem will occur for the next frame of data received for the same queue.
- In transmission
  - The transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.
  - The frame will be lost from the transmission FIFO.

Errors in access to read descriptors from the URAM are detected from the response signal of the AXI- Bus.

(b) Illegal Configuration of a Descriptor by an Application

The same descriptor may be processed again because the current descriptor address (ETNBnCDARq.CDA) is not changed.

If this problem occurs in a divided frame, the sequence may be broken.

- In reception
  - The received frame will be lost.
  - The same problem will occur for the next frame of data received for the same queue.
- In transmission
  - The transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.
  - The frame will be lost from the transmission FIFO.

(c) Errors in Access to the URAM for Writing of Descriptors

As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated.

As DESCR.DT was not updated, hardware and software synchronization may have been disengaged.

Errors in access to write descriptors to the URAM are detected from the response signal of the AXI- Bus.

(4) Items for Monitoring in Reception

(a) Errors in Access to the URAM for Writing of Data or Timestamps

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) is updated.
- DESCR.EI is set to indicate incorrect contents.
- When an access error occurs in a divided frame, the descriptor sequence may be damaged and queues may be unusable.

Errors in access to write data or descriptors to the URAM are detected from the response signal of the AXI-Bus.

(b) Errors in Access for Reading from the Reception FIFO

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) is updated.
- DESCR.EI is set to indicate incorrect contents.
- When an access error occurs in a divided frame, the descriptor sequence may be damaged and queues may be unusable.
- Errors of this type are detected by the ECC checker for the reception FIFO.

(c) Damaged Data in the Reception FIFO

- Received frames are all invalidated.
- All frames stored in reception FIFO are discarded. At this time, the number of frames and queue information cannot be captured.

If damaged data in the reception FIFO is an error due to the reception FIFO, this is detected by the AVB-DMAC.

(5) Items for Monitoring in Transmission

(a) Errors in Access for Reading Data from the URAM

- Data that have already been fetched are discarded from the transmission FIFO.
- When an error of this type occurs during processing of an FSINGLE or FEND descriptor:
  As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.
- When an error of this type occurs during processing of an FSTART or FMID descriptor:
  - The current descriptor address (ETNBnCDARq.CDA) is not updated.
  - The transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.

Errors in access to read data from the URAM are detected from the response signal of the AXI-Bus.

(b) Overflow of the Transmission FIFO

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.
- The frame will be discarded from the FIFO.

(c) Errors in Access for Reading of the Transmission FIFO

The AVB-DMAC is incapable of detecting an error in reading of the transmission FIFO, resulting in the transmission of a broken frame.

Errors of this type are detected by the ECC checker for the transmission FIFO.

(d) Frame size error during transmission

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.

A transmit frame size error is detected when the size setting in one or more (in the case of a divided frame) descriptors for frame transmission is 1997 or more bytes. Such frames are cut out and transmitted.

(e) Damaged Data in the Transmission FIFO

- Fetching is not affected by damaged data.
- Since damaged data from the FIFO is only detected during frame transmission, an unexpected frame data may be transmitted.

If damaged data in the transmission FIFO is an error due to the transmission FIFO, this is detected by the AVB-DMAC.

26.4.3 Descriptors

26.4.3.1 Data Representation in URAM

The AVB-DMAC transfers data for transmission and received data to and from the application software via the URAM.

The memory in the URAM for use by the AVB-DMAC is configured with control structures referred to as descriptors and associated areas to which the frame data are allocated. Dividing the memory into a control area and data area allows the flexible allocation of frame data to the URAM. This enables sharing of the areas to which frame data are allocated and the use of non-contiguous areas. Frame data can be copied without using the CPU. Arbitration that ensures hardware and software access to the memory area is also available without using the AVB-DMAC.

Figure 26.15, Example of URAM Memory Map shows an example of the memory maps for descriptors and the descriptor data area in the URAM.

A descriptor consists of its type (DESCR.DT), which controls the descriptor functions, a descriptor pointer (DESCR.DPTR) indicating the start address for storage of the frame data in the descriptor area, and the data size field (DESCR.DS), indicating the amount of frame data. Respective descriptors can generate interrupts after completion of processing. Enabling and disabling of the interrupt is controlled by the descriptor interrupt enable bits (DESCR.DIE).

The descriptor may also hold information related to content. This information does not affect general descriptor functions. It provides information other than the frame data proper, such as on the state of reception.

For details, see **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.

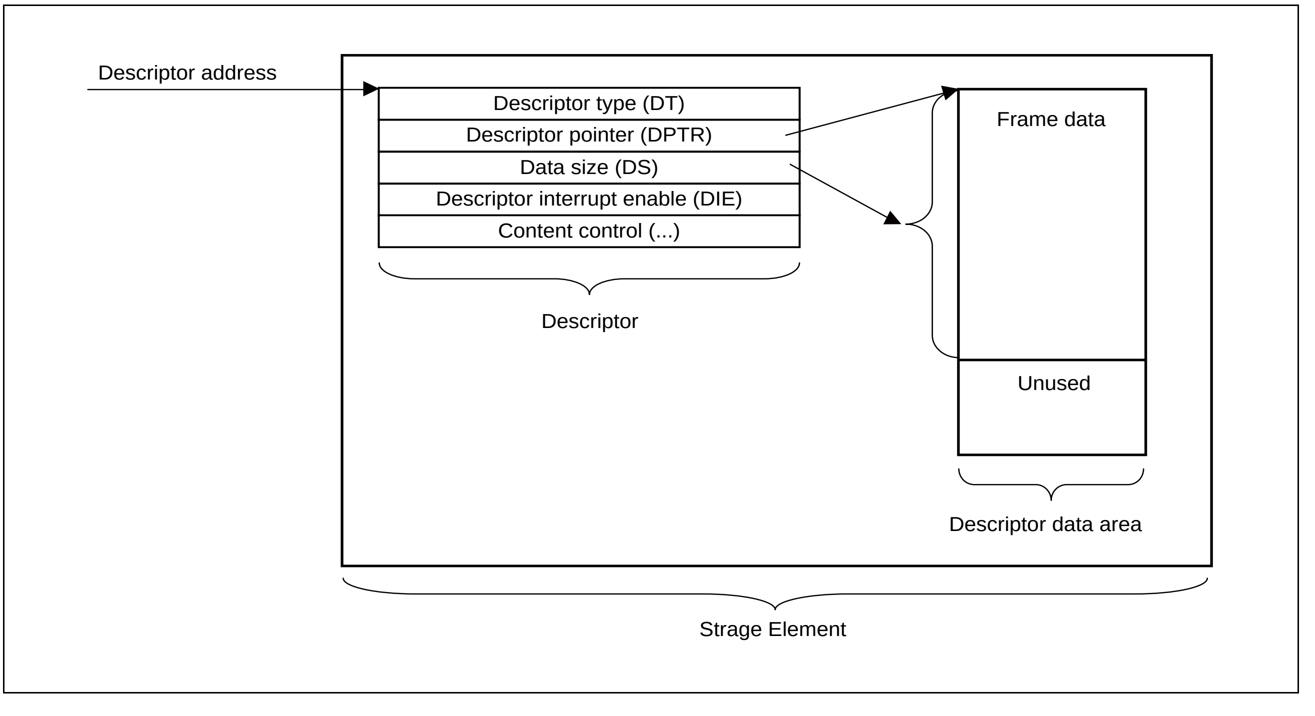


Figure 26.15 Example of URAM Memory Map

The descriptor must be aligned with a 32-bit boundary in the URAM.

Descriptors are generally configured of 64 bits, but are configured of 160 bits when reception and storage of gPTP timestamps is enabled.

The frame data must also be aligned with a 32-bit boundary in the URAM.

The amount of data in the frame is defined by the data size bits (DESCR.DS). In reception, these bits indicate the upper limit on the size of frames to be received. If the data size is not aligned with a 32-bit boundary, the residual bytes in the data area are set as unused bytes.

CAUTION

AVB-DMAC will store the data area pointed by descriptor for specified data size, even if there is no enough space.

26.4.3.2 Using Descriptor Chains in Queues

Transmission and reception descriptors in the URAM are grouped into queues. Each queue handles frames so that they are transmitted in order of priority and received separately. A queue is capable of controlling one or more frames. Accordingly, multiple descriptors can be assigned to one queue. A combination of multiple descriptors is referred to as a descriptor chain.

For a descriptor chain, the three general descriptor types listed below are defined. For details on these descriptor types, see **Section 26.4.3.6, Descriptor Type**.

- Descriptors that define frame data
- Descriptors that control the descriptor chain itself (e.g. LINK, EOS).
- Descriptors that arbitrate access by hardware or software

Figure 26.16, Outline of the Basic Descriptor Chains shows the two basic topologies for descriptor chains. In the simplified examples in the figure, all descriptors allocated to the chain are stored in the array.

- For a linear descriptor chain, the last descriptor in the array is a control descriptor indicating the end of the descriptors (e.g. EEMPTY).
- For a cyclic descriptor chain, the last descriptor in the array is a control descriptor that returns to the first descriptor in the array (e.g. LINK).

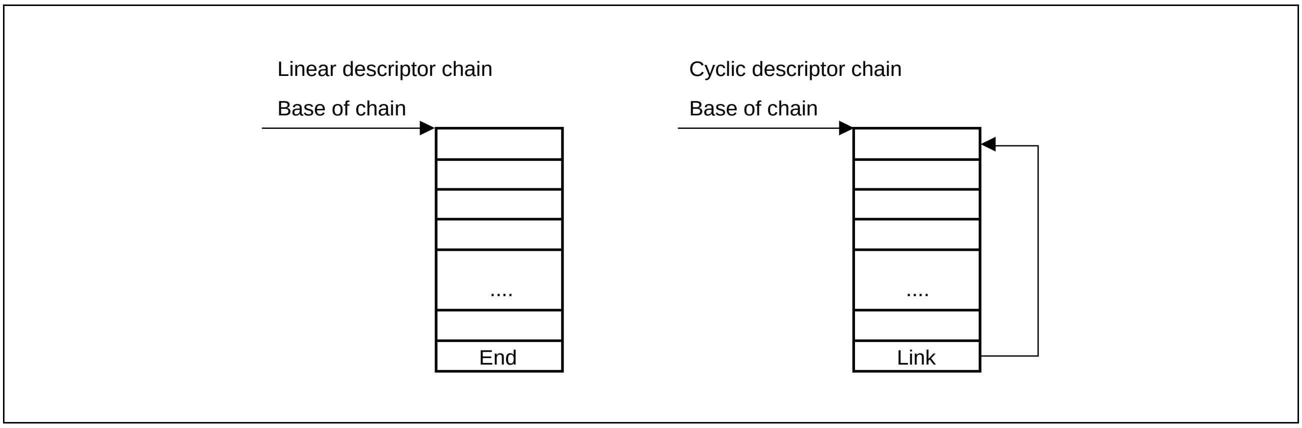


Figure 26.16 Outline of the Basic Descriptor Chains

The relationship between queues and descriptor chains is defined by the base addresses of chains. A queue is connected to one descriptor chain over one round of processing. There is also a method of switching to a different chain while in operation mode.

There are no restrictions on the number of link descriptors and their locations within the chain. The last descriptor of a designed chain determines the topology.

Which chain structure is to be used or which topology is suitable depends on the application. A description of how to design descriptor chains to suit various applications is given in **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.

26.4.3.3 Descriptor Base Address Table

Concerning the base address table to be written in the URAM, set the address of the top descriptor of respective queues.

Entries 0 to 3 are used to access transmission queues 0 to 3. Subsequent entries are used to access reception queues. Entry 4 thus corresponds to reception queue 0.

The configuration of entries in the base address table is the same as the configuration of link descriptors. We recommend using the descriptor type (DESCR.DT) LINKFIX. The link descriptor does not need to be updated because it does not change even after descriptor processing, so it does not require updating. The first descriptor of a chain performs hardware and software synchronization. If the application requires hardware and software synchronization for the base addresses, use the descriptor type (DESCR.DT) LINK.

The CPU is only capable of using LINKFIX and LINK as descriptor types (DESCR.DT) of descriptors in the base address table.

Set the location of the base address table in the URAM in the descriptor base address table register (ETNBnDBAT).

Figure 26.17, Example of a Base Address Table for Reception and Transmission Queues shows an example of a base address table for controlling four transmission and three reception queues. The boxes to the right of the table represent descriptor chains of a topology.

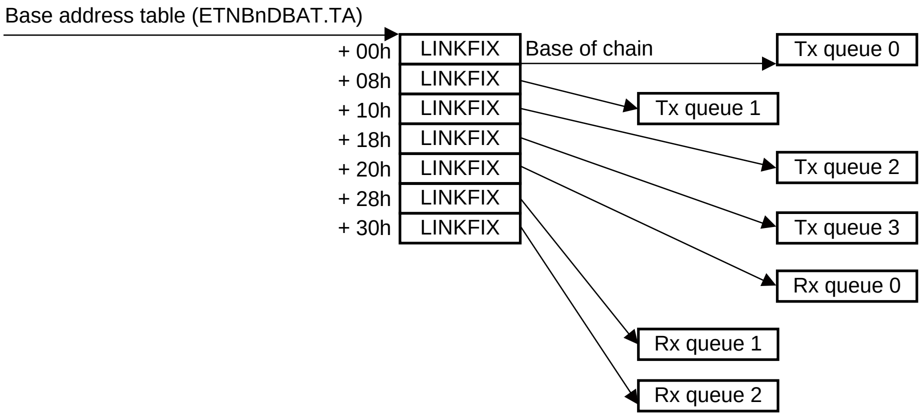


Figure 26.17 Example of a Base Address Table for Reception and Transmission Queues

CAUTION

The size of the descriptors in the base address table is always eight bytes even if the queue itself includes extended descriptors.

26.4.3.4 Descriptor Chain Processing

When a descriptor is currently being processed or the queue for a descriptor is active, the current descriptor is a descriptor to be processed. The current descriptor address for use by a queue q can be checked in the current descriptor address register q (ETNBnCDARq).

The current descriptor is stored in a register or descriptor as described below.

- In the descriptor base address table registers for all q queues (ETNBnDBAT) (ETNBnDBAT.TA + 8 × q) when the operating mode shifts to operation mode.
- In the descriptor base address table register (ETNBnDBAT) (ETNBnDBAT.TA + 8 × q) when a base address load request is issued for a queue q by setting the corresponding bit (ETNBnDLR.LBAq) in the descriptor base address load request register.
- It is set to DESCR.DPTR when a link descriptor (LINK, LINKFIX) is processed.

After a descriptor has been processed, the current descriptor for the same queue is incremented by the size of the descriptors being handled by the queue (8 bytes for normal descriptors and 20 bytes for extended descriptors). The AVB-DMAC updates the descriptor type and informs the CPU that the descriptor has been processed.

26.4.3.5 Descriptor Interrupts

A descriptor is able to issue a descriptor interrupt on completion of its processing. The setting of the descriptor interrupt enable bits (DESCR.DIE) in each descriptor selects disabling or generation of the descriptor interrupt.

The descriptor interrupt is a common resource that is shared between reception and transmission queues. Software control of the descriptor interrupt provides a flexible method of application-specific flag processing.

Figure 26.18, Method of Descriptor Interrupt Generation illustrates the way in which the AVB-DMAC generates descriptor interrupts (or sets bits in the descriptor interrupt status register (ETNBnDIS.DPFi)). Processing of a descriptor with the value i in the descriptor interrupt enable bits (DESCR.DIE) leads to the corresponding bit in the descriptor interrupt status register (ETNBnDIS.DPFi) being set.

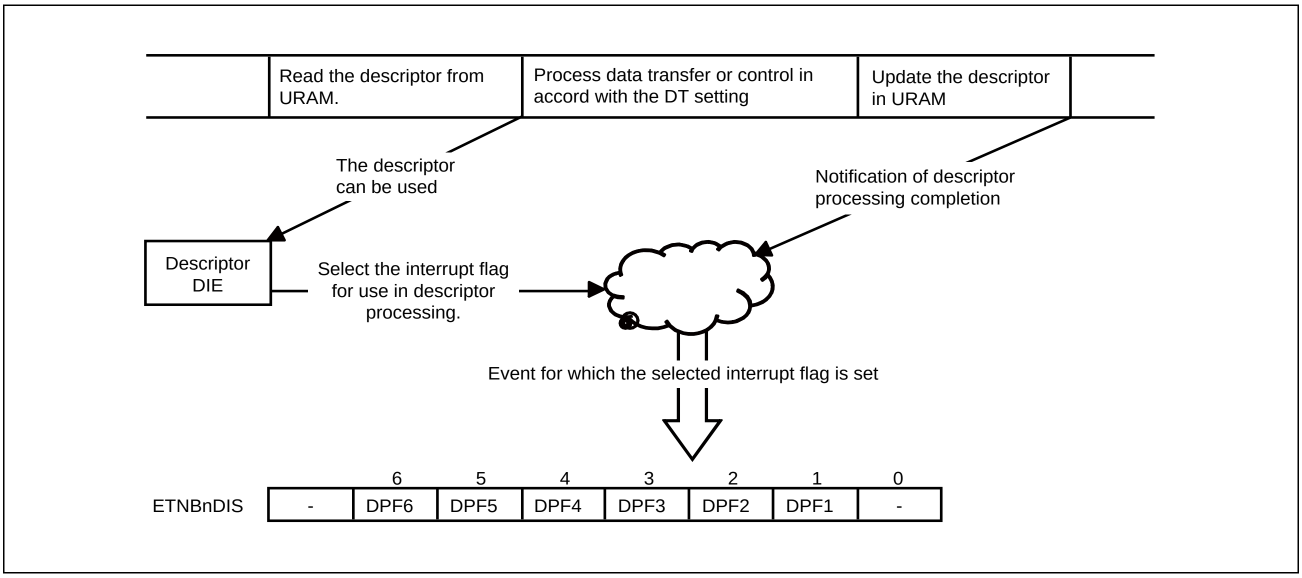


Figure 26.18 Method of Descriptor Interrupt Generation

26.4.3.6 Descriptor Type

The descriptor types (indicated by the DESCR.DT bits) supported by the AVB-DMAC fall into the following three categories.

- Definitions of frame data
- Control of descriptor chains
- Hardware and software arbitration

Table 26.86, Summary of Descriptor Types is a summary of the descriptor types available for the AVB-DMAC. Entries under “Name” are the names of the descriptor types and the values under “DT” are the corresponding values to be set in the descriptor type field (DESCR.DT). A given descriptor may be handled differently according to whether it is in a transmission or reception queue, so the transmission and reception columns list the scopes of control and processing of the descriptor types.

The abbreviations defined below are used in the transmission and reception columns.

Definition of SW:

- The descriptor is processed by software.
- Software has access to and may modify the descriptor and descriptor data area.
- The descriptor cannot be changed by hardware (AVB-DMAC).

Definition of HW:

- The descriptor is processed by hardware (AVB-DMAC).
- Software must modify neither the descriptor nor the descriptor data area.
- Hardware (AVB-DMAC) processes the descriptor and subsequently changes the descriptor type.

Invalid:

This descriptor type is not used in transfer in the given direction (transmission or reception). Do not write the value to the descriptor type (DESCR.DT) field.

Hardware does not process the descriptor type. The current descriptor address (ETNBnCDARq.CDA) is not changed even when the queue is of a descriptor type of this setting.

Table 26.86 Summary of Descriptor Types

Name	DT	Description	Reception	Transmission
Frame data				
FSTART	5	Frame Start The descriptor points to valid data for a frame. The frame starts with the given data and continues with that indicated by the next descriptor.	SW	HW
FMID	4	Frame Middle The descriptor points to valid data for a frame. The frame started with a previous descriptor and continues to the data indicated by the next descriptor.	SW	HW
FEND	6	Frame End The descriptor points to valid data for a frame. The frame continues from the previous descriptor and ends with the data indicated by in this descriptor.	SW	HW
FSINGLE	7	Frame Single The descriptor points to valid data for a complete frame.	SW	HW
Chain control				
LINK	8	Link Defines the next descriptor in the chain.	HW	HW
LINKFIX	9	Fixed Link Defines the next descriptor in the chain, but not changed by AVB-DMAC after processing.	SW	SW
EOS	10	End Of Set Control element to split descriptor chain. Chain stops and waits for user interaction.	HW	HW
HW/SW arbitration				
FEMPTY	12	Frame Empty A descriptor related to frame data but not containing valid data for a frame	HW	SW
FEMPTY_IS	13	Frame Empty Incremental Start A descriptor related to frame data but not containing valid data for a frame DESCR.DPTR sets the base address of an “incremental data area” in the URAM.	HW	Invalid
FEMPTY_IC	14	Frame Empty Incremental Continue A descriptor related to frame data but not containing valid data for a frame Data indicated by the pointer are for storage in an incremental data area in the URAM.	HW	Invalid
FEMPTY_ND	15	Frame Empty No Data Storage A descriptor related to frame data but not containing valid data for a frame The descriptor is processed in the same way as FEMPTY but data are not stored in the URAM.	HW	Invalid
LEMPTY	2	Link Empty A link descriptor for processing by the AVB-DMAC	SW	SW
EEMPTY	3	EOS Empty An EOS descriptor for processing by the AVB-DMAC	SW	SW
DT0	0	Reserved	Invalid	Invalid
DT1	1	Reserved	Invalid	Invalid
DT11	11	Reserved	Invalid	Invalid

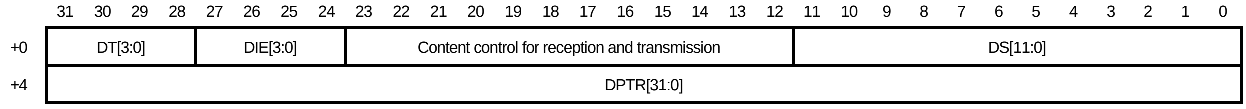
(1) Layout of General Descriptors in the URAM

The AVB-DMAC updates processed descriptors in the URAM. The field to be changed in a descriptor being updated depends upon whether the direction is transmission or reception and the queue mode. Other fields will not be changed. There are no restrictions on the values set in unused descriptor fields (indicated by "—" in the figure).

(2) Frame Data Descriptors

The allocation of bits in the frame data descriptors (FSTART, FMID, FEND, and FSINGLE) is shown below.

- Normal descriptor (usable in both reception and transmission)



- Extended descriptor (usable only in reception)

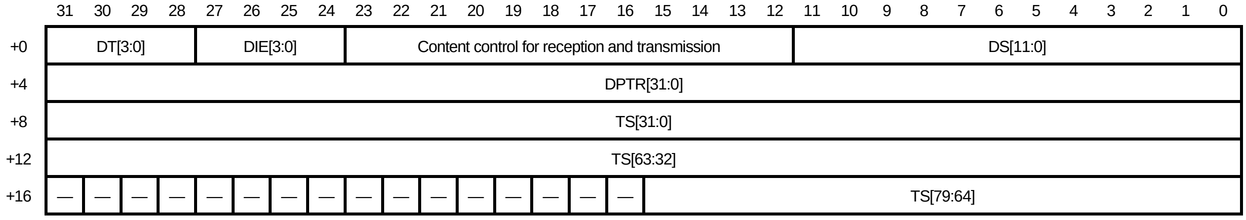


Table 26.87 Contents of Frame Data Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 5: FSTART 4: FMID 6: FEND 7: FSINGLE For details, see **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.
TS[79:0]	Timestamp Timestamp of the received frame (only available in extended descriptors)

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

The upper reserved bits of TS[79:0] are set to 0000_H after the time stamp is stored.

(3) Hardware/Software Arbitration Descriptors (Only for Reception)

The allocation of bits in the descriptors for hardware/software arbitration (FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND) is shown below.

The allocation of bits in the arbitration descriptors for use in reception is the same as in frame data descriptors.

- Normal descriptor

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>+0</td><td colspan="4">DT[3:0]</td><td colspan="4">DIE[3:0]</td><td colspan="12">Content control for reception and transmission</td><td colspan="12">DS[11:0]</td></tr>
<tr><td>+4</td><td colspan="32">DPTR[31:0]</td></tr>
</table>

- Extended descriptor (usable only in reception)

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>+0</td><td colspan="4">DT[3:0]</td><td colspan="4">DIE[3:0]</td><td colspan="12">Content control for reception and transmission</td><td colspan="12">DS[11:0]</td></tr>
<tr><td>+4</td><td colspan="32">DPTR[31:0]</td></tr>
<tr><td>+6</td><td colspan="32">TS[31:0]</td></tr>
<tr><td>+12</td><td colspan="32">TS[63:32]</td></tr>
<tr><td>+16</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td colspan="16">TS[79:64]</td></tr>
</table>

Table 26.88 Contents of Hardware/Software Arbitration Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 12: FEMPTY 13: FEMPTY_IS 14: FEMPTY_IC 15: FEMPTY_ND For details, see **Table 26.86, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.
TS[79:0]	Timestamp Timestamp of the received frame (only available in extended descriptors)

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

When the descriptor is an extended descriptor, it has a 12-byte unused area.

In an FEMPTY descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), data size (DS), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IS descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IC descriptor, the descriptor type (DT) and descriptor interrupt enable (DIE) are used.

In an FEMPTY_ND descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and data size (DS) are used.

(4) Link Descriptors

The allocation of bits in the link descriptors (LINK and LINKFIX) is shown below.

- Normal descriptor

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+0	DT[3:0]				DIE[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
+4	DPTR[31:0]																															

Table 26.89 Contents of Link Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 8: LINK 9: LINKFIX For details, see **Table 26.86, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 26.4.4.2, Setting Up Reception Descriptors**, and **Section 26.4.5.2, Setting Up Transmission Descriptors**.
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

When the descriptor is an extended descriptor, it has a 12-byte unused area.

(5) Other Descriptors

The allocation of bits in the other descriptors (EOS, FEMPTY (only for transmission), LEMPTY, and EEMPTY) is shown below.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+0	DT[3:0]				DIE[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
+4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 26.90 Contents of Other Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 10: EOS 12: FEMPTY (only for transmission) 2: LEMPTY 3: EEMPTY For details, see **Table 26.86, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).

CAUTION

When the descriptor is an extended descriptor, it has a 12-byte unused area.

(6) How to Use Frame Data Descriptors

The descriptor data area size field (DESCR.DS) can specify up to 2048 bytes of Ethernet frame data per data area. Settings higher than 2048 (bytes) cannot be made.

In general, Ethernet frames are not of uniform length. The AVB-DMAC is capable of dividing frame data into multiple descriptors in order to minimize the memory capacity for frame data. This function allows processing of frames that are longer than the limit for descriptor data areas. Frames divided based on a frame data structure can also be supported.

To handle both frames divided up into multiple data areas and descriptors for complete single frames, four types (DESCR.DT) FSTART, FEND, FMID, and FSINGLE are defined.

Figure 26.19, Mapping of Frame Data shows the mapping of frame data by frame data descriptors. The descriptor data areas are allocated to the URAM. For frames that require division into four or more data areas, additional FMID descriptors can be added as required.

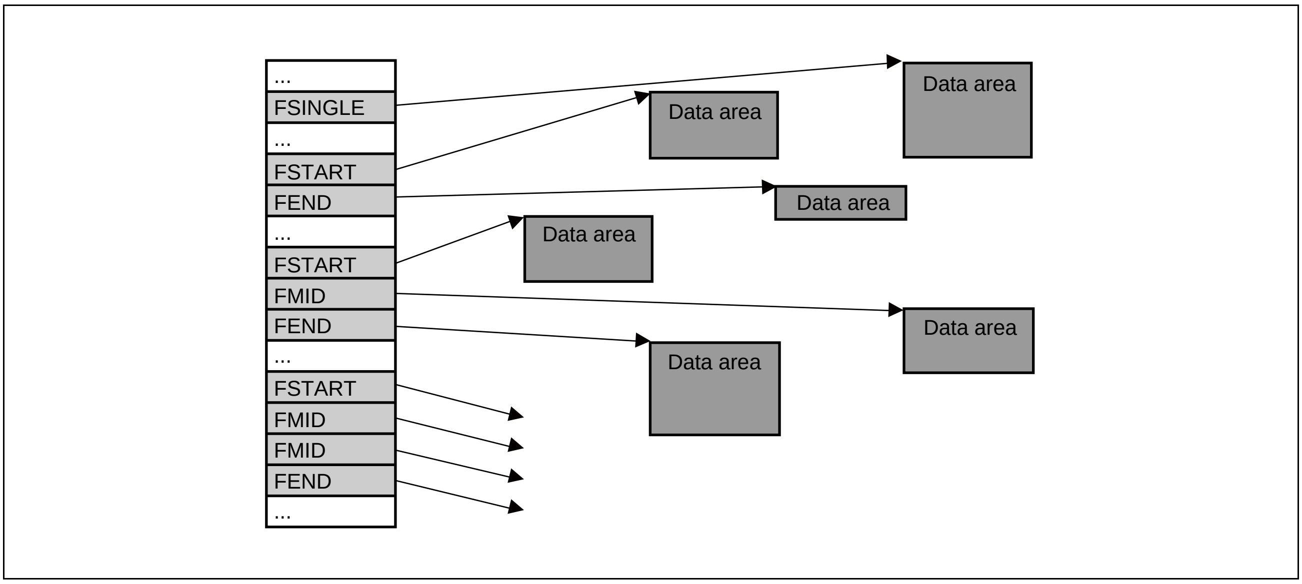


Figure 26.19 Mapping of Frame Data

For reception, set the descriptor data areas to the maximum size (i.e. give DESCR.DS its maximum value). The AVB-DMAC will store received frame data in the given area. If a received frame has more data than the maximum size, the AVB-DMAC will divide the data up.

For transmission, set the frame data size to the actual data size. The AVB-DMAC modifies the descriptor type (DESCR.DT) to FEMPTY after processing the relevant descriptor. The data size (DESCR.DS) and descriptor pointer (DESCR.PTR) fields retain their settings.

A descriptor data area including unused space produces an empty space between frame data. In reception, an “incremental data area” can be used to prevent empty spaces. For incremental data areas, see **Section 26.4.4.3(2), Incremental Data Areas**.

As well as reducing the memory capacity taken up by the descriptor area in the URAM, division into frames can be used to identify different sections of data (e.g. for separating a header and data).

(7) How to Use Chain Control Descriptors

(a) Link Descriptors

The link descriptors can be used to set up cyclic descriptor chains (for details, see **Section 26.4.3.2, Using Descriptor Chains in Queues**)

After a LINK descriptor is processed, its descriptor type (DESCR.DT) is changed to LEMPTY. The descriptor pointer (DESCR.PTR) retains its setting.

After processing of a LINKFIX descriptor, the descriptor type (DESCR.DT) is not updated. Software can change the descriptor type (DESCR.DT), descriptor interrupt enable (DESCR.DIE), and descriptor pointer (DESCR.DPTR). Take care, however, to check the current descriptor address register q (ETNBnCDARq.CDA) before changing the descriptor pointer (DESCR.DPTR).

(b) EOS Descriptor

Use the EOS descriptor to divide a descriptor chain into various segments. The queue can continue even after an EOS descriptor.

In transmission, the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQq) is cleared.

In reception, when the frame currently received is one divided for storage (e.g., received data whose frames are the FMID type or FEND type for storage), the data is not fully stored and the receive queue full interrupt (ETNBnRIS2.QFFr) is generated.

(8) How to Use Hardware and Software Arbitration Descriptors

In hardware processing of descriptors, the empty descriptor types (FEMPTY, LEMPTY, and EEMPTY) are used to distinguish various descriptors. For software, they can be used to initiate checking for empty spaces, etc.

(a) FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND

These descriptor types (DESCR.DT) are used for descriptors that do not contain effective data. Of these, only FEMPTY is used in transmission.

(b) LEMPTY

This descriptor type (DESCR.DT) is assigned to LINK descriptors after they have been processed. The descriptor pointer (DESCR.DPTR) of an LEMPTY descriptor still points to the linked descriptor.

(c) EEMPTY

This descriptor type (DESCR.DT) is assigned to EOS descriptors after they have been processed. The descriptor pointer (DESCR.DPTR) of an EEMPTY descriptor is not used.

(9) Synchronization between Descriptor Access by Hardware and Software

In primary HW/SW synchronization, the descriptor type (DESCR.DT) allocated to the URAM is usable. This makes it possible to minimize access by the AVB-DMAC to the SFR via the CPU, leading to higher performance.

Basic concepts of synchronization:

- Descriptor sets are allocated so that it is exclusively used by hardware or software, depending on the direction of transmission (see **Table 26.86, Summary of Descriptor Types**).
- Software must not change a descriptor assigned to hardware processing (the hardware does not change descriptors assigned to software processing).

In the case of software processing, the software must process the information in the descriptor and the corresponding frame data before changing the descriptor type. If a descriptor type assigned to hardware is set in DESCR.DT, the software should not change any part of the descriptor or of the corresponding frame data.

26.4.3.7 Tips for Optimizing Performance in Handling Descriptors

The following items are recommended as ways to ensure the optimal use of data structures in the URAM.

They are not requirements, but using a different approach may increase the load on the system bus within the LSI chip.

- Register descriptors with 64-bit alignment (this does not apply to extended descriptors).
- While in operation mode, use LINKFIX instead of LINK whenever a descriptor need not be changed. Hardware modifies the descriptor type (DESCR.DT) fields of LINK descriptors.
- Make an access to frame data in units of 128-byte blocks maximum.
- Design the descriptor chains in ways that minimize parallelism of processing.
  This helps in dividing the chains into segments allocated to different cache pages, and in arranging the different segments exclusively for access by software or hardware.
- Minimize the number of divided frames. This can reduce the overhead of descriptor handling.

26.4.4 Control in Reception

The point of the AVB-DMAC is to transfer data between the E-MAC and URAM without intervention by the CPU.

Create descriptors that define the amounts of frame data to be stored and the locations. After the E- MAC receives a frame, it stores the received frame data and the conditions of reception as the MAC state. If the descriptor is extended, the timestamp is also stored. For a description of how to set up descriptors for use in reception, see **Section 26.4.4.2, Setting Up Reception Descriptors**.

The AVB-DMAC filters received frames to separate them into various classifications (separation filtering). More specifically, this is done to separate received frames into the various reception queues and to set the priorities of different classes of received frames. For more on separation filtering, see **Section 26.4.4.1(1), Separation Filtering**.

Figure 26.20, Mechanism of General Reception Queue Selection shows the reception data bus and the selection of queues for use in reception.

Each frame received from the E-MAC is stored in the reception FIFO; in parallel with this, the frame is analyzed to identify its type and the target queue number. After the E-MAC completes reception, the target queue number is generated and stored in the reception FIFO. Appending of a reception flag depends on the storage of one frame among the reception queues in the URAM, and the unread frame counter (UFC) is also associated with frame storage.

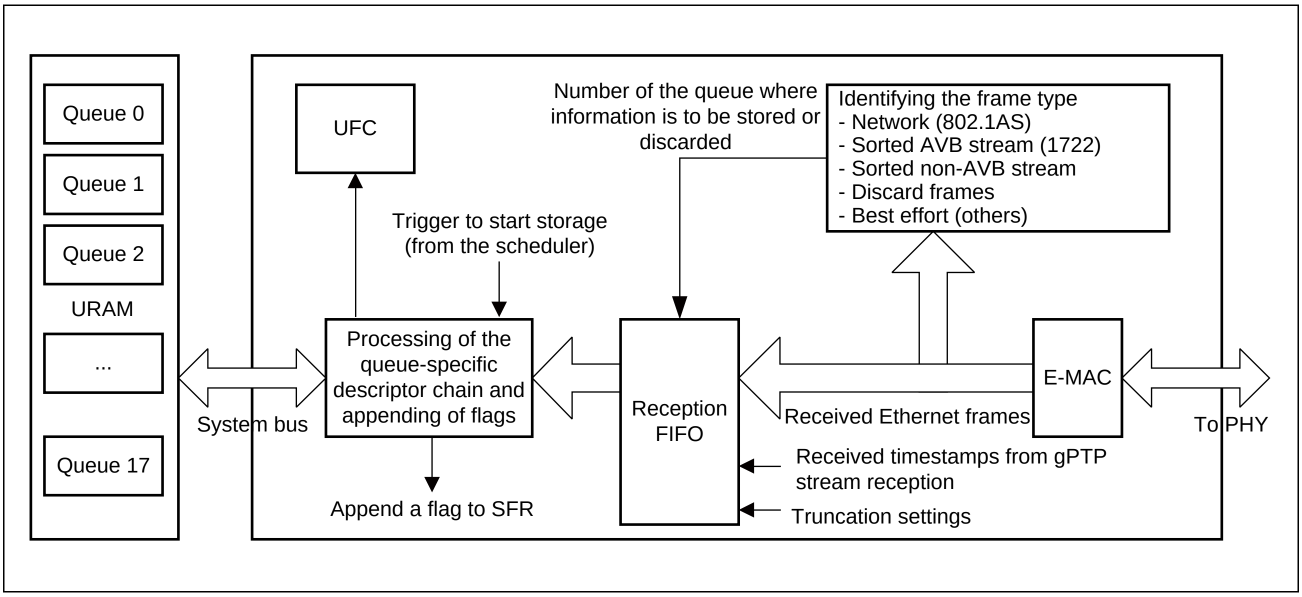


Figure 26.20 Mechanism of General Reception Queue Selection

26.4.4.1 Reception Queues

The AVB-DMAC applies its separation filtering mechanism to select the reception queue for storing a received frame. The AVB-DMAC stores all received frames in the URAM.

There are two conditions for the AVB-DMAC to discard a received frame.

Detection of an Error During Reception by the E-MAC

- Whether error frames are discarded or stored in reception queue 0 (best effort) depends on the setting of the error frame enable bit in the receive configuration register (ETNBnRCR.EFFS). If error frames are to be stored (ETNBnRCR.EFFE = 1), they are always stored in queue 0 (best effort). In this case, characteristics specific to the queue (e.g. truncation) will vary. If the storage of timestamps for reception queue 0 (best effort) is enabled (the timestamp enable bit in the receive configuration register ETNBnRCR.ETS0 = 1), timestamps are stored even for error frames.
- The separation filter is unable to determine where the frame data should be stored.

Receive Frame Failure in the Separation Filter

- Whether discarding error frames or storing in reception queue 0 (best effort) depends on the stream filtering select bit (ETNBnRCR.ESF) in the receive configuration register.

The flowchart in **Figure 26.21, Mechanism of Reception Queue Selection** shows how the AVB-DMAC selects the reception queue in accord with the frame type, including judgment by the separation filter. Selection of the queue starts when the E- MAC completes frame reception. The result is storage of the frame in the proper queue or the frame being discarded.

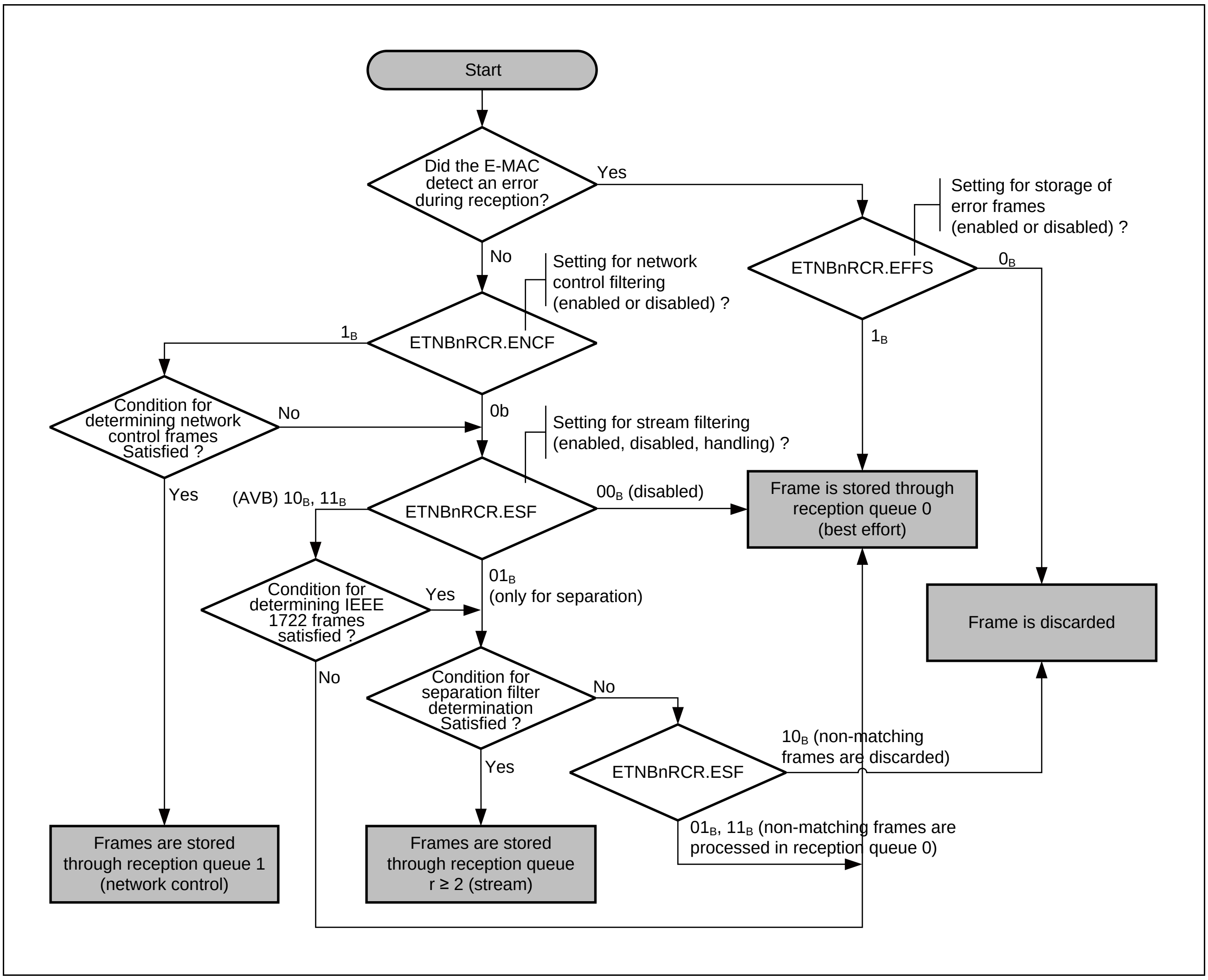
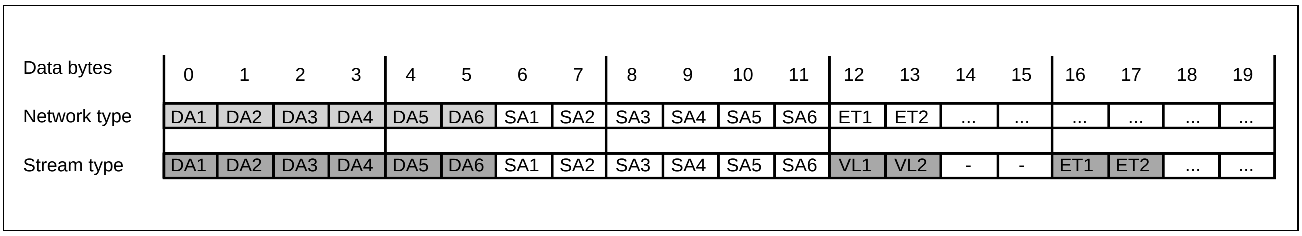


Figure 26.21 Mechanism of Reception Queue Selection

Notes on the Meanings of Entries in the Flowchart

- “Condition for determining network control frames”
  The Ethernet destination address (DA) is 01:80:C2:00:00:0E.
  The Ethernet type (ET) is 88:F7.
- “Condition for determining IEEE 1722 frames”
  The Ethernet destination address (DA) is within the range from 91:E0:F0:00:00:00 to 91:E0:F0:00:FE:FF.
  The VLAN tagged TPID (tag protocol identifier) field (VL) is 81:00. The Ethernet type (ET) is 22:F0.
- “Condition for separation filter determination”
  See **Section 26.4.4.1 (1), Separation Filtering**.

Figure 26.22, Data Bytes of Ethernet Frames Used in Classification shows the allocation of bits related to the network and stream types in Ethernet frames. The preambles of Ethernet frames are not taken into account.



Data bytes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Network type	DA1	DA2	DA3	DA4	DA5	DA6	SA1	SA2	SA3	SA4	SA5	SA6	ET1	ET2	...	...	...	...	...	...
Stream type	DA1	DA2	DA3	DA4	DA5	DA6	SA1	SA2	SA3	SA4	SA5	SA6	VL1	VL2	-	-	ET1	ET2	...	...

Figure 26.22 Data Bytes of Ethernet Frames Used in Classification

(1) Separation Filtering

Separation filtering involves the checking of up to 64 bits (eight successive bytes) in received Ethernet frames. The setting for the first byte (i.e. the setting of the separation filter offset configuration register (ETNBnSFO.FBP)), selects the part of frames to be used in separation filtering. There is also a common filter mask (separation filter mask configuration register i(ETNBnSFMi.CFM)), which limits the separation filter processing to a bit mask that can be set to the lower number of bytes or can be set appropriately.

Examples

To use one byte in separation, set separation filter mask configuration register 0 (ETNBnSFM0.CFM) to 0000 00FF$_H$ and separation filter mask configuration register 1 (ETNBnSFM1.CFM) to 0000 0000$_H$.

To use seven bytes in separation, set separation filter mask configuration register 0 (ETNBnSFM0.CFM) to FFFF FFFF$_H$ and separation filter mask configuration register 1 (ETNBnSFM1.CFM) to 00FF FFFF$_H$.

CAUTION

If bits at some positions are set to 0b in the separation mask, in order to match with the pattern, the bits at the corresponding positions of the pattern must also be set to 0b. Only those bits in which the separation filter pattern configuration register i (ETNBnSFPi.FPs) setting is equal to the separation filter mask configuration register i (ETNBnSFMi.CFM) are sorted by matching with received data.

Figure 26.23, Separation Filtering shows separation filtering. The selected data from a received frame (Rx_Frame[63:0]) are masked by the common filter mask. As a result, the selected frame data can be obtained. This value is compared with all filter patterns. The separation filter circuit in the AVB-DMAC selects the filter pattern that matches the queue having the lowest index s or selects a flag to indicate that there is no matching separation pattern.

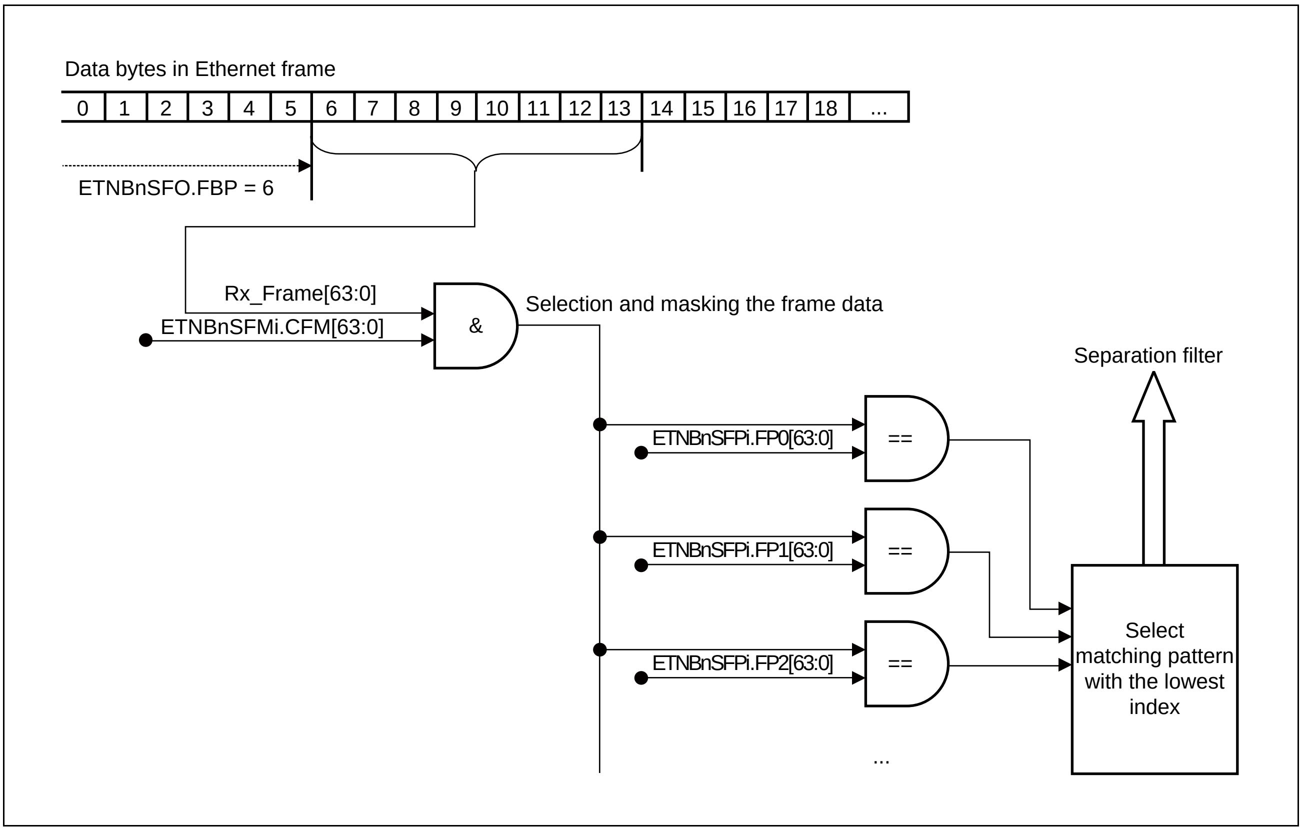


Figure 26.23 Separation Filtering

(2) Stream Separation

The AVB-DMAC applies separation filtering to sort frames received in streams. An AVB network has a concept of “Talker” and “Listener”. A Talker is an end station that generates one or more streams. A Listener is an end station that has the role of being a sink for at least one stream. The various A/V streams are identified by 8-byte stream IDs.

The number of end stations within an AVB network and their roles differ with the application.

The stream ID is a general pattern of the AVB network for identifying one stream. **Figure 26.24, IEEE 1722 Frame Layout and Stream ID** shows the bit allocation of bits in IEEE1722 Ethernet frames and stream ID fields.

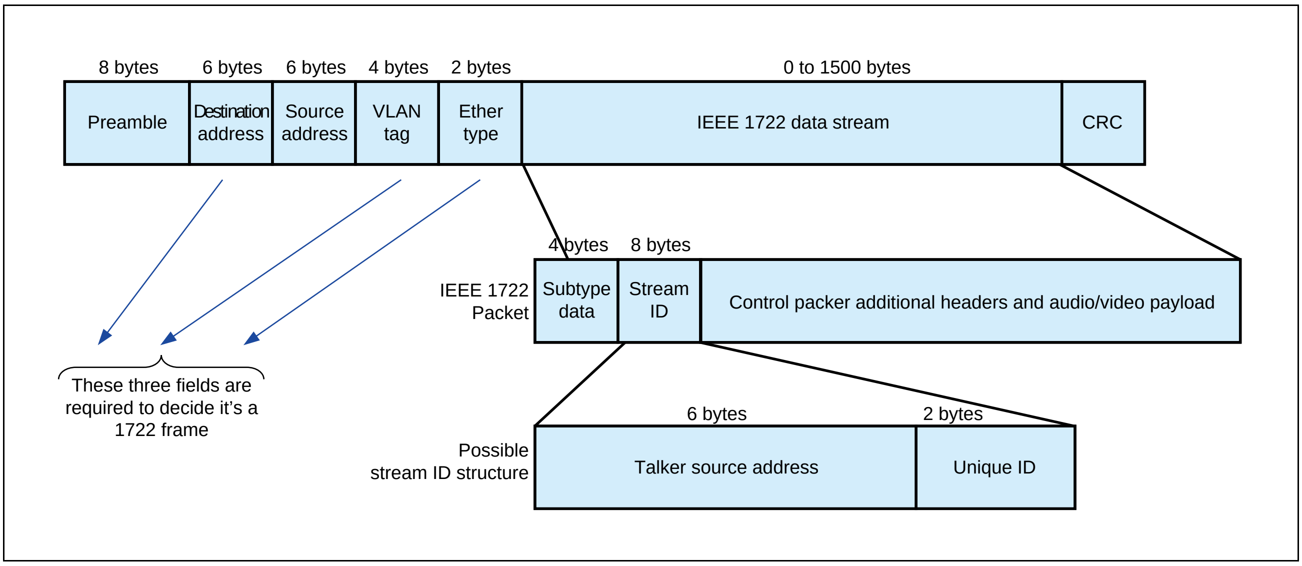


Figure 26.24 IEEE 1722 Frame Layout and Stream ID

The IEEE 1722 standard stipulates that the stream ID field starts from the 23rd byte (not counting the preamble). Accordingly, set the separation filter offset (ETNBnSFO.FBP) to 22 in operations on IEEE 1722 streams. Set the separation filter mask (ETNBnSFMi) and separation filter pattern (ETNBnSFPi) in accord with the specification of the product in which the chip is being used.

Example: In the example of a stream ID shown in **Figure 26.24, IEEE 1722 Frame Layout and Stream ID**, the current application divides the field into the talker source address and the unique stream ID. The unique ID is used to differentiate between multiple streams from the same talker. Based on this, there are two settings for separation filter masking:

- To divide various streams into individual queues, set ETNBnSFM0.CFM to FFFF $FFFF_H$ and ETNBnSFM1.CFM to FFFF $FFFF_H$.
- To divide streams from respective talkers into individual queues, set ETNBnSFM0.CFM to FFFF $FFFF_H$ and ETNBnSFM1.CFM to 0000 $FFFF_H$. This excludes the unique ID from the filter condition.

26.4.4.2 Setting Up Reception Descriptors

For reception, the descriptor mechanism is essentially as described in **Section 26.4.3, Descriptors**.

This section describes memory operations that are especially required in handling reception queues.

(1) Reception Descriptor Types

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 26.91, Descriptor Types in Reception shows the descriptor types used in reception.

Table 26.91 Descriptor Types in Reception

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	No data is stored in receive queues. The ETNBnRIS2.QFFr bit indicates that queue r is full and the received frame is not stored. This descriptor is used again for processing of the next reception, if any.	Not changed
Frame Middle (FMID)	Same as FSTART	Not changed
Frame End (FEND)	Same as FSTART	Not changed
Frame Single (FSINGLE)	Same as FSTART	Not changed
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	A stop point defined by software has been reached. A frame of this type within a divided frame (writing of FMID or FEND) stops the frame being stored and the frame is lost. ETNBnRIS2.QFFr indicates that the frame has been lost. If this happens at the start of a frame (writing of FSTART or FSINGLE), storing of frames starts from the next descriptor. In either case, processing shifts to the next descriptor in the chain.	EEMPTY
Frame Empty (FEMPTY)	The descriptor can be used to store received data. Up to DESCR.DS bytes are stored in the descriptor data area. For details, see **Section 26.4.4.3(1), Storing Frame Data in the Descriptor Data Area**.	FSTART, FMID, FEND, or FSINGLE
Frame Empty Incremental Start (FEMPTY_IS)	The descriptor can be used to store received data. All data for the frame are stored in the descriptor's data area. DESCR.DPTR indicates the base address of the incremental data area. For details, see **Section 26.4.4.3(2), Incremental Data Areas**.	FEND or FSINGLE
Frame Empty Incremental Continue (FEMPTY_IC)	The descriptor can be used to store received data. The remaining bytes of frame data are stored in the descriptor's data area. DESCR.DPTR is undefined, but is written back at the start position within the incremental data area after processing. For details, see **Section 26.4.4.3(2), Incremental Data Areas**.	FEND or FSINGLE
Frame Empty No Data storage (FEMPTY_ND)	The descriptor can be used to store received data. Up to DESCR.DS bytes are captured from the reception FIFO but not stored. After processing, DESCR.DS is written back as 0. For details, see **Section 26.4.4.3(1)(c), No Data are Stored**.	FSTART, FMID, FEND or FSINGLE
Link Empty (LEMPTY)	Same as FSTART	Not changed
EOS Empty (EEMPTY)	Same as FSTART	Not changed

(2) Configuration of Reception Frame Data Descriptors

Figure 26.25, Configuration of Descriptor for a Received Frame shows the configuration of descriptors for use with reception queues. The reception- specific fields are the same whether the descriptor is normal or extended. The reception-specific fields (DESCR.MSC, DESCR.PS, DESCR.EI, and DESCR.TR) are described in **Table 26.92, Configuration of a Received Descriptor**.

For the other fields and the descriptor types, see **Section 26.4.3.6, Descriptor Type**.

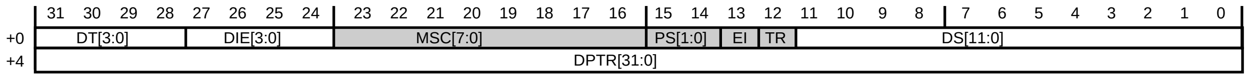


Figure 26.25 Configuration of Descriptor for a Received Frame

Table 26.92 Configuration of a Received Descriptor

Bit Name	Function
MSC	MAC Status Code These bits indicate errors in reception detected by the E-MAC. In the case of a divided frame, these bits are set to the same value within all descriptors for the frame data. Details of the bits are as follows. MSC[7]: Received frame has a multicast address. MSC[6]: Fixed to 0 MSC[5]: Fixed to 0 MSC[4]: Received frame has residual bits. MSC[3]: Received frame is too long*1 MSC[2]: Received frame is too short MSC[1]: Error in frame reception*2 MSC[0]: Received frame has a CRC error.
PS	Padding Selection These bits specify whether frame data are to be padded when stored in the incremental data area. Insertion of padding data is in accord with the settings in the ETNBnRPC register. 00_B: Padding is not to be inserted. 01_B: Padding data may be inserted. This depends on the ETNBnRPC settings. Other settings are prohibited.
EI	Error Indication This bit indicates the detection of an error in frame data while a frame was being stored. The bit is set to 1 for a descriptor in which an error has been detected. If the descriptor is for a divided frame, storage of the frame is aborted. 0: No error 1: Error is detected
TR	Truncation Indication This bit will be set if received data is more than 4092 bytes, and received data will be truncate to 4092 bytes. These bits are set to the same value within all frame data descriptors for a divided frame. 0: Data have not been truncated. 1: Data have been truncated.

Note 1. “Received frame is too long” is set when MAC receive the frame that length is more than ETNBnRFLR configuration value.

Note 2. “Error in frame reception” is set when MAC detect AVB_RX_DV = 1 and AVB_RX_ER = 1 while reception.

CAUTION

The ETNBnRCR.EFFS bit specifies whether or not frames with errors detected by the E-MAC are to be stored in the URAM. When the storing of error frames is disabled, error codes are not written to DESCR.MSC.

26.4.4.3 Reception Processing

After initialization, the AVB-DMAC becomes able to select a proper reception queue and store it in the descriptor data area in the URAM. Received data is stored into the URAM as long as the descriptor data area capacity suffices.

Received frames are classified and stored in the reception FIFO in accord with the algorithm described in **Section 26.4.4.1(1), Separation Filtering**. The frames are sorted, truncated, or discarded by the separation filter on reception by the MAC before stored in the reception FIFO. The following data are stored in the reception FIFO.

- MAC status of received frames
- Length of received frames
- Timestamp of received frames
- Target reception queue
- Received frame data

If the reception FIFO contains even one frame, the scheduler executes storing in the reception queue (see **Section 26.4.2.2, Scheduling Reception and Transmission**).

If there is even one empty data descriptor in a queue for which reception has started, the storage of frame data starts. Received frames for a queue that is already full (there is no empty frame descriptor or the UFC stop level has been reached) are discarded from the reception FIFO. This ensures that one queue being full does not prevent the storage of data in the other queues.

(1) Storing Frame Data in the Descriptor Data Area

Frame data for storage are assumed to be in either of the two patterns described below.

- The data for an entire frame will fit in the descriptor data area.
  - In this case, the descriptor type (DESCR.DT) is FSINGLE.
- Frame data to be stored in the descriptor data area arrive in divided form.
  - In this case, FSTART is written to the descriptor type (DESCR.DT) bits of the first of the frame data to arrive and FMID and FEND are written to the type bits of descriptors for subsequent data.

The descriptor type is updated by the AVB-DMAC in the last step of descriptor processing, so software can always access the descriptor assigned to DESCR.DT.

When normal synchronization mode is used, the CPU can write FEMPTYxxx directly to the descriptor type field after processing the stored element. Do not change the descriptor or any part of the descriptor data area after FEMPTYxxx is written to DESCR.DT.

(a) Storing Frame Data for a Whole Single Frame

For a frame with an FSINGLE descriptor, all data for the frame are held at the position defined by DESCR.DPTR. DESCR.DS indicates the length of the received frame.

If DESCR.DS is bigger than the actual size of a received frame, the FEMPTY or FEMPTY_ND descriptor is stored in place of the FSINGLE descriptor after processing.

Also, the FEMPTY_IS and FEMPTY_IC descriptors, which always hold the full frame data for the reception FIFO, are stored in place of the FSINGLE descriptor after processing.

(b) Storing Frame Data as Divided Frames

Divided frames are handled in the same way as a single frame. A frame stored with divided descriptors must be recombined before use. DESCR.EI and DESCR.TS are only valid in the last descriptor of the sequence for a divided frame.

CAUTION

If the data area size setting in DESCR.DS is not a multiple of four, the number of bytes set in DESCR.DS is fetched from the reception FIFO and the remaining bytes are used as the next storage area.

After a received frame is divided into different descriptors, handle each storage element separately, and allocate the descriptor type, intending software, after processing.

Accordingly, an error frame (FEMPTYxxx instead of FMID or FEND) may exist while a descriptor chain is being processed. In such a case, the CPU must postpone processing of the error frame to the next trigger point.

(c) No Data are Stored

The application specification may lead to some types of received frames being unimportant (for example, when the application only requires stream data from the Ethernet frames). Storing frames in divided form makes separating out the unnecessary parts of Ethernet frames possible.

If part of a divided frame is not required, use the FEMPTY_ND descriptor for that part so that it is not stored in the URAM. Not storing the data negates the need for bandwidth on the data bus, improving the overall performance.

When an FEMPTY_ND descriptor is processed, DESCR.DS is set to 0. This brings the frame data section of the descriptor into agreement with the FEMPTY type. DESCR.DS = 0 is for the unique identification of the descriptor after writing.

(2) Incremental Data Areas

Secure space in the URAM for storing received data. Even when data are placed in the URAM area such that all descriptor data areas of a chain are contiguous, a received frame being shorter than the descriptor data area will lead to an empty space. **Figure 26.26, A Reception Queue Using Individual Descriptor Data Areas** shows an example of settings and the memory map.

Certain applications require that data areas be contiguous (e.g. when received data are to be processed other than by hardware as A/V codec modules). When the length of received frames differs (e.g. when payloads vary between having one or two A/V packages), the use of a static pointer in the descriptor produces empty spaces in the data area. This may necessitate direct additional processing to remove the empty spaces.

Accordingly, and to reduce the CPU load imposed by copying data, the AVB-DMAC supports an “incremental data area” function.

When incremental data areas are in use, all descriptors use a common data area for storage. One descriptor (FEMPTY_IS) defines the base address of the incremental data area and the next descriptor (FEMPTY_IC) in the descriptor chain to store receive data. **Figure 26.27, Reception Queue Using a Common Incremental Data Area** shows an example of settings and the memory map.

Use of an incremental data area does not reduce the memory space in the individual descriptor data areas.

The hardware and software synchronization strategy and performance are also not changed.

It is also possible to divide a frame up among various descriptors in a way that reflects its structure (e.g. one descriptor for the Ethernet header and one for the data payload).

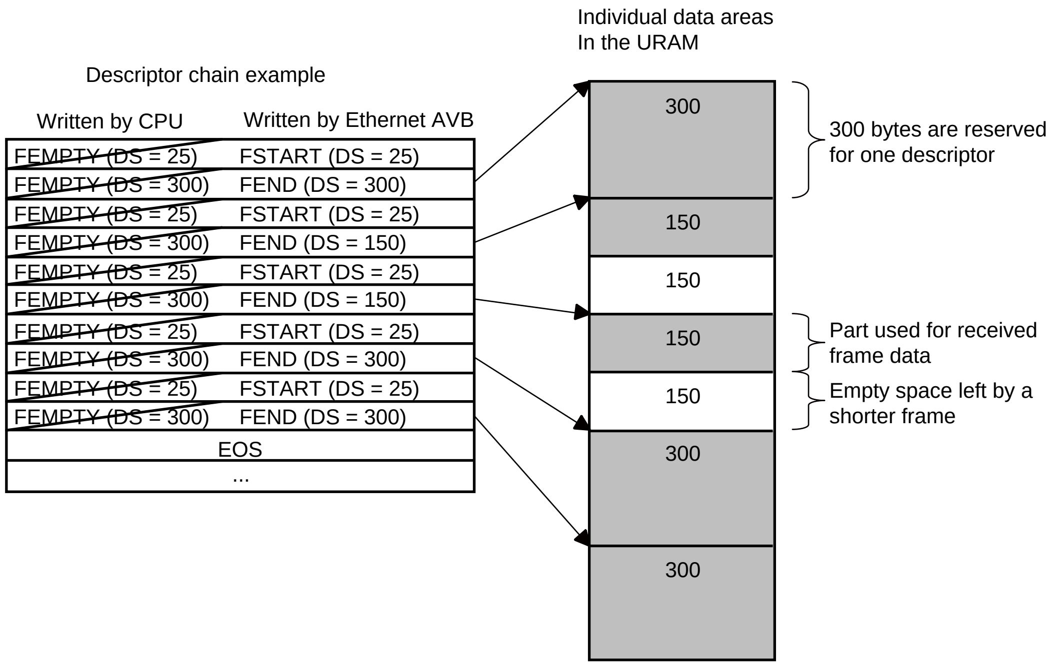


Figure 26.26 A Reception Queue Using Individual Descriptor Data Areas

Figure 26.26, A Reception Queue Using Individual Descriptor Data Areas and **Figure 26.27, Reception Queue Using a Common Incremental Data Area** show how control of the data storage areas by a descriptor chain varies according to whether individual or incremental data areas are in use. The chains are configured for storing received frames consisting of a 25-byte header ((which is treated as one descriptor outside of the range of this example) and a 150- or 300-byte payload (whether one or two 150-byte payload packages are transmitted with one Ethernet frame depends on the data source).

In **Figure 26.26, A Reception Queue Using Individual Descriptor Data Areas**, the EOS descriptor is added as an example of a re-synchronization point. If the frame source transmits a frame containing more than 325 bytes, the frame will be divided among three descriptors, meaning that synchronization of the header and data sequences is lost. Despite this, however, the frame is not divided across the EOS descriptor, so even if the synchronization is off before EOS, AVB-DMAC handles the next descriptor chain as normal. The EOS is not required because the incremental descriptor always stores all data being processed while an incremental data area is in use.

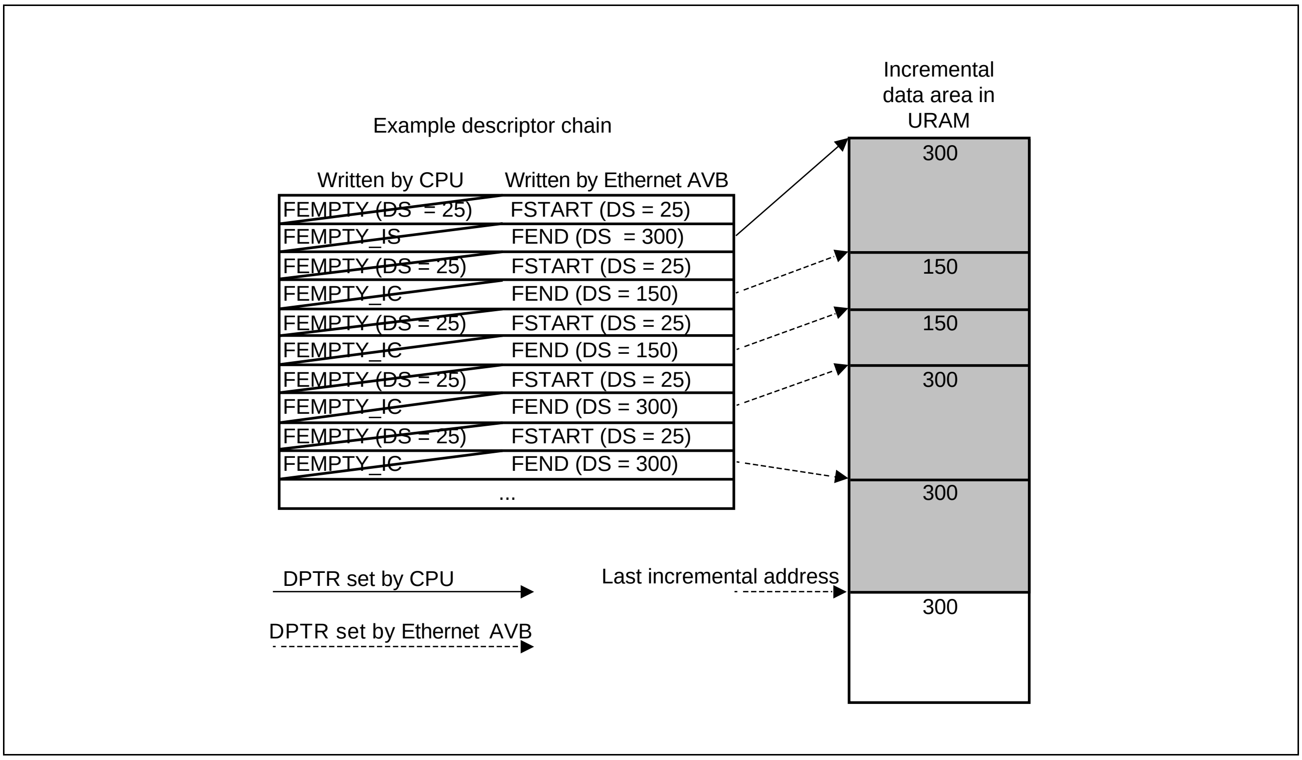


Figure 26.27 Reception Queue Using a Common Incremental Data Area

As **Figure 26.27, Reception Queue Using a Common Incremental Data Area** shows, when data are stored in an incremental data area, the descriptor pointers in the FEMPTY_IC descriptors (DESCR.DPTR) are updated. Accordingly, the resulting FEND or FSINGLE descriptor is in the same format as after writing to an FEMPTY descriptor.

Software captures received data from an incremental data area that has no empty storage areas between frame data. All empty spaces are allocated to the end of the incremental data areas. Incremental data areas are limited to multiples of four bytes. When the amount of data for storage in an incremental data area is not a multiple of four bytes, from one to three bytes of empty space will be produced.
DESCR.DS can be read to check for such empty spaces.

It is not possible to directly control the amount of received data to be stored from the incremental descriptor (FEMPTY_IS, FEMPTY_IC) because other descriptors (FEMPTY and FEMPTY_ND) are also applicable with DESCR.DS. All received data in the chain are always stored in an incremental descriptor.

(a) Setting Up an Incremental Data Area

A descriptor chain in the incremental data area having N descriptors (one FEMPTY_IS and N-1 FEMPTY_IC) means that a storage area for the maximum of N times the capacity must be prepared.

As **Figure 26.27, Reception Queue Using a Common Incremental Data Area** shows, DESCR.DPTR of an FEMPTY_IS descriptor indicates the base address of the incremental data area. The DPTR in FEMPTY_IC descriptor indicates the address of next store data.

(b) Processing an Incremental Data Area Based on Descriptors

Since data processing by the CPU is the same regardless of how the AVB-DMAC stores the data, data stored in an incremental data area do not require any special handling.

(c) Padding

Use padding for received frame data that are not aligned correctly in the specified memory structure. Padding can be set individually for each descriptor. Accordingly, in the reception of divided frames, padding can be restricted to only those frames that require it (e.g. A/V payload data.)

Padding can also be used to optimize system performance in an incremental data area (e.g. to prevent inefficient access by aligning received data with 32-byte boundaries in the incremental data area), as well as to fulfill application-specific requirements for specified memory structures (e.g. formats required by other modules that will be processing the received data).

Padding can only be used in an incremental data area. The value 0000 0000_H is always used in padding.

Padding is the addition of the number of words (from one to seven 32-bit words) set in the stored padding counter in the receive padding configuration register (ETNBnRPC.PCNT). This padding is repeatedly inserted in accord with the value in the stored data counter (ETNBnRPC.DCN) (from one to 255 32-bit words). When the stored data counter (ETNBnRPC.DCNT) reaches 0, however, padding is not repeated.

The first word of padding is always inserted at the position specified by DESCR.DPTR. When divided frames are in use, a padding word can be inserted at any byte position, and padding is handled on a 32- bit basis (e.g. an incremental data area where the first descriptor is for a 42-byte header data and the second descriptor holds padded payload data).

The next figure shows a general example of how padding is inserted and an example of setting up padding. A indicates frame data A received from the E-MAC, while B indicates frame data already stored in the descriptor data area (32-bit word units).

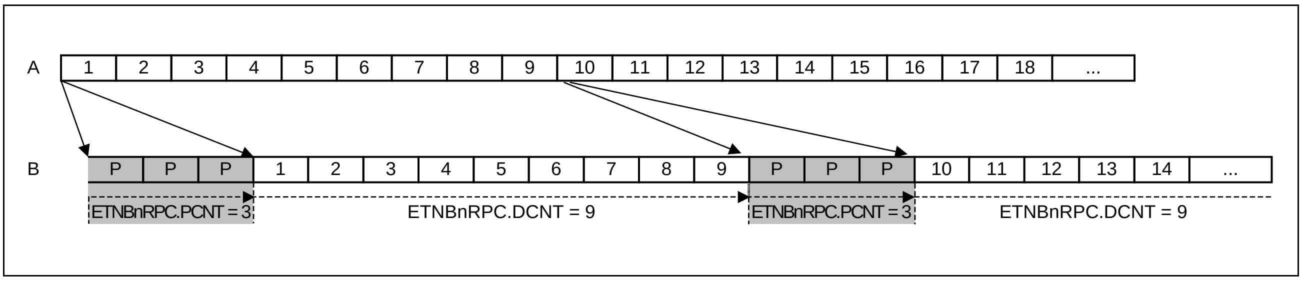


Figure 26.28 Example of a Padding Setting

Both padding and received frame data are counted in the descriptor size (DESCR.DS).

(3) Mode with Write-Back

Constructing a descriptor chain requires software (see **Figure 26.29, Flow of Reception Descriptor Processing (with Write-Back)**).

In the example in the figure, the variable SWdescr (software descriptor pointer) is a structure to identify a descriptor being processed. SWdescr must be initialized after operation mode is entered and a descriptor base address load request (ETNBnDLR.LBAq) is executed (condition for starting the flow of software operations).

The frame_processing() function processes the stored data. The function can use SWdescr.DT to check whether processing of a frame is completed. How frame data are processed differs with the application, so create functions that handle processing in accord with the specification.

The processing section is common to all modes of reception. The number of frames processed in response to each trigger can be restricted. When multiple frames have to be processed in a batch, waiting for individual trigger boxes must be skipped for these frames.

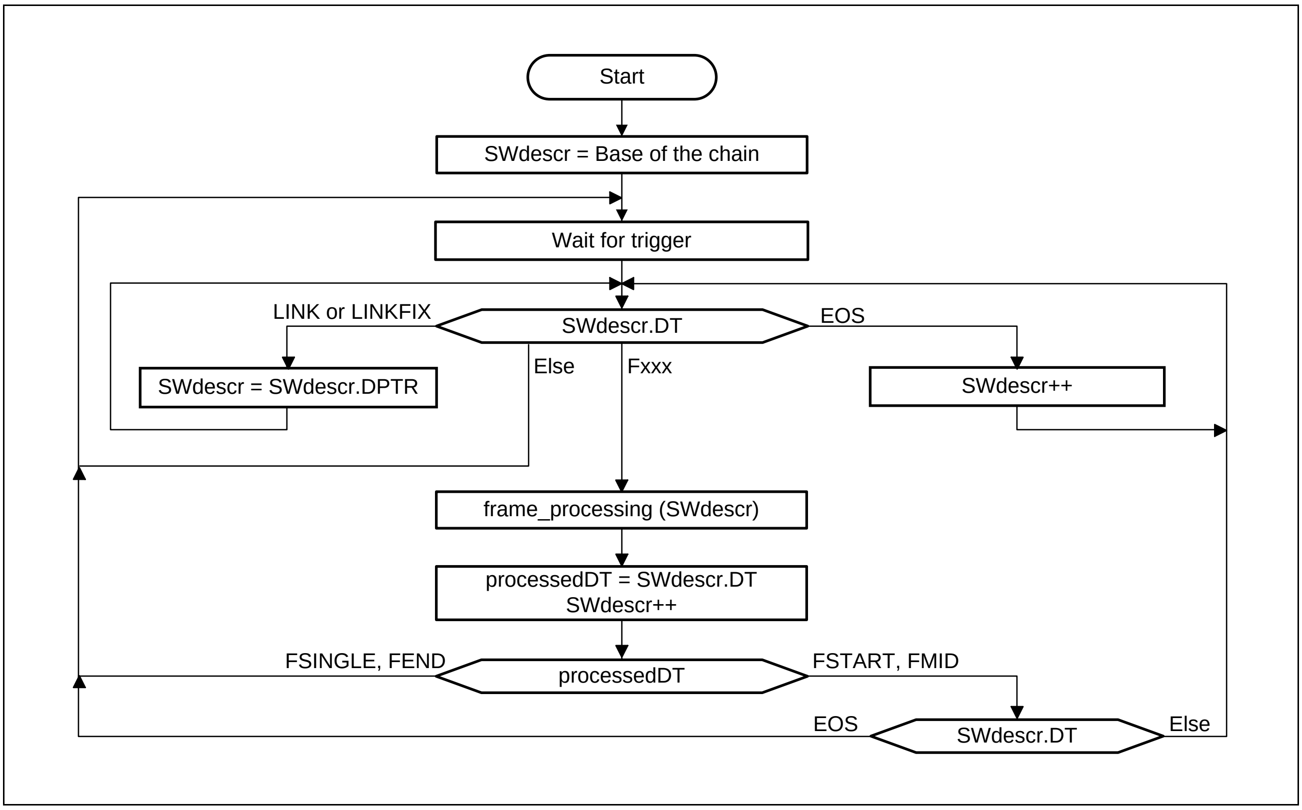


Figure 26.29 Flow of Reception Descriptor Processing (with Write-Back)

(4) Support for Reception Timestamps

Capturing reception timestamps is essential for IEEE 802.1AS time synchronization. Other types of received frames may also require that a reception timestamp be appended; this depends on the application. The AVB-DMAC supports reception timestamps based on the gPTP timer by storing timestamps that have been captured when the frame delimiter (SDF) for a received frame starts in the last frame data descriptor (FEND or FSINGLE). For the gPTP timer, see **Section 26.4.7.1, gPTP Timer**.

When timestamps are to be stored, use extended descriptors for the entire reception queue. Furthermore, timestamps are always stored for reception queue 1 (Network Control). Timestamps for reception queue 0 (best effort) and reception queue r ($r \geq 2$; for stream data) can be selected by the timestamp enable bits in the receive configuration register (ETNBnRCR.ETS0 or ETNBnRCR.ETS2).

26.4.4.4 Unread Frame Counters

Each reception queue has an unread frame counter (ETNBnUFCVi). Use the unread frame counter configuration bits in the receive queue configuration register (ETNBnRQCi.UFCCr) to select from among the four stop levels for each unread frame counter. The 0 setting disables the stop functions. For how to set this up, see **Figure 26.30, Overview of an Unread Frame Counter**.

Operations of the AVB-DMAC (hardware) and CPU (software) drive an unread frame counter (UFC) in the following ways.

- The hardware indicates that it has added a new frame to the descriptor chain for the queue (this increments the counter).
- Software indicates how many frames from the descriptor chain it has processed by writing to the corresponding bits of the unread frame counter decrement register for the queue (this decrements the register by the number written).

The unread frame counter is based on the number of frames stored in the URAM and is only incremented by one even when a received frame is divided into different descriptors. Failure in storing a descriptor chain requires care because this may unread frame counter may fail in synchronization as described in **Section 26.4.4.4(1), Unread Frame (UFC) Synchronization Failure**.

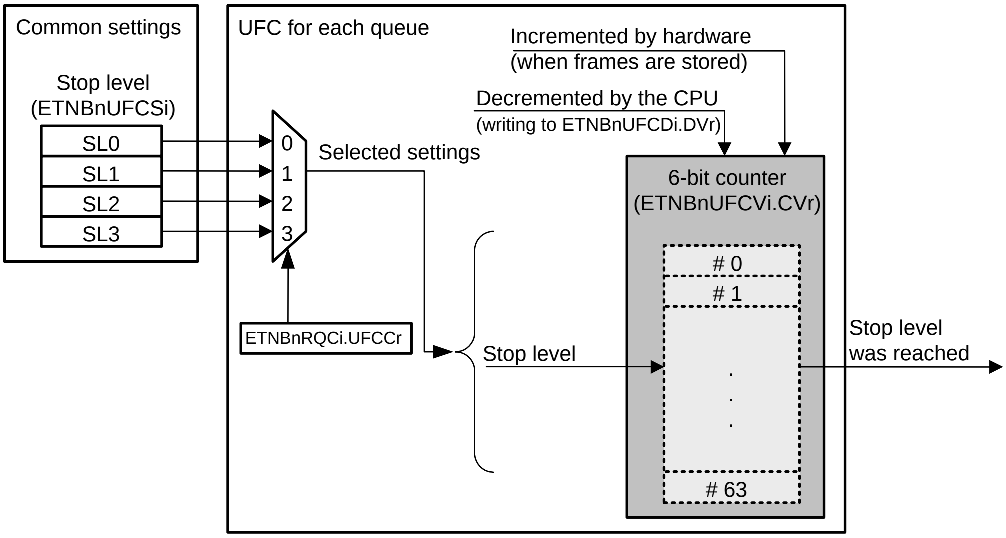


Figure 26.30 Overview of an Unread Frame Counter

Unless synchronization of hardware and software is not established, the current unread frame counter value (ETNBnUFCVi.CVr) indicates the number of unread frames in the queue.

The indicator that the stop level has been reached prevents the storage of further received frames in the descriptor chain. Selecting 0 as the stop level disables this function. Otherwise, further received frames for the queue are discarded once its unread frame counter reaches the stop level. When the unread frame counter stop function activates, the receive queue full interrupt flag in the receive interrupt status register 2 (ETNBnRIS2.QFFr) is set.

Set the unread frame counter stop level configuration register (ETNBnUFCS) for each reception queue that will use the unread frame counter function while the current operating mode is configuration mode.

(1) Unread Frame (UFC) Synchronization Failure

The unread frame counters do not recognize failure to store a frame in the URAM. In other words, the AVB-DMAC increments the counter for a queue each time it captures a frame for that queue from the reception FIFO whether or not it succeeds in storing the frame normally in the descriptor chain.

In general, synchronization of hardware and software fails under the following conditions.

- An unread frame counter reaching its maximum value

  When the value of a counter in an unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) reaches 63, synchronization can fail.
  The CPU can only judge that a failure in synchronization has not occurred when the stop level is set to 63.

- A queue not having enough space for a descriptor or the associated data

  In this case, the corresponding receive queue full interrupt flag (ETNBnRIS2.QFFr) in receive interrupt status register 2 is set.
  If an unread frame counter reaches its stop level write-back mode (ETNBnRQCi.RSM[1:0] = 00_B), the receive queue full interrupt flag (ETNBnRIS2.QFFr) in the receive interrupt status register 2 is set. Software must respond to this.

- A problem occurring during access to memory

  The unread frame counter may set flags to frames over the number of frames actually usable for a descriptor chain, and consequently, it causes synchronization to fail. To retrieve the correct starting point for operations, use the descriptor base address load request (ETNBnDLR.LBAq) for the given queue.

26.4.5 Transmission Control

Areas in the URAM for storing transmission descriptors must also be secured (for descriptors, see **Section 26.4.3, Descriptors**).

The AVB-DMAC fetches data from the URAM in accord with the procedure the descriptor describes. Descriptors are retaining information about tags of transmit frames, too. The tag information is used to maintain the relationships between state information and timestamps for the software and the AVB- DMAC. After completion of transmission of frames, information about the statuses and timestamps of transmitted frames are accessible.

26.4.5.1 Transmission Modes

The AVB-DMAC has two modes of transmission.

- AVB transmission mode

  Selection of the transmit configuration register transmission queue priority (ETNBnTGC.TQP[1:0] bits set to 01_B or 11_B).

- Non-AVB transmission mode

  Selection of the transmit configuration register transmission queue priority (ETNBnTGC.TQP[1:0] bits set to 00_B).

(1) AVB Transmission Mode

AVB transmission supports the control of traffic through the output port to implement various traffic classes.

(a) Support for Traffic Classes and Associated Priority

When transmission is in AVB transmission mode, streams of traffic are transmitted in accord with the part of the AVB specification called Forwarding and Queuing for Time Sensitive Streams (FQTSS; for details on this, see the IEEE 802.1Q standard).

In the AVB specification, at least one queue for a reserving stream under the Stream Reservation Protocol (SR stream) and at least one queue for a non-SR stream are present, and high-priority queues are reserved for SRP traffic.

The AVB-DMAC supports four traffic classes: SR Class A, SR Class B, Network Control (NC) traffic (gPTP frames), and best effort (BE) traffic. Allocating a specific queue to Network Control (NC) frames ensures the control of synchronization.

The AVB-DMAC realizes compliance with the AVB standards by handling queues with the following architecture (in terms of traffic classes).

- Four transmission queues (Q3, Q2, Q1, and Q0) are available.
- Q3 and Q2 are for SR streams (one each for Class A and Class B).
- Q1 is for low-bandwidth Network Control (NC) traffic (gPTP frames)
- Q0 is for other types of traffic (MSRPDU*[1], MVRPDU*[2], best effort (BE), etc.)

Note 1. MSRPDU: Multiple Stream Registration Protocol Data Unit
Note 2. MVRPDU: Multiple VLAN Registration Protocol Data Unit

Fetching from queues proceeds in order of priority of the above traffic types. Three systems of priority are available through the setting of the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP[1:0]). In the default priority scheme, which is called AVB mode 1 (selected by ETNBnTGC.TQP[1:0] = 01_B), operation of the AVB-DMAC is fully in accord with the AVB specification. AVB mode 2 (transmit queue priority bits (ETNBnTGC.TQP[1:0] = 11_B) is an alternative priority scheme and varies from the AVB specification. Using this scheme thus requires more care. The other setting (ETNBnTGC.TQP[1:0] = 00_B) is for non-AVB-mode transmission.

Table 26.93 Default and Alternative Priority Orders in AVB Transmission Mode

Priority Schemes (AVB Mode)	Priority Order of Queues
AVB mode 1 (Default)	Q3 (SR Class A) > Q2 (SR Class B) > Q1 (NC) > Q0 (BE)
AVB mode 2 (Alternative)	Q1 (NC) > Q3 (SR Class A) > Q2 (SR Class B) > Q0 (BE)

(b) Transmission Selecting Algorithm and CBS

The algorithm the AVB-DMAC applies to select frames for transmission is in accord with the specifications under *Section 8.6.8, Transmission selection, of the IEEE 802.1Q standard*. For AVB mode, the CBS (credit-based shaping) algorithm is applied to the Class A and Class B SR queues (Q3 and Q2). Use of the CBS enables correct handling of the priorities of transmission from the SR queues.
For the CBS algorithm, see **Section 26.4.6, CBS (Credit-Based Shaping)**.

When all the following conditions are determined as True, an SR queue (Q3 or Q2) is selected and transmitted at the specified time.

- The queue contains at least one frame ready for transmission.
- The queue has available credit.
- Unless an SR queue satisfies the above conditions, a higher priority queue is not present (not ready for transmission).

A non-SR queue (Q1 or Q0) is selected if the conditions below both hold.

- The queue contains at least one frame ready for transmission.
- As well as the above condition, a higher priority queue is not present (not ready for transmission).

Figure 26.31, Flow of Selection for Transmission in AVB Mode 1 (Default) and **Figure 26.32, Flow of Selection for Transmission in AVB Mode 2 (Alternative)** are flowcharts of selection for transmission in AVB mode 1 (default) and AVB mode 2 (alternative).

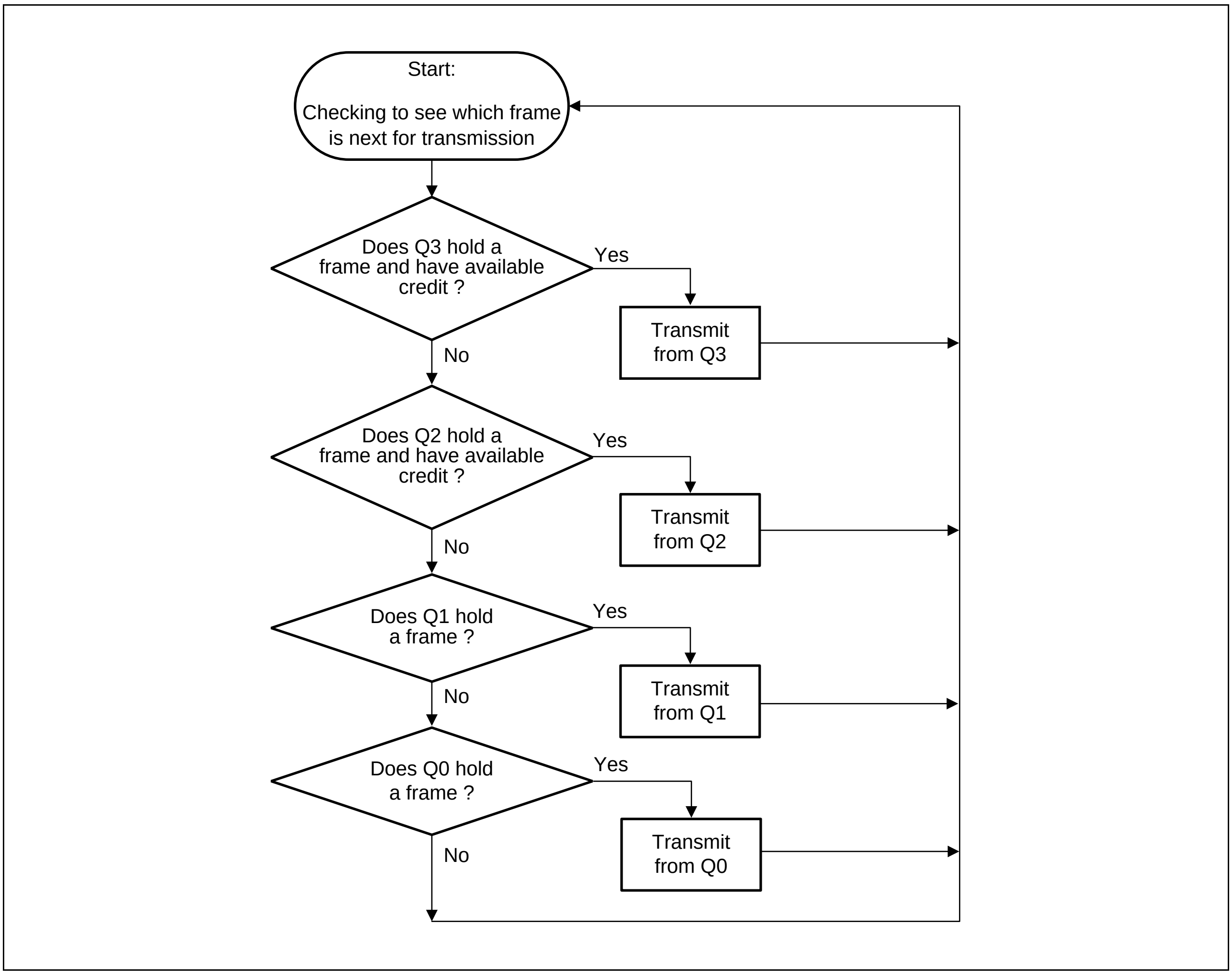


Figure 26.31 Flow of Selection for Transmission in AVB Mode 1 (Default)

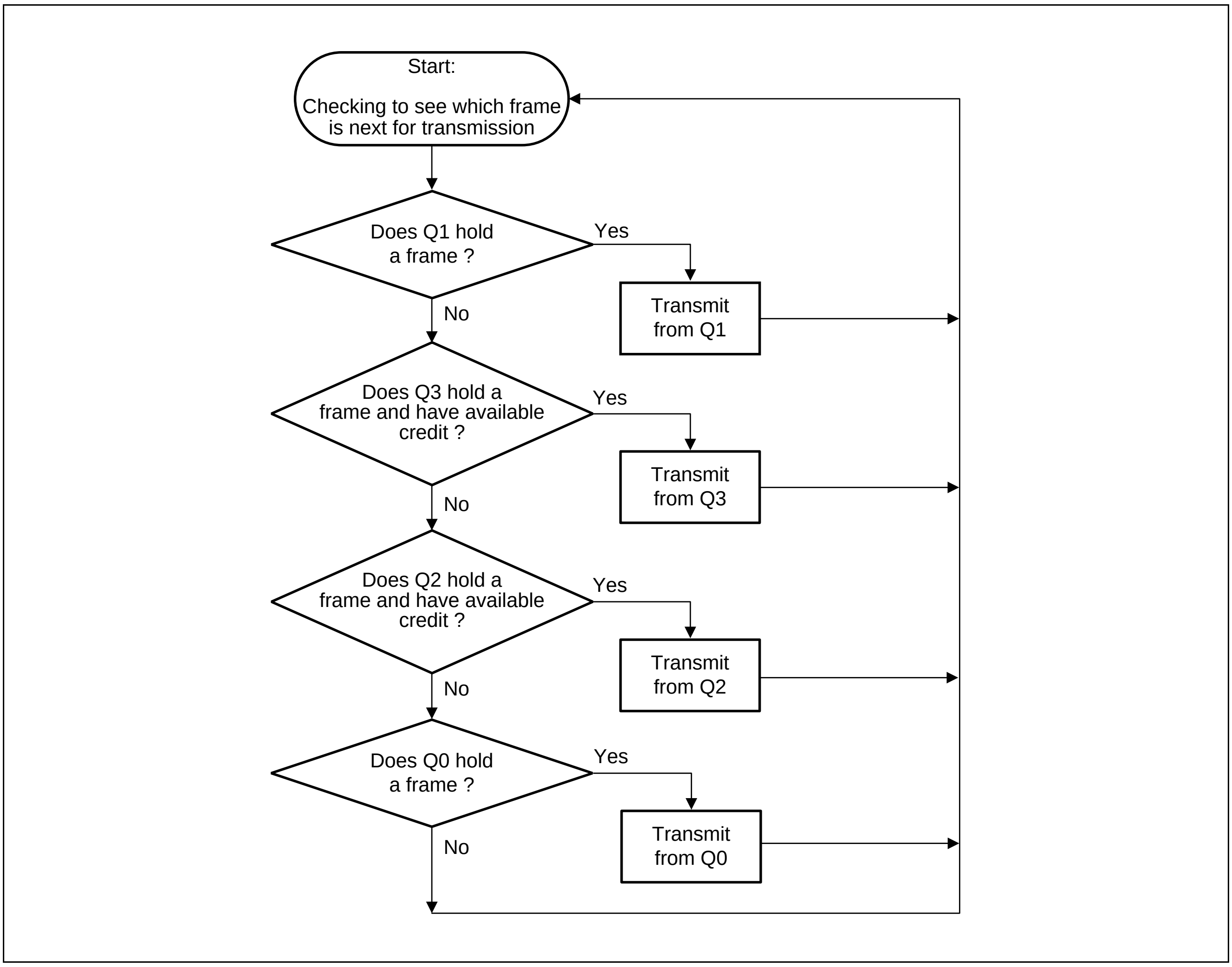


Figure 26.32 Flow of Selection for Transmission in AVB Mode 2 (Alternative)

(2) Non-AVB Transmission Mode

In non-AVB transmission mode, an absolute priority scheme is used. The SR Class is not supported and the CBS algorithm is not used.

In non-AVB transmission mode (when the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP[1:0]) are 00_B), data is fetched for transmission in a strict order of priority (Q3 > Q2 > Q1 > Q0).

Figure 26.33, Flow of Selection for Transmission in Non-AVB Mode shows the flow of selection in non-AVB transmission mode.

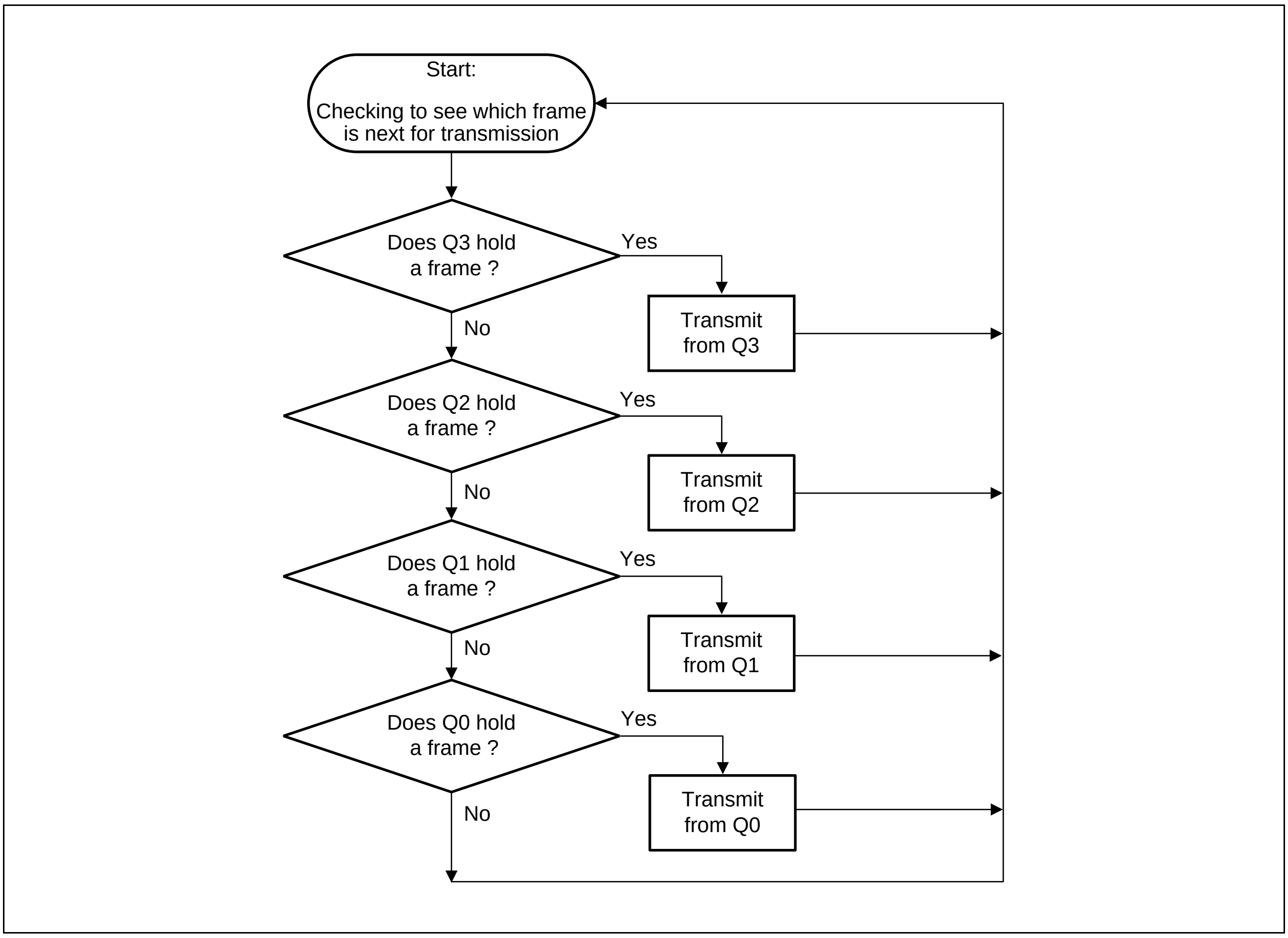


Figure 26.33 Flow of Selection for Transmission in Non-AVB Mode

(3) Setting the Size of the Transmission FIFO

The transmission FIFO is made up of 124 clusters. Each cluster can hold up to 128 bytes.

The size of the part of the transmission FIFO for use by each of the four transmission queues can be set by the corresponding transmit queue configuration t bits in the transmit control register (ETNBnTGC.TBDt). The maximum number of clusters required can be determined from the maximum length of frames for transmission from the queue t.

General Usage Examples:

Q0: Frames containing up to 1500 bytes → 1500/128 = 11.7 → 12 clusters

Q1: Frames containing up to 1024 bytes → 1024/128 = 8.0 → 8 clusters

Q3: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

Q4: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

When the depth of all transmission queues is 2, the following number of clusters is required.
$2 \times (12 + 8 + 16 + 16) + 16 = 2 \times 52 + 16 = 120$

26.4.5.2 Setting Up Transmission Descriptors

(1) Transmission Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 26.94, Descriptor Types in Transmission shows the descriptor types used in transmission.

Table 26.94 Descriptor Types in Transmission

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	The AVB-DMAC fetches the first of the data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame Middle (FMID)	The AVB-DMAC fetches the second or subsequent data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame End (FEND)	The AVB-DMAC fetches the last of the data for the divided frame. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Frame Single (FSINGLE)	The AVB-DMAC fetches the frame of data. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	This is a transmission stop point defined by software This leads to clearing of the transmit start request bit (ETNBnTCCR.TSRQt), which stops transmission. When the ETNBnTCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing proceeds to the next descriptor.	EEMPTY
Frame Empty (FEMPTY)	No frame data are ready for transmission This leads to clearing of the transmit start request bit (ETNBnTCCR.TSRQt), which stops transmission. When the ETNBnTCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing starts at this descriptor.	Not changed
Link Empty (LEMPTY)	Same as FEMPTY	Not changed
EOS Empty (EEMPTY)	Same as FEMPTY	Not changed

(2) Configuration of Transmission Frame Data Descriptors

Figure 26.34, Configuration of Descriptor for a Transmitted Frame shows the configuration of descriptors for use with transmission queues. The transmission-specific fields (DESCR.TSR, and DESCR.TAG) are described in **Table 26.95, Configuration of a Transmission Descriptor**.

For the other fields and the descriptor types, see **Section 26.4.3.6, Descriptor Type**.

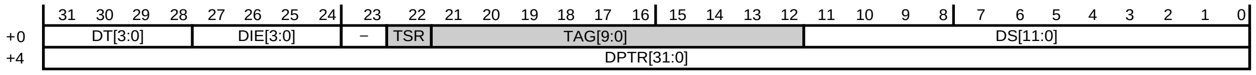


Figure 26.34 Configuration of Descriptor for a Transmitted Frame

Table 26.95 Configuration of a Transmission Descriptor

Bit Name	Function
TSR	Timestamp Store Request This bit specifies whether the transmission timestamp is to be stored within the EthernetAVB module. 0: The timestamp FIFO within the EthernetAVB module does not retain a transmission timestamp. 1: The timestamp FIFO within the EthernetAVB module retains a transmission timestamp. Only control this bit while the current DESCR.DT is FEND or FSINGLE.
TAG	Frame Tag This TAG field is used to associate each frame data with a timestamp. Frame TAG is not required but is recommended. Only control this bit while the current DESCR.DT is FEND or FSINGLE.

For the timestamp FIFO function, see **Section 26.4.5.4, Timestamping in Transmission**.

26.4.5.3 Transmission

(1) Transmitting Frames

Setting the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) starts the transfer of frames to the corresponding transmission queue.

The descriptor to the current descriptor address of the queue (ETNBnCDARq.CDA) is read first.

If this descriptor is a descriptor for frame transmission (FSINGLE, etc.), the AVB-DMAC fetches the frame data from the data area indicated by the descriptor, writes FEMPTY back to the descriptor type (DESCR.DT) bits, then proceeds to processing of the next descriptor.

If the descriptor is not for transmission, processing is as dictated by the given descriptor (for these descriptors, see the descriptions in **Section 26.4.3, Descriptors**).

If a base address load request is issued for a descriptor chain while it is being processed (by setting 1 in the LBAq bit for transmission queue that is currently being processed in the descriptor base address load request register, ETNBnDLR), processing proceeds to the new descriptor chain. Changing the chain does not interrupt frame fetching, but note that frames that have not been fetched from the old chain remain where they are.

Figure 26.35, Descriptor Processing During Transmission shows descriptor processing during transmission.

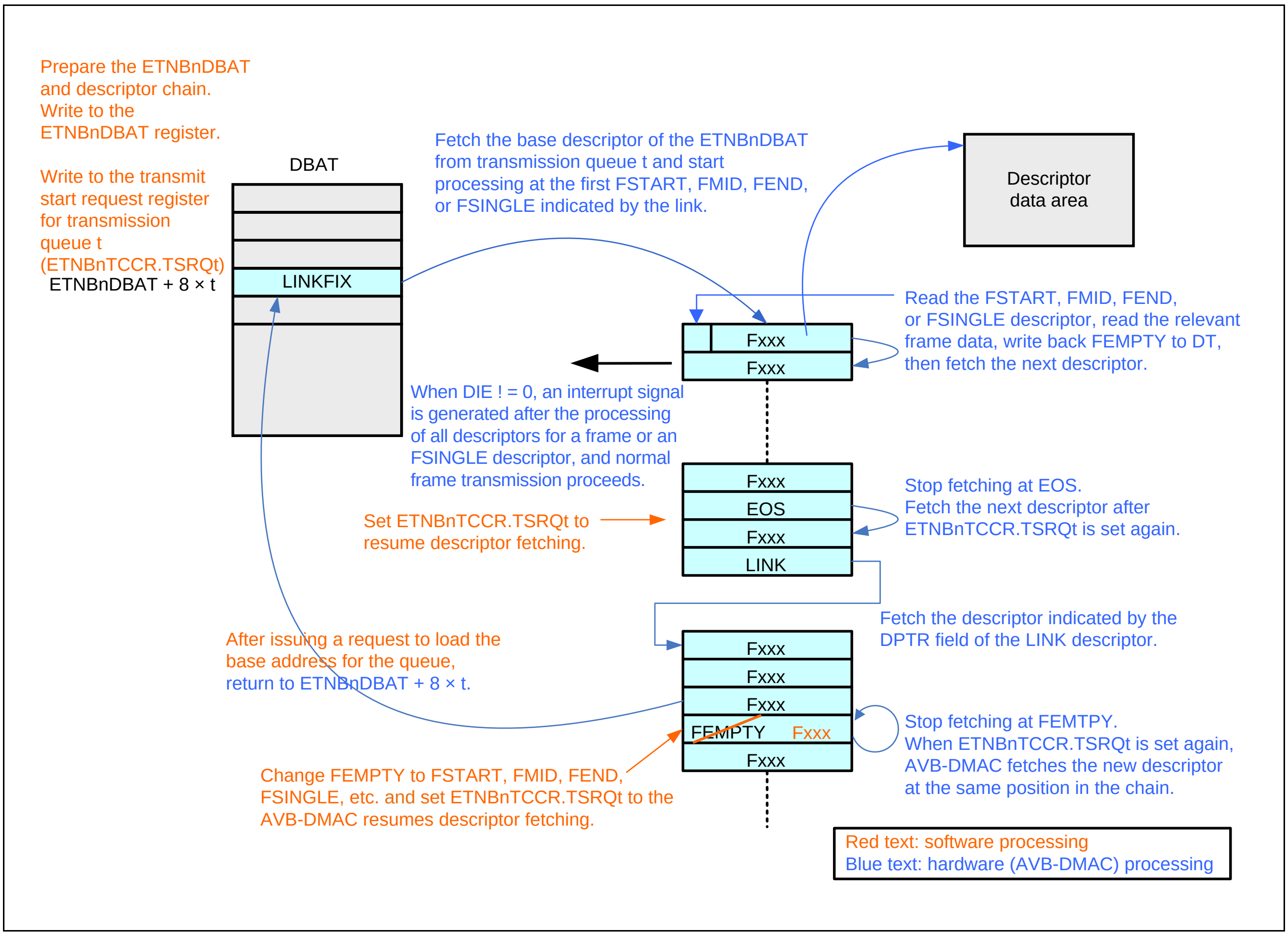


Figure 26.35 Descriptor Processing During Transmission

(2) Examples of Descriptor Usage

(a) Immediate Frame Transmission

Immediate frame transmission is a pattern in which fetching by the AVB-DMAC starts whenever software adds data to a queue. FEMPTY descriptors are used as HW/SW synchronization stop points.
Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 26.36, Software Flow for Immediate Frame Transmission shows the flow for software implementing this pattern.

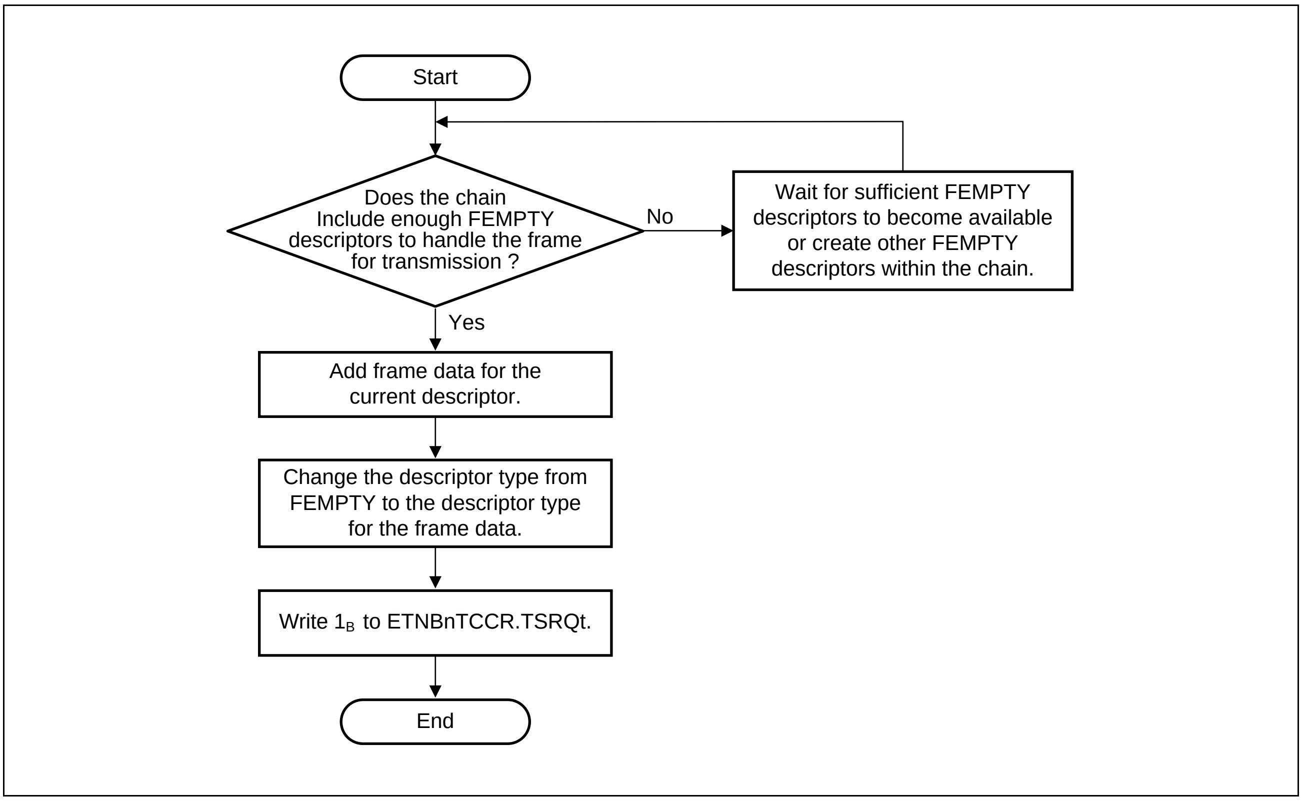


Figure 26.36 Software Flow for Immediate Frame Transmission

Figure 26.37, Software and AVB-DMAC Operations for Immediate Frame Transmission shows software and AVB-DMAC operations for immediate frame transmission.

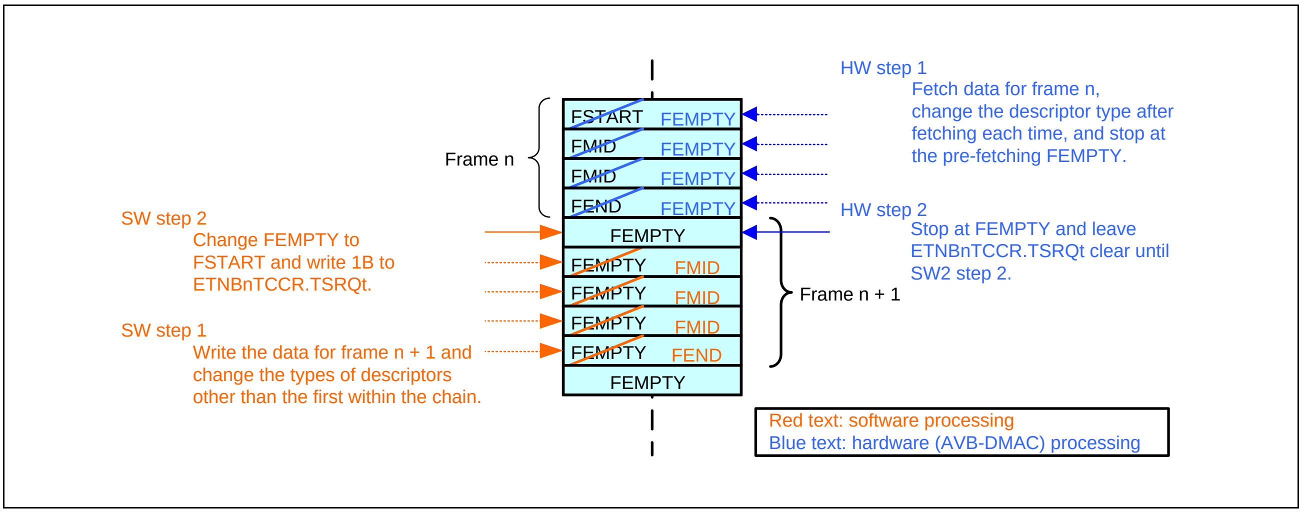


Figure 26.37 Software and AVB-DMAC Operations for Immediate Frame Transmission

(b) Frame Set Transmission with Changing of the Active Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. EOS descriptors are used for the stop points.

Start by creating a descriptor chain that has a FEMPTY descriptor at its stop point.

Figure 26.38, Software Flow for Frame Set Transmission with Changing of the Active Descriptor Chain shows the software flow in this pattern.

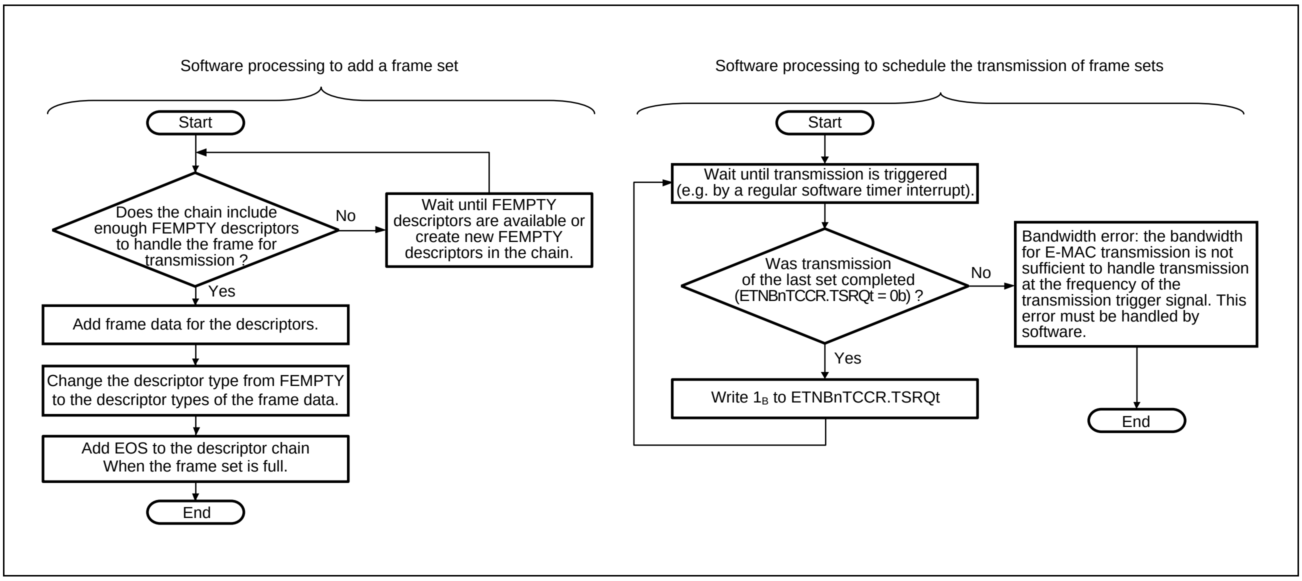


Figure 26.38 Software Flow for Frame Set Transmission with Changing of the Active Descriptor Chain

Figure 26.39, SW and AVB-DMAC Operations for Frame Set Transmission with Changing of the Active Descriptor Chain shows software and AVB-DMAC operations for frame set transmission.

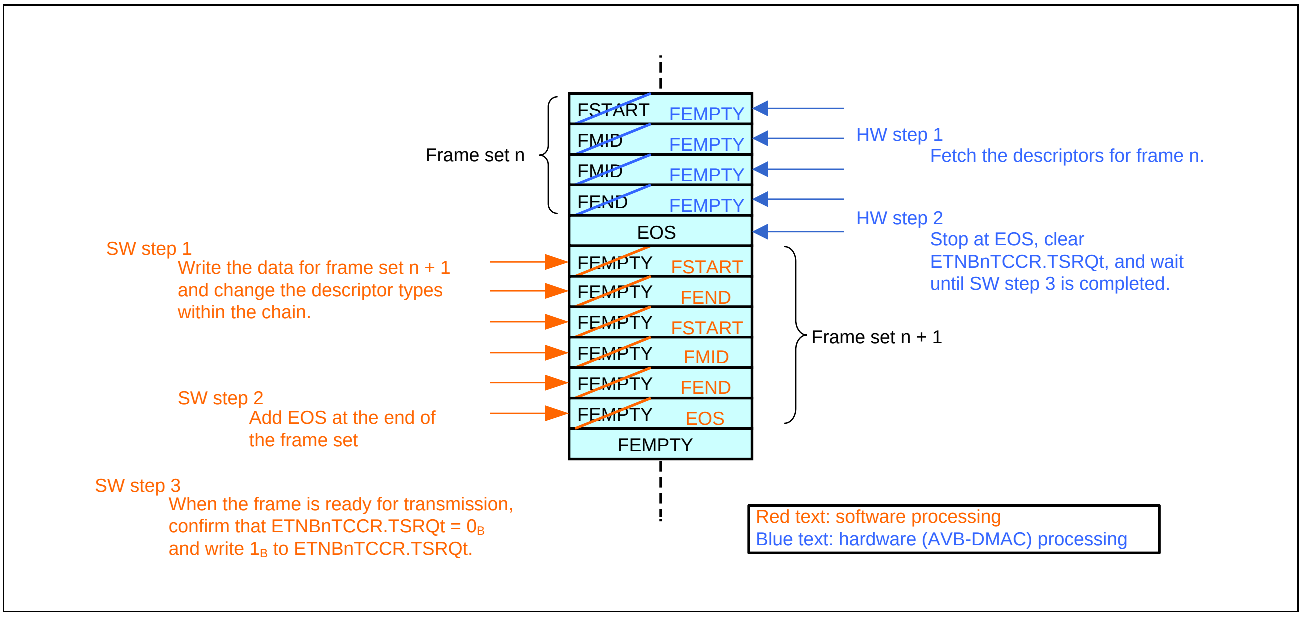


Figure 26.39 SW and AVB-DMAC Operations for Frame Set Transmission with Changing of the Active Descriptor Chain

(c) Frame Set Transmission Using a Shadow Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. Two or more descriptor chains are used. The chains are classified into the active chain and shadow chains. EOS descriptors are used for the stop points.

Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 26.40, Software Flow for Frame Set Transmission Using the Shadow Descriptor Chain shows the flow for software implementing this pattern.

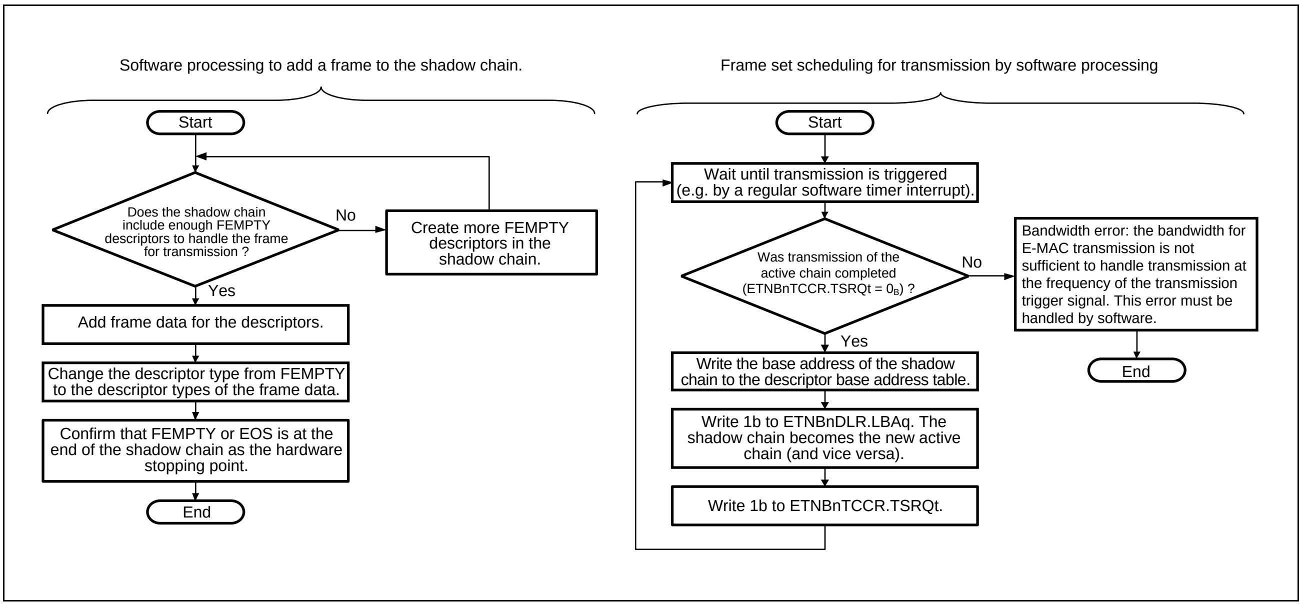
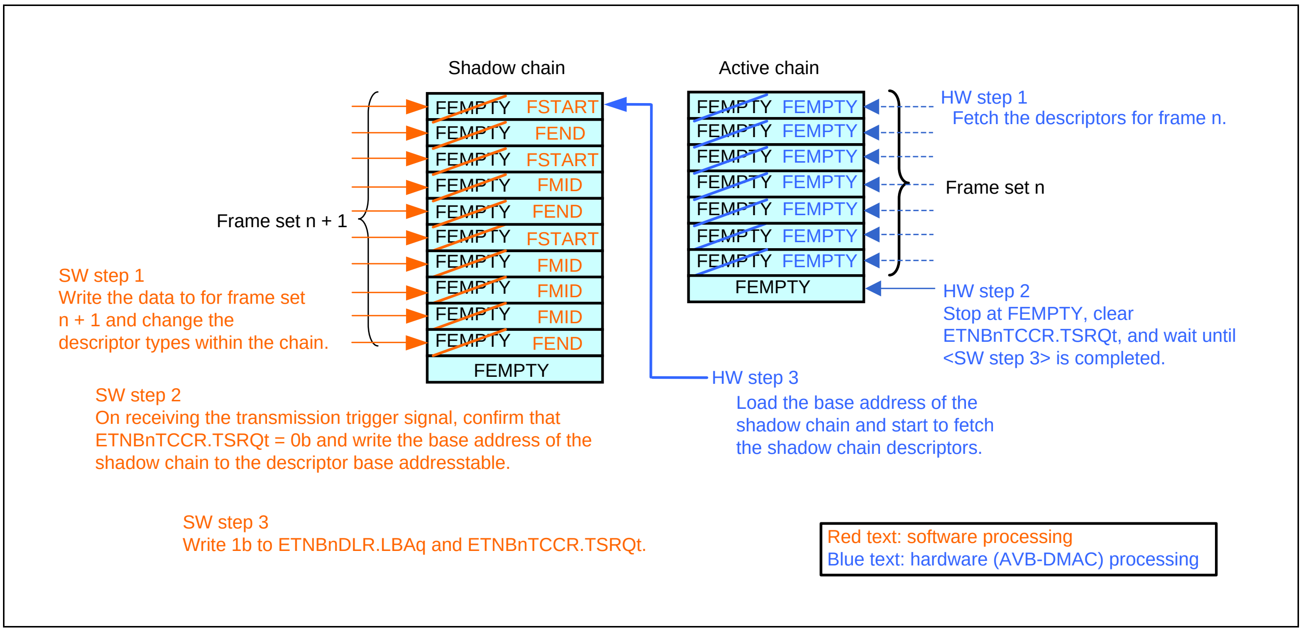


Figure 26.40 Software Flow for Frame Set Transmission Using the Shadow Descriptor Chain

Figure 26.41, SW and AVB-DMAC Operations for Frame Set Transmission Using the Shadow Descriptor Chain shows software and AVB-DMAC operations for frame set transmission.



HW step 1
Fetch the descriptors for frame n.

HW step 2
Stop at FEMPTY, clear ETNBnTCCR.TSRQt, and wait until <SW step 3> is completed.

SW step 1
Write the data to for frame set n + 1 and change the descriptor types within the chain.

HW step 3
Load the base address of the shadow chain and start to fetch the shadow chain descriptors.

SW step 2
On receiving the transmission trigger signal, confirm that ETNBnTCCR.TSRQt = 0b and write the base address of the shadow chain to the descriptor base addresstable.

SW step 3
Write 1b to ETNBnDLR.LBAq and ETNBnTCCR.TSRQt.

Red text: software processing
Blue text: hardware (AVB-DMAC) processing

Figure 26.41 SW and AVB-DMAC Operations for Frame Set Transmission Using the Shadow Descriptor Chain

26.4.5.4 Timestamping in Transmission

Transmission timestamps are important in satisfying the requirements for time synchronization of the IEEE 802.1AS standard. This information can also be useful to other applications and in testing. The AVB-DMAC supports the storage of timestamps for transmitted frames. The timestamp values are based on the gPTP timer and are captured at the same time as sending of the Start of Frame Delimiter (SFD) for transmitted frames.

When the timestamp storage request field (DESCR.TSR) is set to 1, selecting storage of a timestamp, the tag number defined in the tag field (DESCR.TAG) of the last descriptor in a set (FEND) or of an FSINGLE descriptor for the frame being transmitted is stored with the timestamp. To make identification and association easy, the values of timestamp are stored together with tag numbers. The timestamp FIFO is accessible at any time.

Figure 26.42, Mechanism to Support Transmission Timestamps shows the mechanism supporting transmission timestamping.

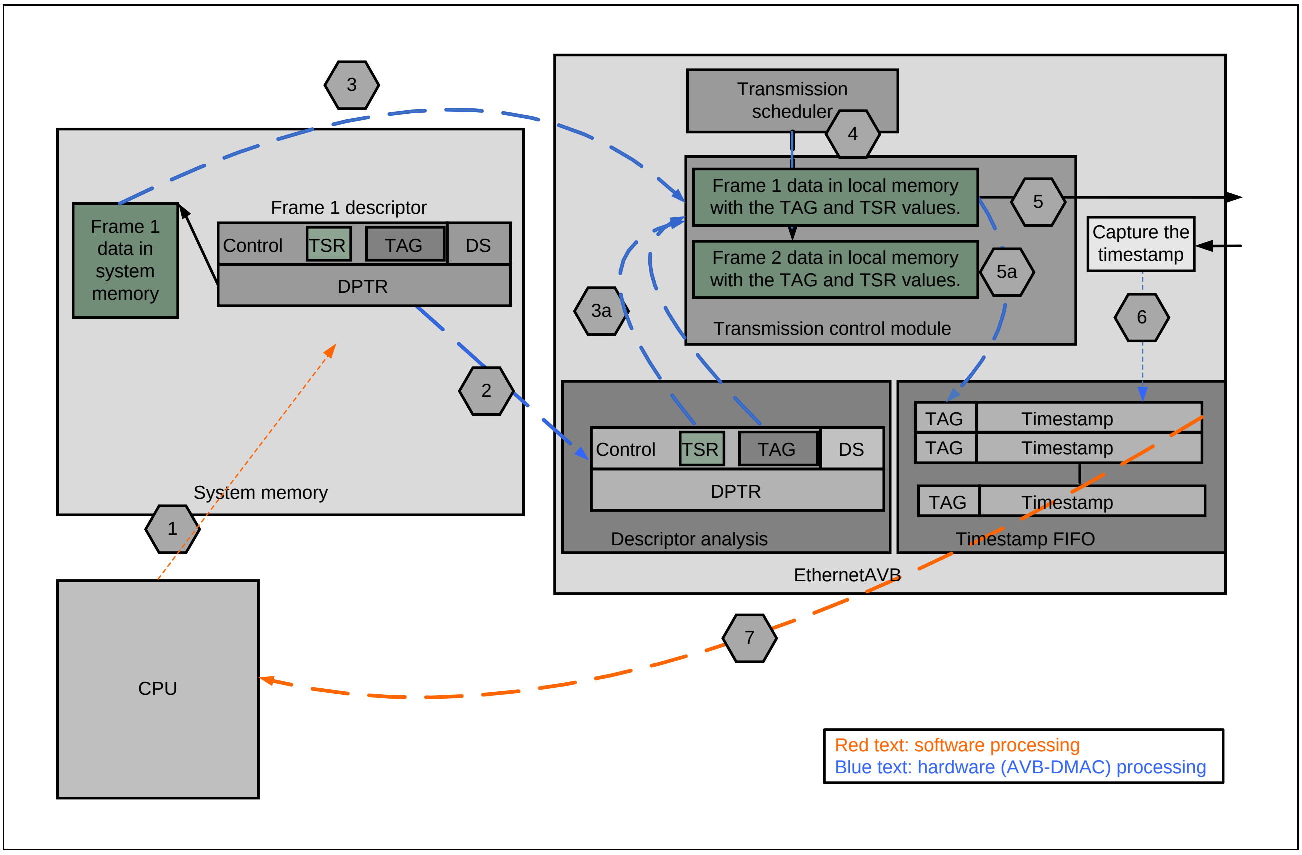


Figure 26.42 Mechanism to Support Transmission Timestamps

The method of using this function is described below:

1. Secure space in the URAM for the frame requiring timestamping.
   Write the tag number of the frame to the frame tag field (DESCR.TAG) and set the timestamp storage request field (DESCR.TSR) to 1.
2. The AVB-DMAC fetches and analyzes the descriptor. The timestamp storage request field (DESCR.TSR) is 1, so it recognizes that transmitting this frame also requires storage of the timestamp.
3. The AVB-DMAC fetches the data for frame 1 and temporarily stores the frame in internal memory for scheduling.
   3a: The frame tag field (DESCR.TAG) and timestamp storage request field (DESCR.TSR) are stored with the fetched data.)
4. Under the control of priority settings according to credit-based shaping (CBS) or another scheme, the transmission scheduler determines the time to transmit frame 1.
5. Transmission of frame 1 starts.
   5a: The information relating to frame 1 is stored in the timestamp FIFO.
6. The gPTP timestamp is captured at the start of sending the frame delimiter (SFD) for transmission and stored with the tag in the timestamp FIFO. On completion of the transmission, an interrupt is generated. For this to happen, the descriptor interrupt control register (ETNBnDIC) must be set beforehand.
7. The entry can now be read from the timestamp FIFO.

Use the timestamp FIFO for the time synchronization of frames with IEEE 802.1AS compliance.

Timestamping can also be used with other frames, but take care not to allow the timestamp FIFO to overflow. When the FIFO is full, further timestamps supplied to it are lost.

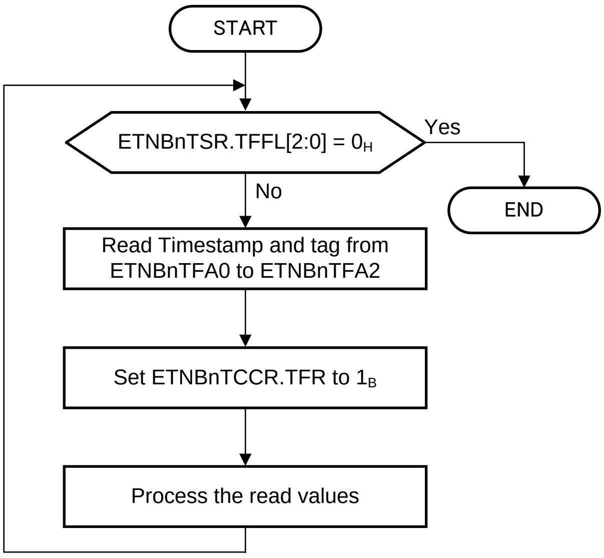


Figure 26.43 Flow of Transmission Timestamping

(1) Ending Transmission

Figure 26.44, Procedures for Ending Transmission shows the procedure for ending transmission.

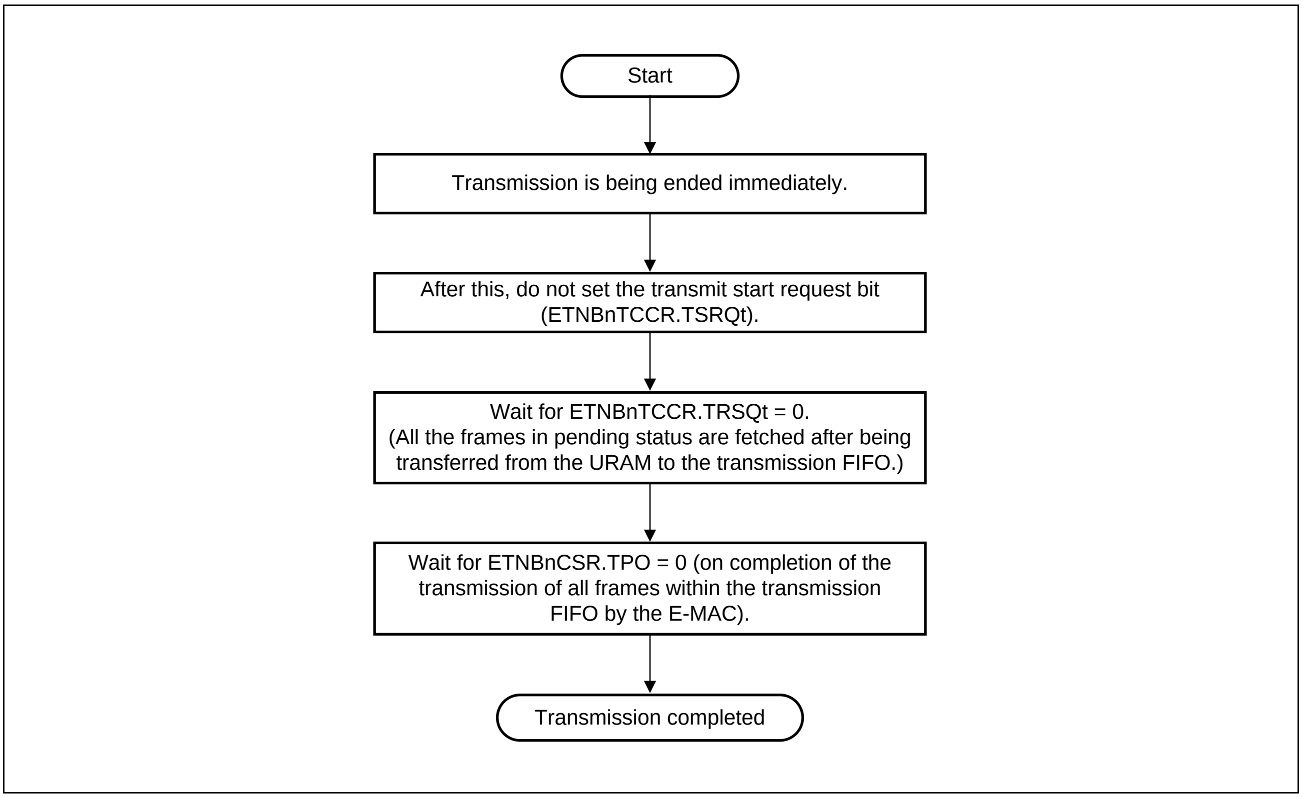


Figure 26.44 Procedures for Ending Transmission

26.4.6 CBS (Credit-Based Shaping)

In AVB transmission mode (i.e. when the transmit queue priority field in the transmit configuration register (ETNBnTGC.TQP) is 01B or 11B), transmission queues Q3 and Q2 are respectively assigned to Class A and Class B stream traffic and the CBS (Credit Based Shaping) algorithm is used to select the transmission queues in order to satisfy the Forwarding and Queuing for Time Sensitive Streams (FQTSS) specification (see *Section 8.6.8* or *Section 34* in *IEEE 802.1Q*).

The CBS algorithm is based on the concept of transmission credit for each queue. Credit can be thought of as the degree to which a queue has the “right” to transmit at a given time. Actually, in AVB transmission mode as specified in IEEE 802.1Q, queues that are subject to the CBS algorithm are able to transmit when the following conditions are met.

- At least one frame is stored in the queue.
- The credit for the queue is 0 or a positive value.

The credit for a transmission queue is incremented while one or more frames from the queue are present in the transmission FIFO but transmission of these frames is not proceeding. This state is indicated by the transmission process status bit for queue t in the AVB-DMAC status register (ETNBnCSR.TPOt) being clear (0). The credit is decremented while transmission of a frame from the queue is in progress. This mechanism is used to control transmission so that the transmission of frames from the queues for each of the traffic classes does not exceed the specified maximum bandwidth.

IEEE 802.1Q defines the following parameters for queues under the control of the CBS algorithm.

portTransmitRate:	Maximum transmission data rate of an external port. The E-MAC determines this parameter.
bandwidthFraction:	Maximum fraction of portTransmitRate that can be used for a queue.
idleSlope:	Rate of change of credit for a queue when transmission of frames from the queue is not proceeding so the credit value (in bits per second) is increasing. idleSlope is also equal to the maximum fraction of the total bandwidth (portTransmitRate) that is available to the given queue under a specified condition (frames from the queue can be placed in a continuous stream. See Annex L of IEEE 802.Q. idleSlope = bandwidthFraction × portTransmitRate
sendSlope:	Rate of change of credit for a queue while transmission of a frame from the queue is in progress so the credit value is decreasing. (in bits per second). The value of sendSlope is defined as follows: sendSlope = idleSlope – portTransmitRate

Furthermore, the values below are used to define individual traffic classes (or queues for the classes) under control of the algorithm. See Annex L of IEEE 802.Q.

maxFrameSize:	Maximum size of frames (in bits) of the corresponding traffic class that can be transmitted from a port
maxInterferenceSize:	Maximum burst size (in bits) by which delays for the corresponding traffic class can be allowed
hiCredit:	Maximum credit value (positive number). Can be calculated by using the following equation: hiCredit = maxInterferenceSize × (idleSlope / portTransmitRate)
loCredit:	Minimum credit value (negative number). Can be calculated by using the following equation: loCredit = maxFrameSize × (sendSlope / portTransmitRate)

Figure 26.45, CBS (Credit-Based Shaping) Operation shows how the CBS algorithm works and the meaning of the above parameters.

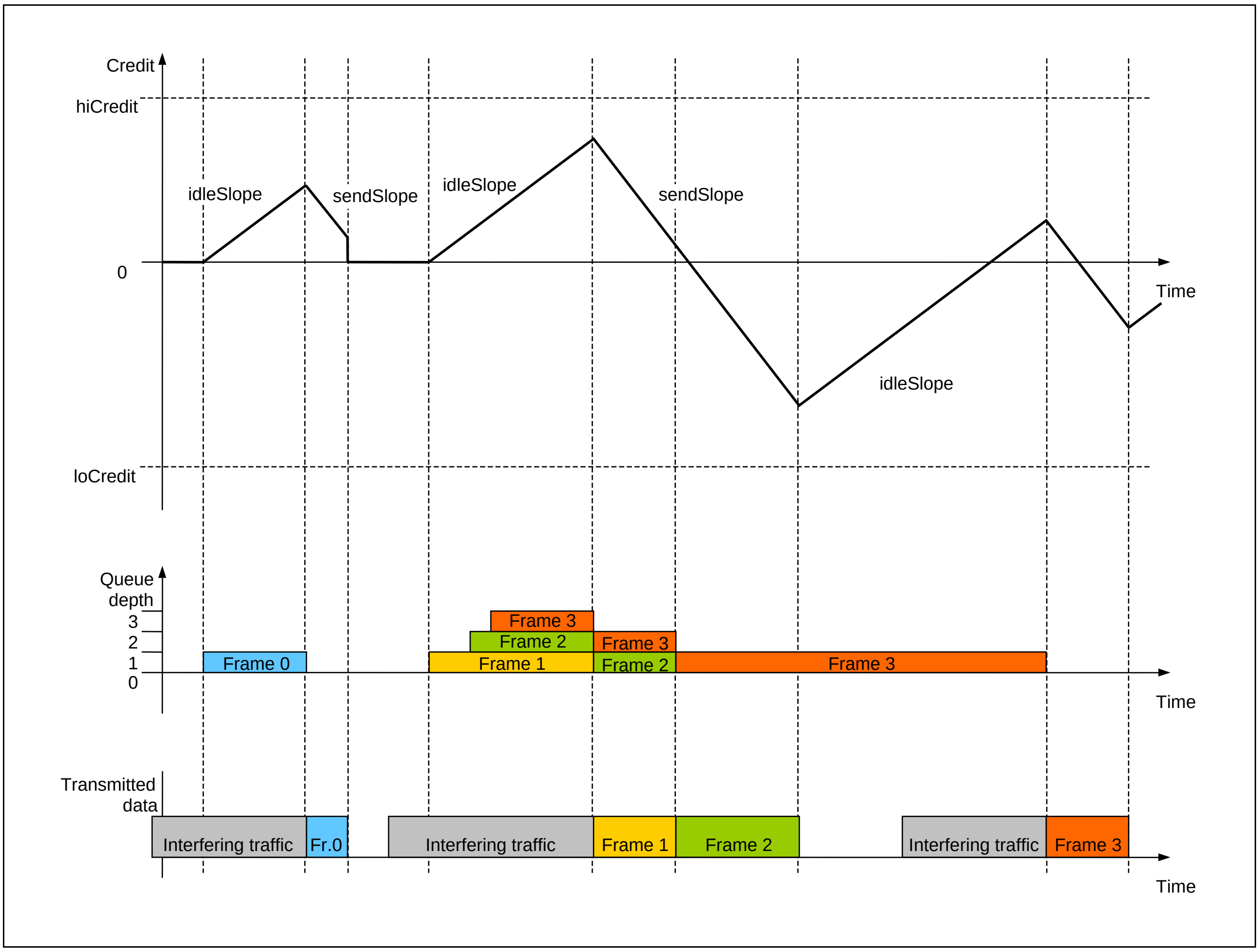


Figure 26.45 CBS (Credit-Based Shaping) Operation

Figure 26.46, CBS (Credit-Based Shaping) Operation in the AVB-DMAC shows the implementation of CBS in the AVB-DMAC.

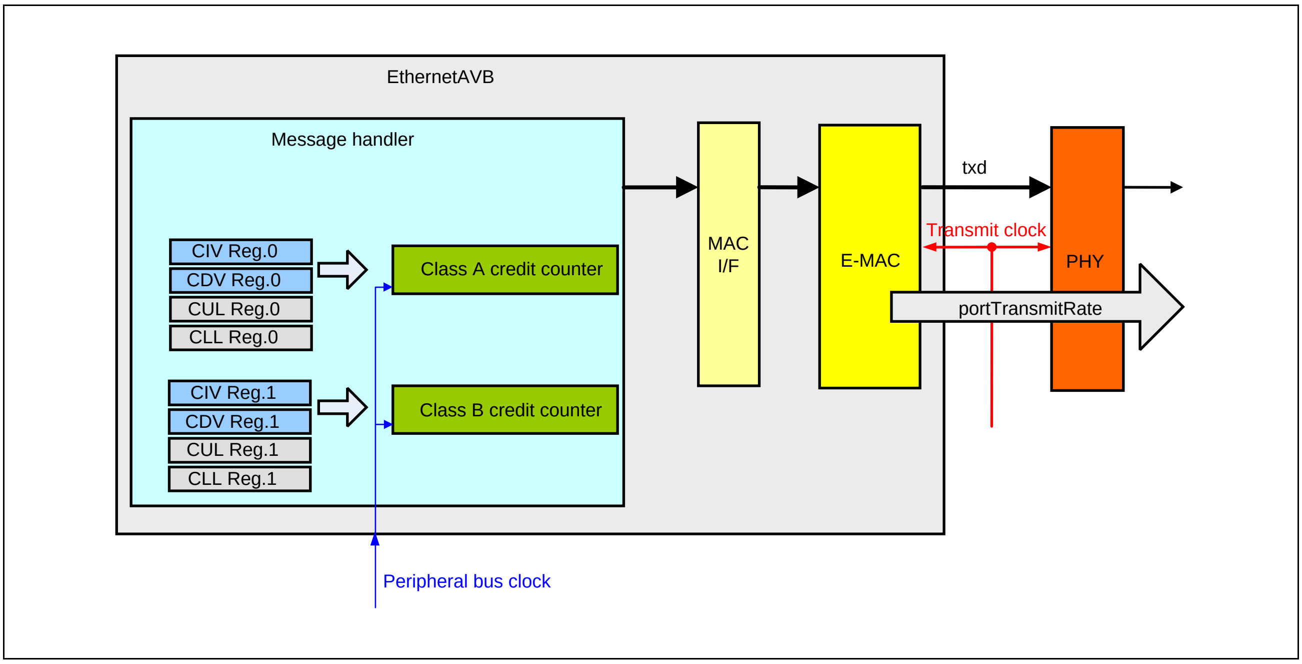


Figure 26.46 CBS (Credit-Based Shaping) Operation in the AVB-DMAC

The above implementation is based on “credit counters” for the respective traffic classes (SR Class A and Class B). The following parameters apply for these classes.

CBS increment value (CIV): Signed positive number
The credit is incremented by this amount every peripheral bus clock cycle while a frame from the queue is pending but transmission has not started (idleSlope).

CBS decrement value (CDV): Signed negative number
The credit is decremented by this amount every peripheral bus clock cycle while transmission of a frame from the queue is proceeding (sendSlope).

The CBS increment value (CIV) and CBS decrement value (CDV) are defined as follows.
CIV = idleSlope × Mfactor
CDV = sendSlope × Mfactor

Mfactor is a multiplier factor to ensure accuracy for CIV and CDV. CIV and CDV are calculated by using the following equations.
CIV = (portTransmitRate/CHI_freq) × bwFraction × Mfactor
CDV = (portTransmitRate/CHI_freq) × (bwFraction – 1) × Mfactor

CHI_freq is the frequency of the peripheral bus clock. The credit counters are driven by the peripheral bus clock, so calculating the slope parameters for CBS requires (1/CHI_freq).

Use software to prepare Mfactor for the CBS parameters. All queues for the same Class must have the same CBS parameters. Mfactor for a specified Class c can be changed during operation, unless transmission is pending for that Class (i.e. the transmit process status bit in the AVB-DMAC status register (ETNBnCSR.TPOt) = 0). At that time, the credit counter values for Class A and Class B are 0.

Note that the credit value will not match a new incrementation or decrementation parameter if Mfactor is changed while the credit counter value is non-zero. Mfactor is not present in the AVB-DMAC registers.

Set the CIV and CDV parameters in the CBS increment value registers c (ETNBnCIVRc) and the CBS decrement value registers c (ETNBnCDVRc). These are treated as dynamic settings since they should be updated when streams are registered or erased in accord with IEEE 802.1Qat.

The AVB-DMAC also has CBS upper limit registers c (ETNBnCULc) (the upper limit registers for Classes A and B) and CBS lower limit registers c (ETNBnCLLc) (the lower limit registers for Classes A and B). Set Mfactor to match the credit value and set the upper limit (hiCredit) and the lower limit (loCredit) for each class as defined above.

CUL = hiCredit × Mfactor = maxInterferenceSize × bwFraction × Mfactor
CLL = loCredit × Mfactor = maxFrameSize × (bwFraction - 1) × Mfactor

Example:

Assume that portTransmitRate = 100 Mbps, CHI_freq = 60 MHz and bwFraction = 3 %.
Then idleSlope and sendSlope represented as one bit vs. cycles of the peripheral bus clock are as follows.

idleSlope = (portTransmitRate/CHI_freq) × bwFraction = 100/60 (Mbps/MHz) × 3 % = 0.050 of a bit per peripheral bus clock cycle

sendSlope = idleSlope – (portTransmitRate / CHI_freq) = –1.616 bits per peripheral bus clock cycle Let Mfactor be 100, then CIV and CDV parameters are determined as follows.

CIV = idleSlope × Mfactor = 5.0
CDV = sendSlope × Mfactor = –161.6

26.4.6.1 Restrictions on CIV, CDV and Mfactor

The maximum value (the minimum value for negative numbers) up to which the credit counter will not overflow determines the maximum values of CIV and CDV that can be set in the CBS registers. This maximum credit value is equivalent to the worst case of the hiCredit value, and the maximum values for Class A and Class B are calculated as follows.

<Conditions>

- Class A maximum value (hiCredit_max_classA)
  classA bwFraction ≅ 100 %
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum frame size.
  hiCredit_max_classA ≅ maxInterferenceSize for Class A = Interference due to one max. sized frame = header + max. size payload + CRC (2000 bytes) + preamble (8 bytes) + IFG (12 bytes) + processing_delay (≅ 80 bytes) ≅ 2100 bytes
- Class B maximum value (hiCredit_max_classB)
  classB bwFraction ≅ 100 %
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum size of frames in the Class A transmission queue and other transmission queues.
  hiCredit_max_classB ≅ maxInterferenceSize for Class B = Interference due to two max-size frames = 2× hiCredit_max_classA ≅ 4200 bytes

hiCredit_max_classA = 16800
hiCredit_max_classB = 33600

The maximum values that can be selected with Mfactor for the 32-bit signed counter without overflow are:
Mfactor_max_classA = $2^{31} - 1$ / hiCredit_max_classA ≅ 127826 and
Mfactor_max_classB = $2^{31} - 1$ / hiCredit_max_classB ≅ 63913.

A high degree of accuracy can be achieved even with a low bandwidth. In Class B, bandwidthFraction = 0.05 % and the bandwidth error < 0.1 %.
The maximum value of CIV is calculated from the following equation.
CIV = idleSlope x Mfactor = (portTransmitRate / CHI_freq) × bandwidthFraction × Mfactor

When Mfactor is the maximum value and bandwidthFraction is the maximum value (up to 100 %):
CIV_max_classA = (portTransmitRate / CHI_freq) × Mfactor_max_classA and
CIV_max_classB = (portTransmitRate / CHI_freq) × Mfactor_max_classB.

Table 26.96, Example of Maximum Values for Class A and Class B CIV Parameters shows examples of values for portTransmitRate and peripheral clock frequency. Note that the values in the table are the limits of CIV up to which the 32-bit credit counter will not overflow. The CIV parameters are implemented as 16 bits + a sign bit, so a further limit of CIV ≤ 65535 applies to both Class A and Class B.

Table 26.96 Example of Maximum Values for Class A and Class B CIV Parameters

portTransmitRate	clk_chi[MHz]	CIV_max_classA	CIV_max_classB
100 Mbps	50	255652	127826
100 Mbps	60	213043	106521

26.4.6.2 Credit Incrementation During Inter-Frame Gaps (IFGs)

The inter-frame gap (IFG) after a frame is transmitted is not treated as part of frame transmission by the CSB credit counter. During an IFG, the credit is incremented for all SR queues that have pending frames or negative credit. **Figure 26.47, Credit Operations during IFGs** illustrates credit operations during IFGs.

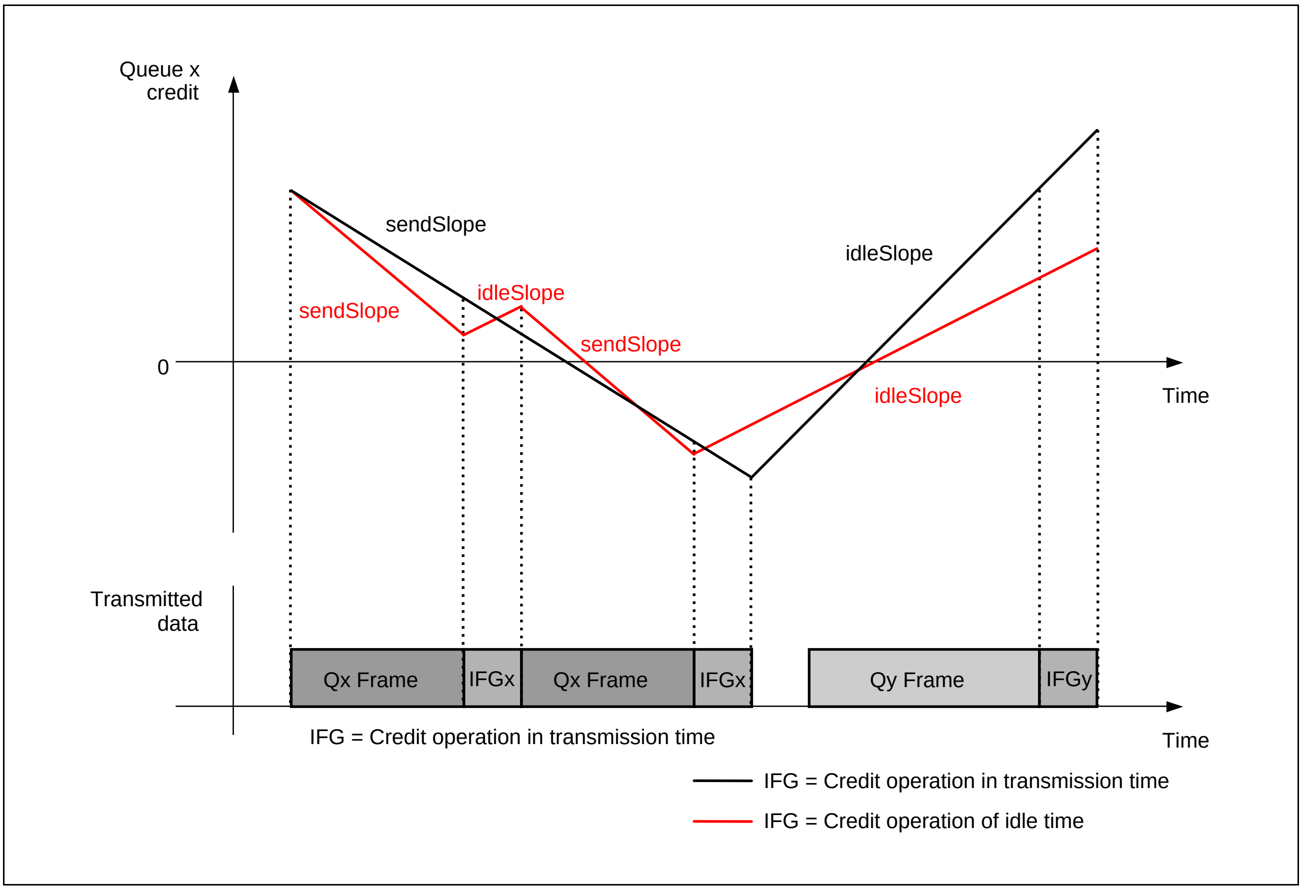


Figure 26.47 Credit Operations during IFGs

Accordingly, the IFG need not be included in calculation of the bandwidth requirements for the specified SR Class when deciding the idleSlope, sendSlope, and CIV and CDV parameters. However, IFG must also be included in the calculation in order to confirm that the total bandwidth allocated to all SR Classes does not exceed 100 % of portTransmitRate. This is described in *Section 35.2.2.8.4 of IEEE 802.1Q*.

26.4.6.3 Example

The case of a Class A 48-kHz stereo audio stream among Ethernet frames is described as an example.

After every Class A measurement interval (125 µs), 80 octets consisting of two sets of six 32-bit samples plus a 32-octet header are stored as audio data within a frame. The IEEE 802.3 also imposes a 42-octet media-specific framing overhead (an 8-octet preamble, 14-octet IEEE 802.3 header, 4-octet IEEE 802.1Q priority/VID Tag, 4-octet CRC, and 12-octet IFG) are also added. Accordingly, the total frame size is 80 + 42 = 122, and one such frame is transmitted after every class measurement interval.

This represents a total bandwidth of about 7.8 Mbits per second (122 octets × 8 bits per octet × 8000 frames per second) for this class. If the E-MAC is assumed to run at 100 Mbps (portTransmitRate), this is equivalent to the allocation of about 7.8 % of the total bandwidth to each Class A queue. If other traffic classes are to share the total transmission bandwidth, checking that this 7.8 % allocation does not lead to the total allocation of bandwidth being greater than 100 % of portTransmitRate is required.

To obtain the CIV and CDV parameters for a given class, the IFG must not be taken into account in calculation of the frame size must not include the IFG. For this case, therefore, we obtain an 80-bit payload + 30-bit overhead = 110-octet measurement interval for the class → the total bandwidth for the class = 7.04 Mbps = 7.04 % of portTransmitRate.

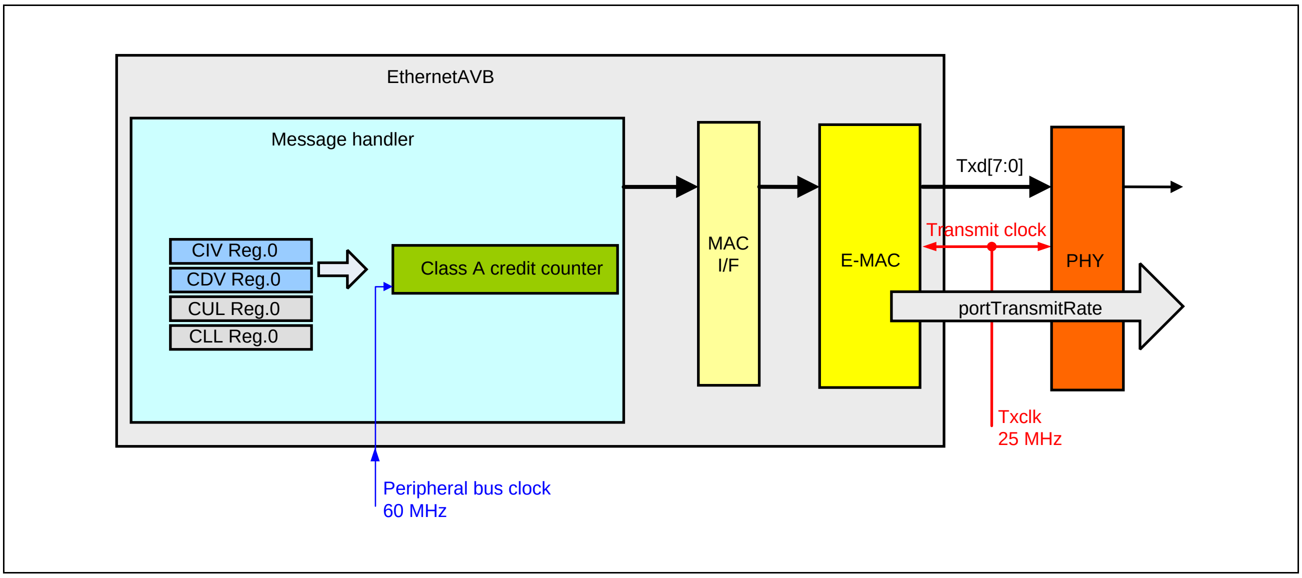


Figure 26.48 Example of CBS Settings

Given that:

- the E-MAC runs at 100 Mbps, so portTransmitRate = 100 Mbps and
- Peripheral bus clock (operating clock for the credit counter) frequency = 60 MHz,

securing a bandwidth of 7.04 Mbits/sec for Class A requires configuring the CBS parameters as follows.

- bandwidthFraction = 7.04 %
- idleSlope = (portTransmitRate / CHI_freq) × bandwidthFraction ≅ 0.1173 bits per peripheral bus clock cycle
- sendSlope = idleSlope – (portTransmitRate / CHI_freq) ≅ –1.5493 bits per peripheral bus clock cycle

When Mfactor = 100, the parameters are as follows.

- CIV = idleSlope × Mfactor = 11.7333 bits per peripheral bus clock cycle
- CDV = sendSlope × Mfactor = –154.9333 bits per peripheral bus clock cycle
  These are the final values for setting in the ETNBnCIVR1 and ETNBnCDVR1 registers.

26.4.7 IEEE802.1: gPTP

26.4.7.1 gPTP Timer

An 84-bit timer is provided to support the gPTP function. **Figure 26.49, Definitions of gPTP Timer Bits and Related Bits** shows the definitions of bits for the timer and in related registers. gPTP timer will start after ETNBnCCC.CSEL configure and transit to operation mode.

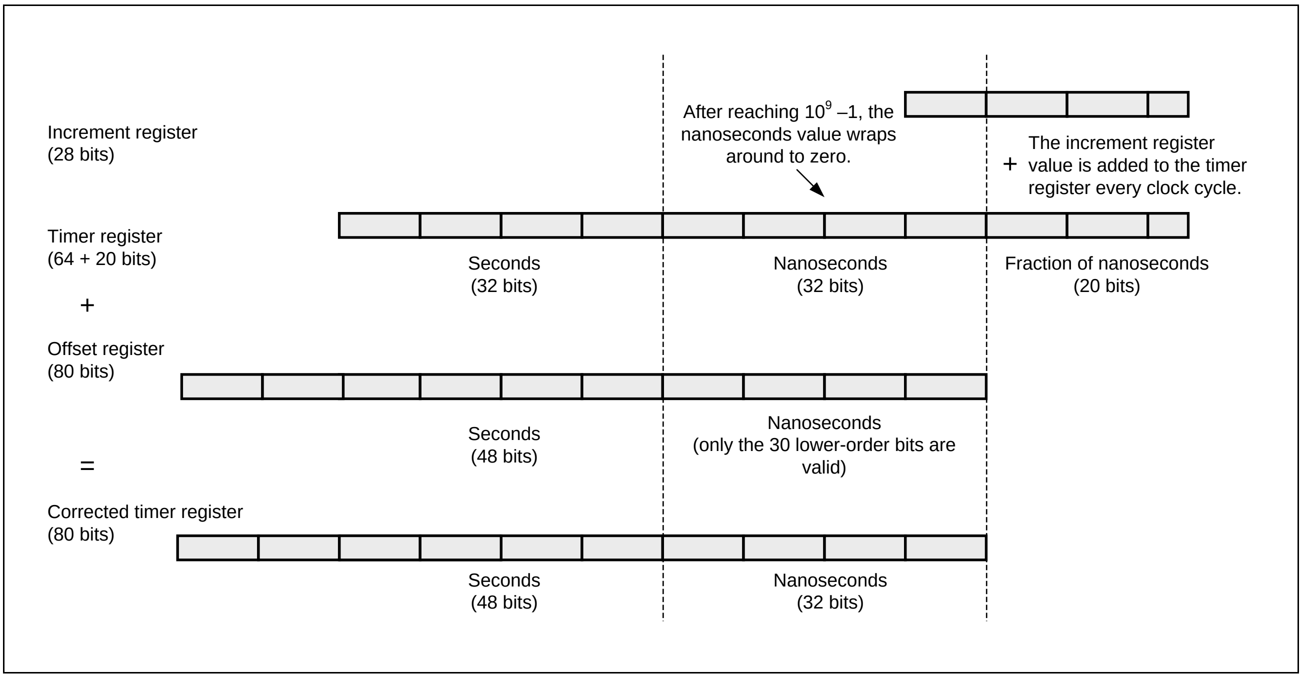


Figure 26.49 Definitions of gPTP Timer Bits and Related Bits

The higher-order 32 bits indicate seconds. For the next 32 bits, counting by one corresponds to the passage of 1 ns. The lower-order 20 bits are a fractional value (less than 1 ns). Software can only read the 32 higher-order bits, indicating seconds, and the subsequent 32-bits, indicating nanoseconds. The 20 lower-order 20 bits, representing less than 1 ns, are not readable. They are only used within the AVB-DMAC to maintain accuracy in time measurement.

The timer can be reset by setting the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 01_B. These bits are set to 00_B on completion of normal resetting of the timer.

After the timer starts, the value in the gPTP timer increment register (ETNBnGTI.TIV) is added to the value of the gPTP timer every clock cycle.

After setting a value in the gPTP timer increment register (ETNBnGTI.TIV), set the timer increment value setting request bit in the gPTP configuration control register (ETNBnGCCR.LTI). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed.

An offset to the gPTP timer is also available. If this is required, set the value in the gPTP timer offset register (ETNBnGTOi.TOV). After setting a value in this register, set the timer offset value setting request bit in the gPTP configuration control register (ETNBnGCCR.LTO). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed. When adding an offset, take care that it does not exceed 80 bits.

The value of the gPTP timer can be read from the gPTP timer capture register (ETNBnGCTi.CTV). Set the timer capture source select bits in the gPTP configuration control register (ETNBnGCCR.TCSS) to select the timer value for capture as the value of the gPTP timer, the corrected value of the gPTP timer (value with the offset added), or the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 11_B initiates the capture. Once normal capture of the timer is complete, the value of the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) returns to 00_B.

The timer for gPTP operates as a free-running timer but can be synchronized with the Grandmaster clock.

26.4.7.2 Free-Running Operation

The IEEE 802.1 AS standard for timing and synchronization does not prescribe the physical adjustment of local clocks to the Grandmaster clock. To avoid negative effects from the correction procedure, we recommend the use of a free-running timer.

As a free-running timer, the timer counts the local time in seconds or nanoseconds. The gPTP timer increment register (ETNBnGTI.TIV) is set to 1 ns (the setting value = $0010\ 0000_H$) and the gPTP timer offset register (ETNBnGTOi.TOV) is set to 0. The ratio information captured at the time of the gPTP delay measurement and synchronization procedures is used to correct the frequency ratio to that of the Grandmaster clock. The Grandmaster clock can be calculated from the local clock by using the information collected during the gPTP measurement and synchronization procedures.

26.4.7.3 Synchronization with the Grandmaster Clock

In situations requiring physical synchronization of the local clock with the Grandmaster clock, the fractional nanoseconds value (the 20 lower-order bits of the gPTP timer) is used to make the adjustment. Specifically, the increment value is finely adjusted to correct for deviations of the clock frequency from that of the Grandmaster clock.

Use the timer offset value (in the gPTP timer offset registers, ETNBnGTOi.TOV) to correct for offsets from the theoretical value (at start-up, etc.). The sum of the timer value and the offset register is the “corrected timer” value.

Note that only the nanoseconds portion of the gPTP timer offset registers (ETNBnGTOi.TOV[31:0]) is valid.

The following equation gives a method of calculating the increment (ETNBnGTI.TIV) from the frequency of the gPTP clock and its deviation from that of the Grandmaster clock. Variable d is the deviation ($d = 10^{-6}$ for 1 ppm).

$$ETNBnGTI.TIV = round\left(\frac{2^{20}GHz}{f_{GPTP}} \times (1 + d)\right)$$

After adjusting for the current deviation of clock frequency, re-set the gPTP timer increment register (ETNBnGTI.TIV).

After calculating the new offset value, re-set the gPTP timer offset register (ETNBnGTOi.TOV).

26.4.7.4 Support Provided by the gPTP Timer in Transmission and Reception

The timer value described above is used in the timestamp values captured when start frame delimiters are detected in reception and generated in transmission.

Captured timestamp values for received frames are stored in the corresponding descriptors. Those for transmitted frames are stored with tag information in the timestamp FIFO. The timestamp values are thus correlated with both transmitted and received frames.

Note that the use of corrected timer values can introduce an error due to the offset correction in the gPTP synchronization procedure.

Errors due to the SDF notification and the asynchronous interface between the timer modules must also be taken into account.

26.4.8 Support for IEEE 1722

For IEEE 1722, the following two functions are supported.

- Output and capture of values in the IEEE 1722 AVTP (Audio/Video Transport Protocol) presentation time format
- Comparison of IEEE 1722 AVTP presentation timestamps

The 32-bit AVTP timestamp field of IEEE 1722 frames holds the AVTP presentation time when the AVTP timestamp enable bit in the frame is 1. The AVTP timestamp field is generated from the pPTP timer and is given as seconds (gPTP_seconds) and nanoseconds (gPTP_nanoseconds) according to the following equation.

AVTP timestamp = (gPTP_seconds × 10^9 + gPTP_nanoseconds) modulo 2^{32}

The AVTP presentation time can be read from the gPTP timer capture register (ETNBnGCTi.CTV). Set the timer capture source select bits in the gPTP configuration control register (ETNBnGCCR.TCSS) to select the timer value for capture as the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 11_B initiates the capture. The value is obtained by adding the maximum transit time defined in the gPTP maximum transit time register (ETNBnGMTT.MTTV) to the corrected timer value. The AVTP presentation time wraps around approximately every four seconds.

CAUTION

The AVTP presentation time captured in ETNBnGCTi.CTV is only valid when the corrected timer value is in synchronization with the Grandmaster clock. That is, the timer increment and timer offset values for the corrected timer value must be adjusted during the synchronization procedure so that the corrected gPTP clock is physically adjusted to match the time kept by the Grandmaster clock.

26.4.9 Flow Control

The E-MAC supports flow control for full-duplex operation in compliance with the IEEE 802.3 standards. This flow control is applicable to both reception and transmission. In regard to the transmission of PAUSE frames, flow control operates in the following ways.

(1) PAUSE Frame Transmission
PAUSE frames can also be transmitted in response to software operations. Writing a timer value to the manual PAUSE frame register (ETNBnMPR) starts the transmission of a PAUSE frame. This only causes the transmission of one PAUSE frame.

(2) PAUSE Frame Reception
After reception of a PAUSE frame, transmission of the next frame does not proceed until the time indicated by the Timer value elapses. However, transmission of a frame currently being transmitted continues. PAUSE frames are only received while the operation mode for flow control in reception in the E-MAC mode register (ETNBnECMR.RXF) is set to "1". The number of received PAUSE frames is counted.

(3) PAUSE Frames with the Timer Value 0
The setting of the 0-time PAUSE frame enable bit (ETNBnECMR.ZPF) enables or disables the reception and transmission of PAUSE frames with the TIME parameter value 0.

- When control of PAUSE frames with the TIME parameter value 0 is enabled
  A PAUSE frame with the TIME parameter value 0 is transmitted when the capacity of the reception FIFO is less than the value of the Receive FIFO Warning Level bits (ETNBnRCR.RFCL[12:0]) while the time indicated by the TIME parameter value has not elapsed.
  Reception of a PAUSE frame with the TIME parameter value 0 leads to release from the transmission standby state.
- When control of PAUSE frames with the TIME parameter value 0 is disabled
  PAUSE frames with the TIME parameter value 0 are not transmitted. Received PAUSE frames with the TIME parameter value 0 are discarded.

26.4.10 Interrupts

The EthernetAVB module has three EI level interrupts from the AVB-DMAC and one EI level interrupt from the E-MAC.

Table 26.97, EthernetAVB Interrupts lists interrupts.

Table 26.97 EthernetAVB Interrupts

Interrupt Source Name
Transmit/receive data management interrupt
Error management interrupt
Other management (FIFO caution level, etc.) interrupt
E-MAC interrupt

The AVB-DMAC related interrupts include descriptor interrupts (15 sources), error interrupts (5 sources), reception interrupts (38 sources), transmission interrupts (2 sources), and gPTP interrupts (3 sources). From the CPU's perspective, each appears as three of the above four interrupt sources.

The states of an AVB-DMAC-related interrupt sources can be checked in the following registers.

- Descriptor interrupt status register (ETNBnDIS)
- Error interrupt status register (ETNBnEIS)
- Receive interrupt status register (ETNBnRIS0 to 2)
- Transmit interrupt status register (ETNBnTIS)
- gPTP interrupt status register (ETNBnGIS)

The interrupts are controlled by the corresponding interrupt enable bits. However, the status flags operate independently of the settings of the enable bits.

The states of grouped interrupts can only be checked by reading the interrupt summary status register (ETNBnISS) and the queue full error interrupt status bit in the error interrupt status register (ETNBnEIS.QFS). This reduces the load on the CPU.

26.4.10.1 Transmit/Receive Data Management Interrupt

The management interrupt for transmission and reception is conveyed when the following interrupt sources are generated:

- Receive frame interrupt in the receive interrupt status register 0 (ETNBnRIS0.FRFr)
- Descriptor interrupt in the descriptor interrupt status register (ETNBnDIS.DPFi)

The general error interrupt state can be checked by reading the descriptor interrupt summary bits in the interrupt summary status register (ETNBnISS.DPSi) or the receive FIFO warning interrupt summary bit (ETNBnISS.RFWS).

26.4.10.2 Error Management Interrupt

The error management interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

- Timestamp FIFO full error interrupt in the error interrupt status register (ETNBnEIS.TFFF)
- CBS limitation interrupts in the error interrupt status register (ETNBnEIS.CULF1, ETNBnEIS.CULF0, ETNBnEIS.CLLF1, ETNBnEIS.CLLF0)
- Separation Error in the error interrupt status register (ETNBnEIS.SEF)
- Queue Error in the error interrupt status register (ETNBnEIS.QEF)
- MAC Transmission Error in the error interrupt status register (ETNBnEIS.MTEF)
- MAC Reception Error in the error interrupt status register (ETNBnEIS.MREF)
- Receive FIFO full interrupt in the receive interrupt status register 2 (ETNBnRIS2.RFFF)
- Receive queue full interrupt in the receive interrupt status register 2 (ETNBnRIS2.QFFr)

The general error interrupt state can be checked by reading the error interrupt summary bit in the interrupt summary status register (ISS.ES).

26.4.10.3 Other Management (FIFO Warning, etc.) Interrupts

The other management (FIFO warning, etc.) interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

(1) Reception related interrupt
Receive FIFO warning interrupt in the receive interrupt status register 1 (ETNBnRIS1.RFWF)

(2) Transmission related interrupts
Timestamp FIFO warning interrupt in the transmit interrupt status register (ETNBnTIS.TFWF) Timestamp FIFO update interrupt in the transmit interrupt status register (ETNBnTIS.TFUF)

(3) gPTP related interrupts
AVTP presentation time match interrupt in the gPTP interrupt status register (ETNBnGIS.PTMF)

The general error interrupt state can be checked by reading the receive FIFO warning error interrupt status bit in the interrupt summary status register (ETNBnISS.RFWS), the timestamp FIFO warning interrupt status bit (ETNBnISS.TFWS), the timestamp FIFO update interrupt status bit (ETNBnISS.TFUS), and the gPTP Interrupt Summary (ETNBnISS.CGIS).

26.4.10.4 E-MAC Interrupt

The E-MAC interrupt is conveyed when the E-MAC interrupt source is generated.

The general error interrupt state can be checked by reading the E-MAC interrupt summary bit in the interrupt summary status register (ETNBnISS.MS).

26.4.11 Flows of Operations

26.4.11.1 Flow of E-MAC Initialization

Figure 26.50, Flow of E-MAC Initialization (for AVB Mode and Full-Duplex Operation) shows the flow of E-MAC initialization (for AVB mode and full-duplex operation).

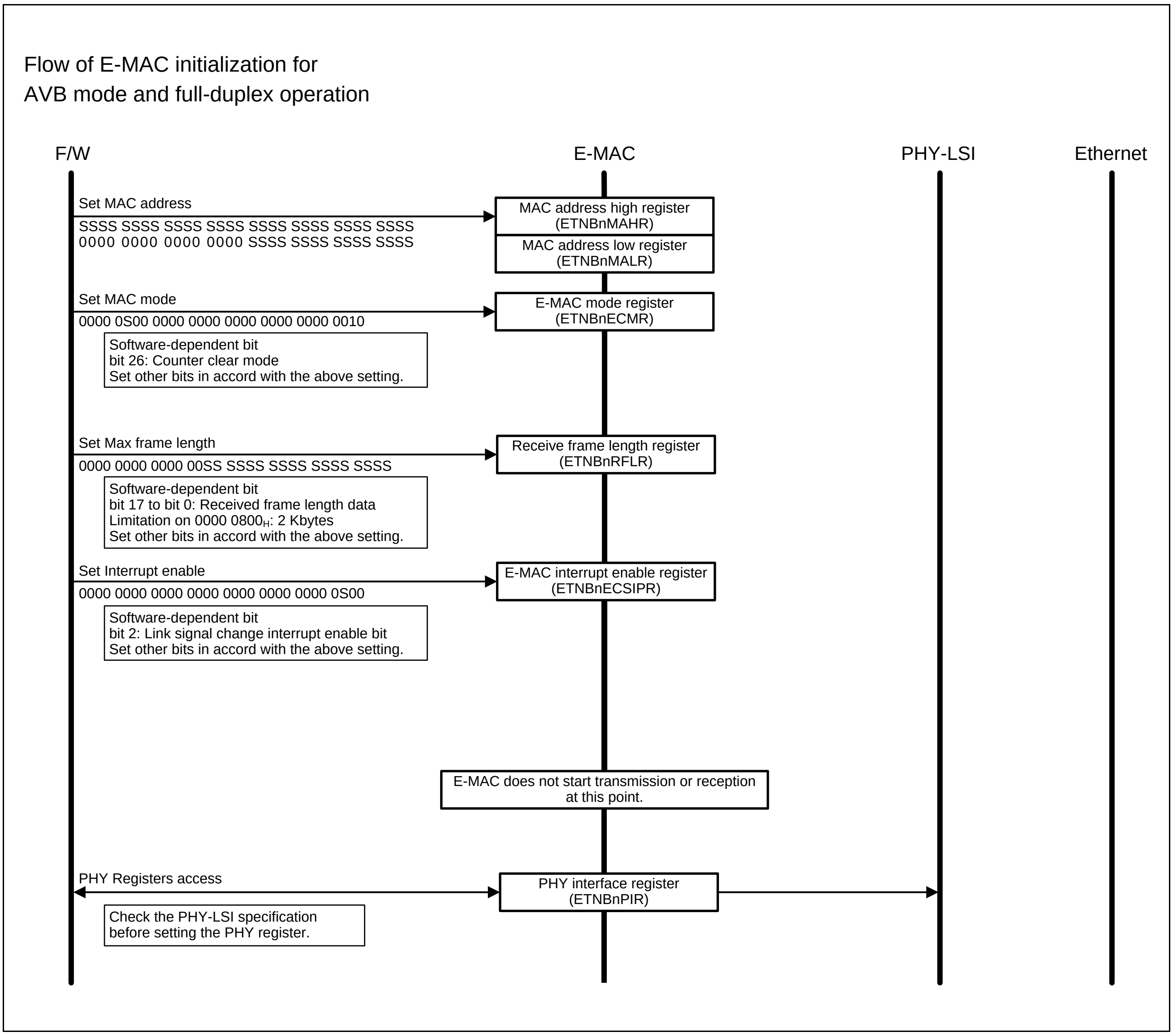


Figure 26.50 Flow of E-MAC Initialization (for AVB Mode and Full-Duplex Operation)

26.4.11.2 Flow of AVB-DMAC Initialization

Figure 26.51, Flow of AVB-DMAC Initialization (for AVB Mode and Full-Duplex Operation) shows the flow of AVB-DMAC initialization (for AVB mode and full-duplex operation).

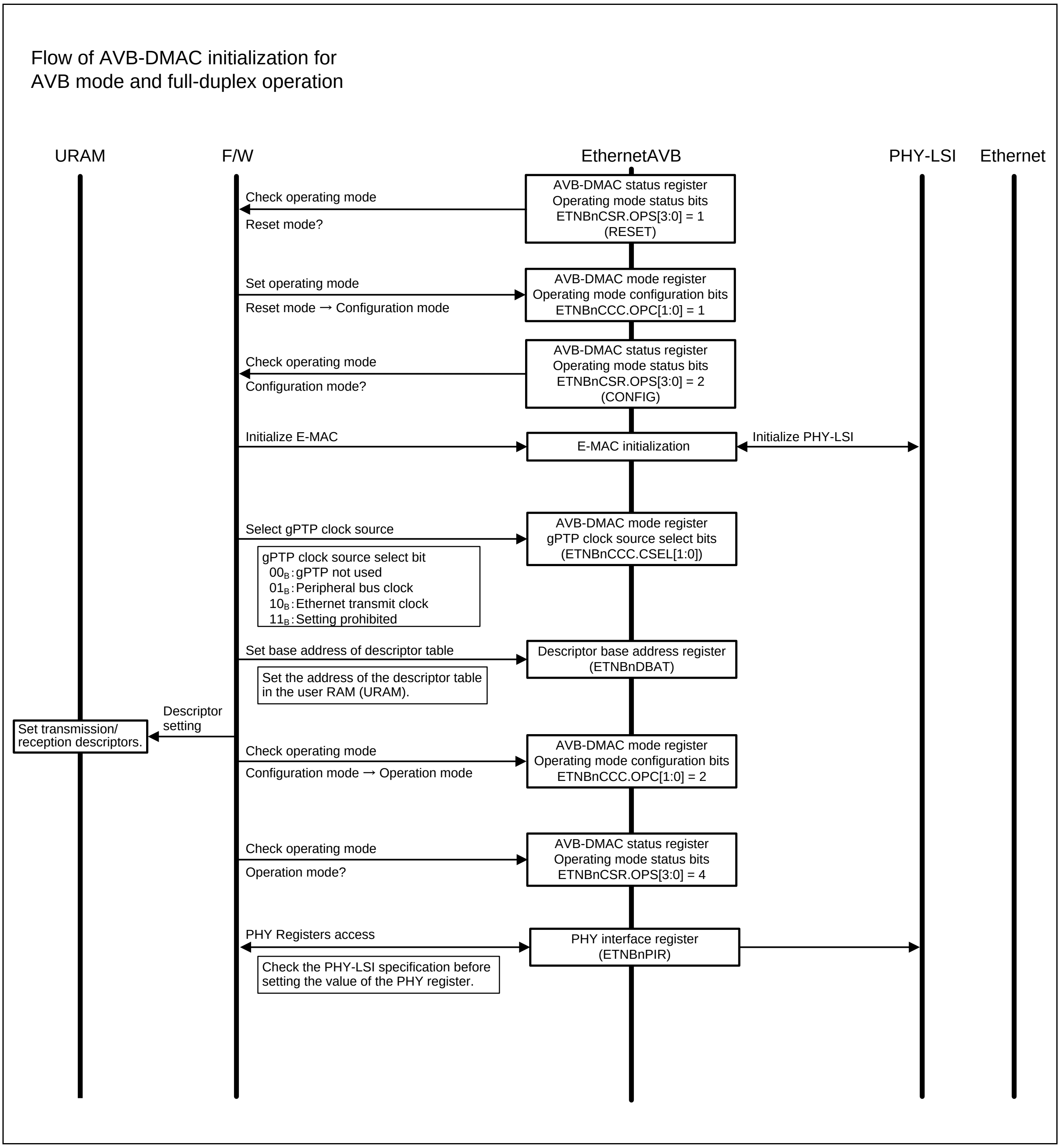


Figure 26.51 Flow of AVB-DMAC Initialization (for AVB Mode and Full-Duplex Operation)

26.4.11.3 Flow for the AVB-DMAC in Reception

Figure 26.52, Flow for the AVB-DMAC in Reception (in AVB Mode and Full-Duplex Operation) shows the flow for the AVB-DMAC in reception (in AVB mode and full-duplex operation).

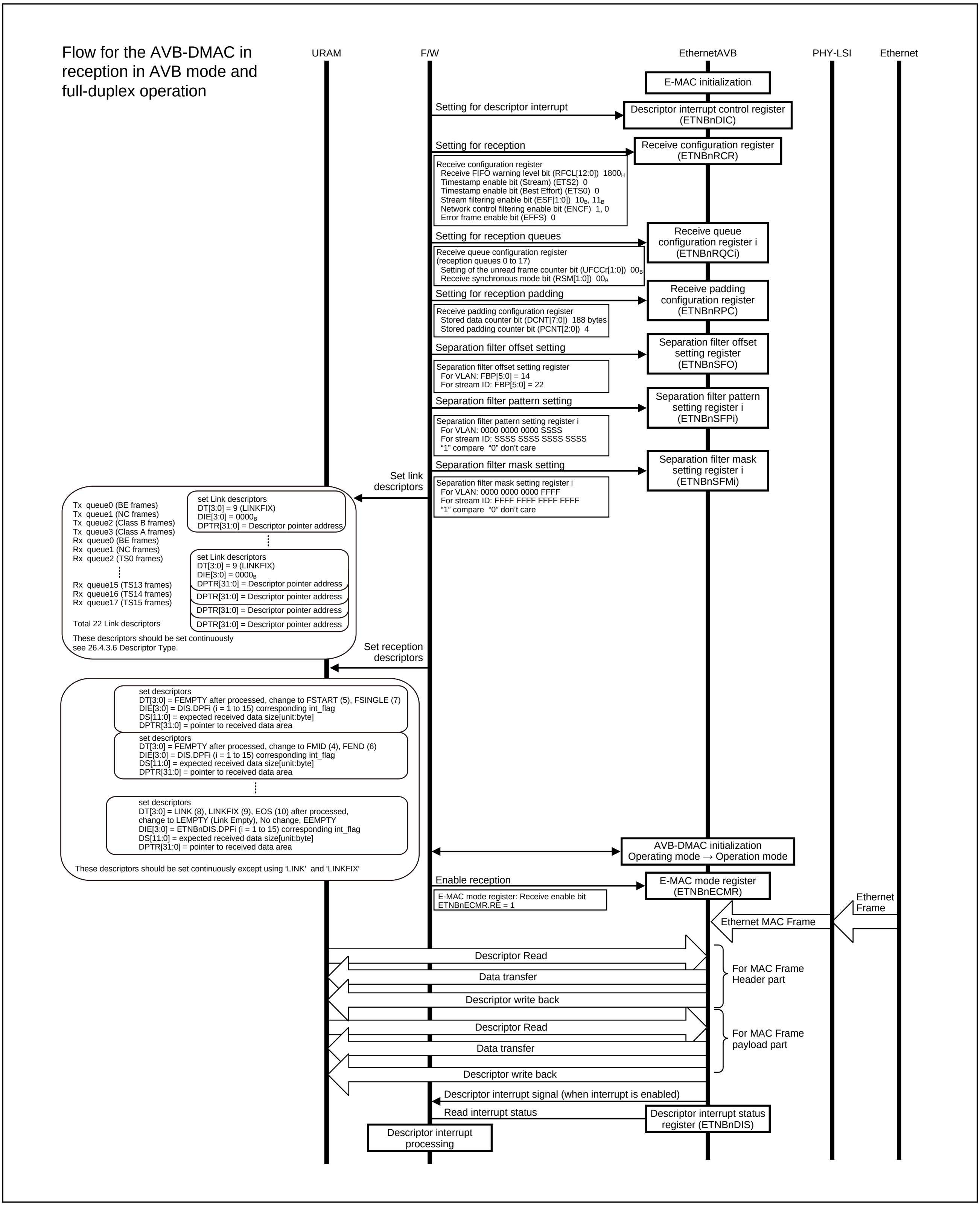


Figure 26.52 Flow for the AVB-DMAC in Reception (in AVB Mode and Full-Duplex Operation)

26.4.11.4 Flow for the AVB-DMAC in Transmission

Figure 26.53, Flow for the AVB-DMAC in Transmission (in AVB Mode and Full-Duplex Operation) shows the flow for the AVB-DMAC in transmission (in AVB mode and full-duplex operation).

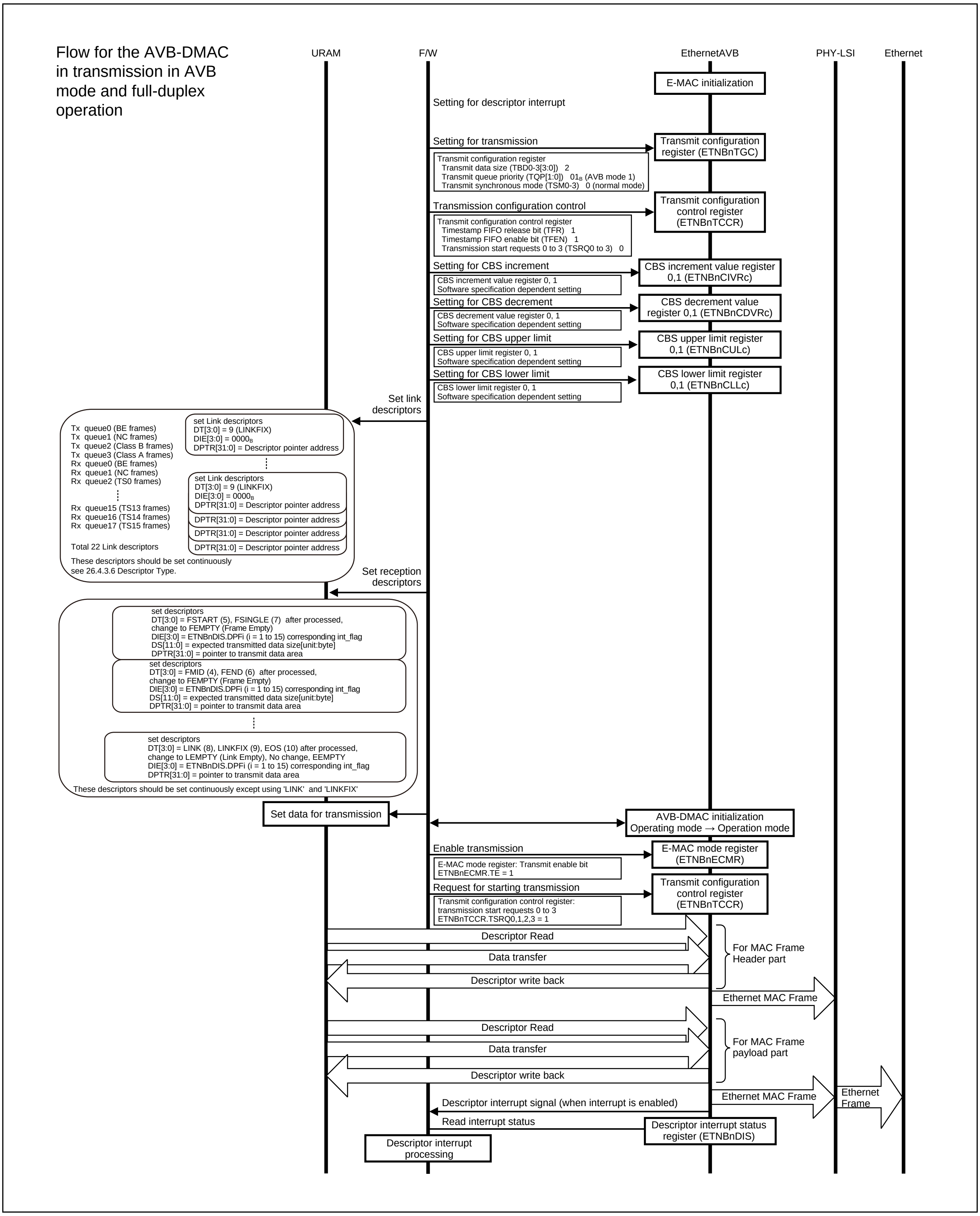


Figure 26.53 Flow for the AVB-DMAC in Transmission (in AVB Mode and Full-Duplex Operation)

26.4.11.5 Flow for Stopping AVB-DMAC Operation in Reception

Figure 26.54, Flow for Stopping AVB-DMAC Operation in Reception (Normal, Common to All Modes) shows the flow for stopping AVB-DMAC operation in reception (normal, common to all modes).

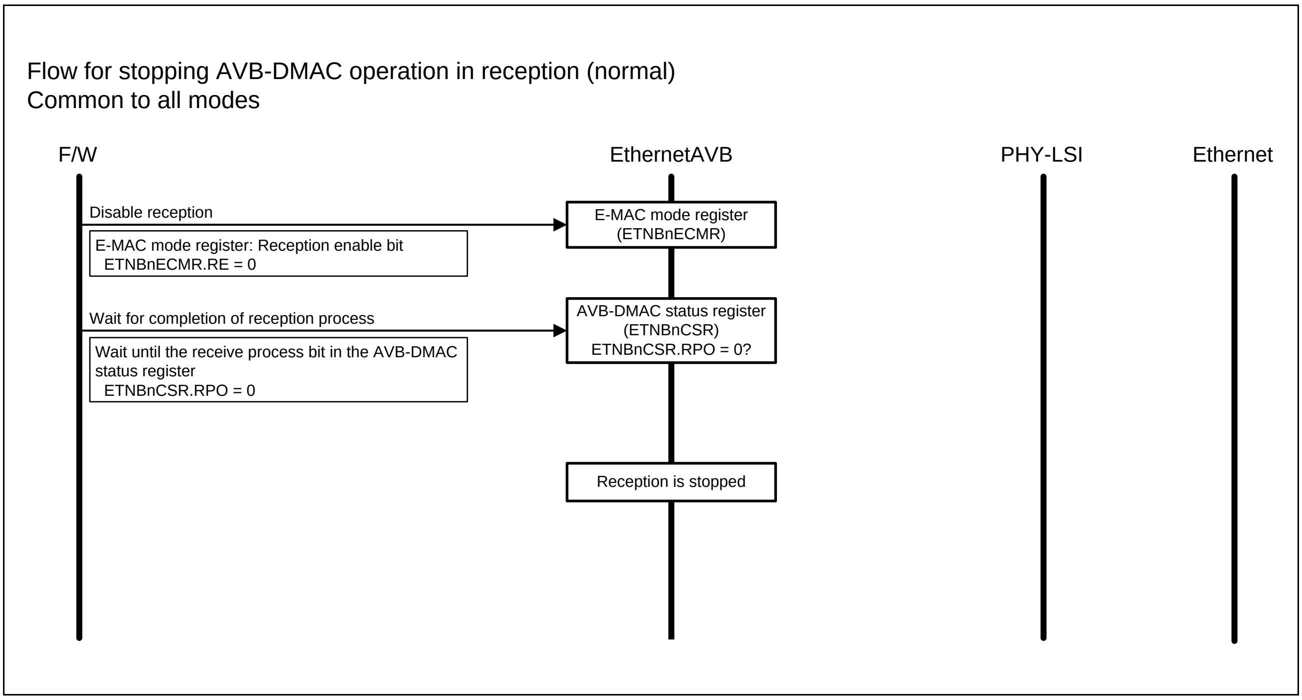


Figure 26.54 Flow for Stopping AVB-DMAC Operation in Reception (Normal, Common to All Modes)

26.4.11.6 Flow for Stopping AVB-DMAC Operation in Transmission

Figure 26.55, Flow for Stopping AVB-DMAC Operation in Transmission (Normal, Common to All Modes) shows the flow for stopping AVB-DMAC operation in transmission (normal, common to all modes).

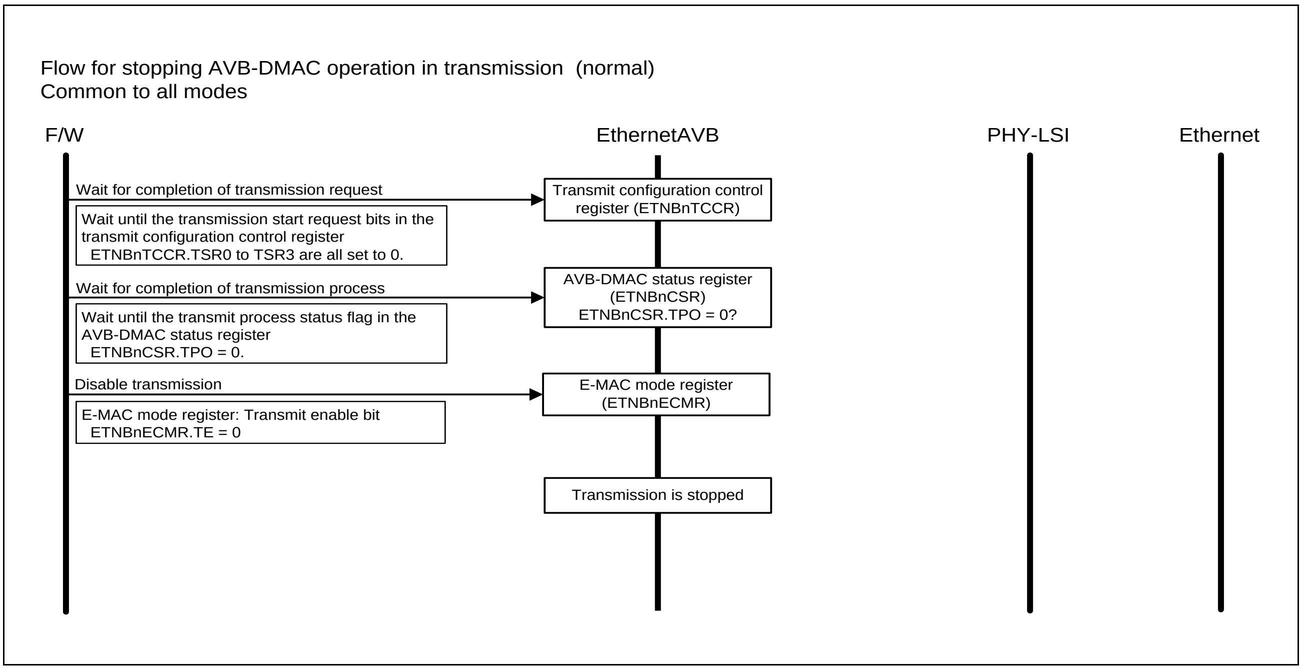


Figure 26.55 Flow for Stopping AVB-DMAC Operation in Transmission (Normal, Common to All Modes)

26.4.11.7 Flow for Stopping and Resetting the AVB-DMAC

Figure 26.56, Flow for Stopping and Resetting the AVB-DMAC (Normal, Common to All Modes) shows the flow for stopping and resetting the AVB-DMAC (normal, common to all modes).

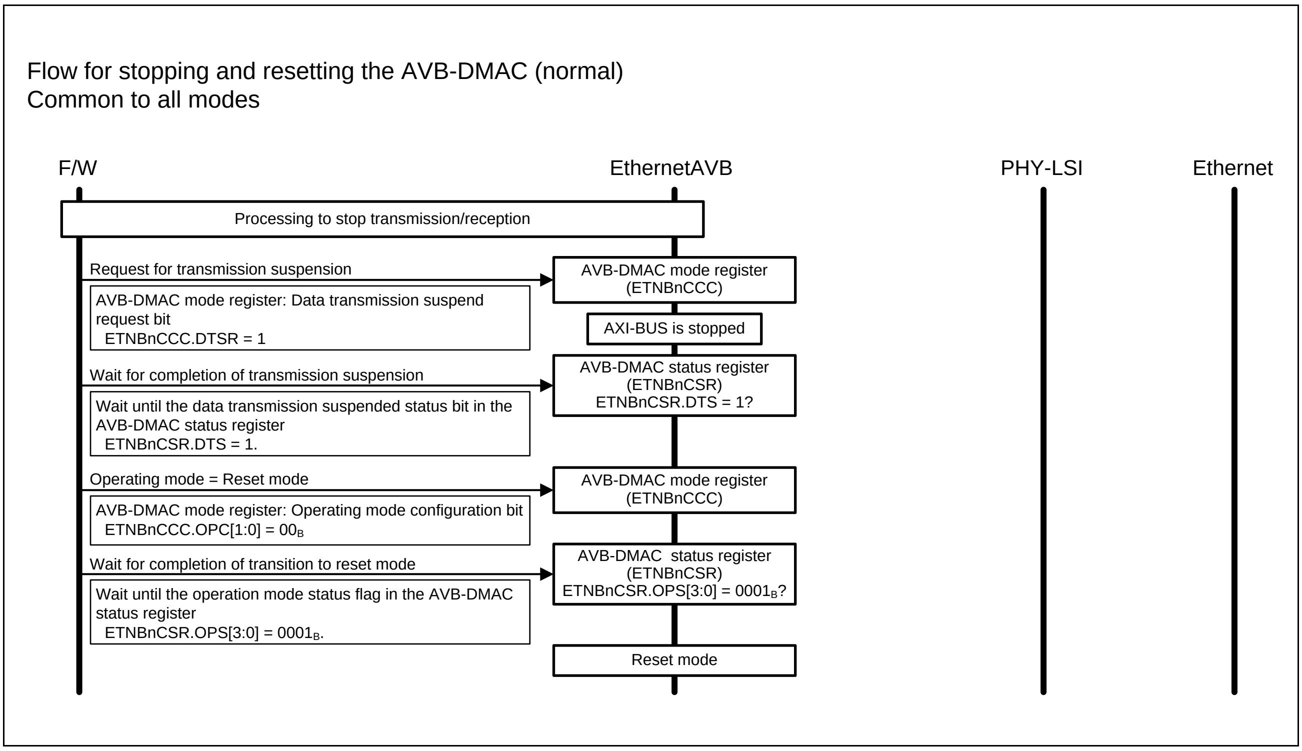


Figure 26.56 Flow for Stopping and Resetting the AVB-DMAC (Normal, Common to All Modes)

26.4.11.8 Flow for Emergency Stopping the AVB-DMAC

Figure 26.57, Flow for Emergency Stopping the AVB-DMAC (Normal, Common to All Modes) shows the flow for emergency stopping the AVB-DMAC (normal, common to all modes).

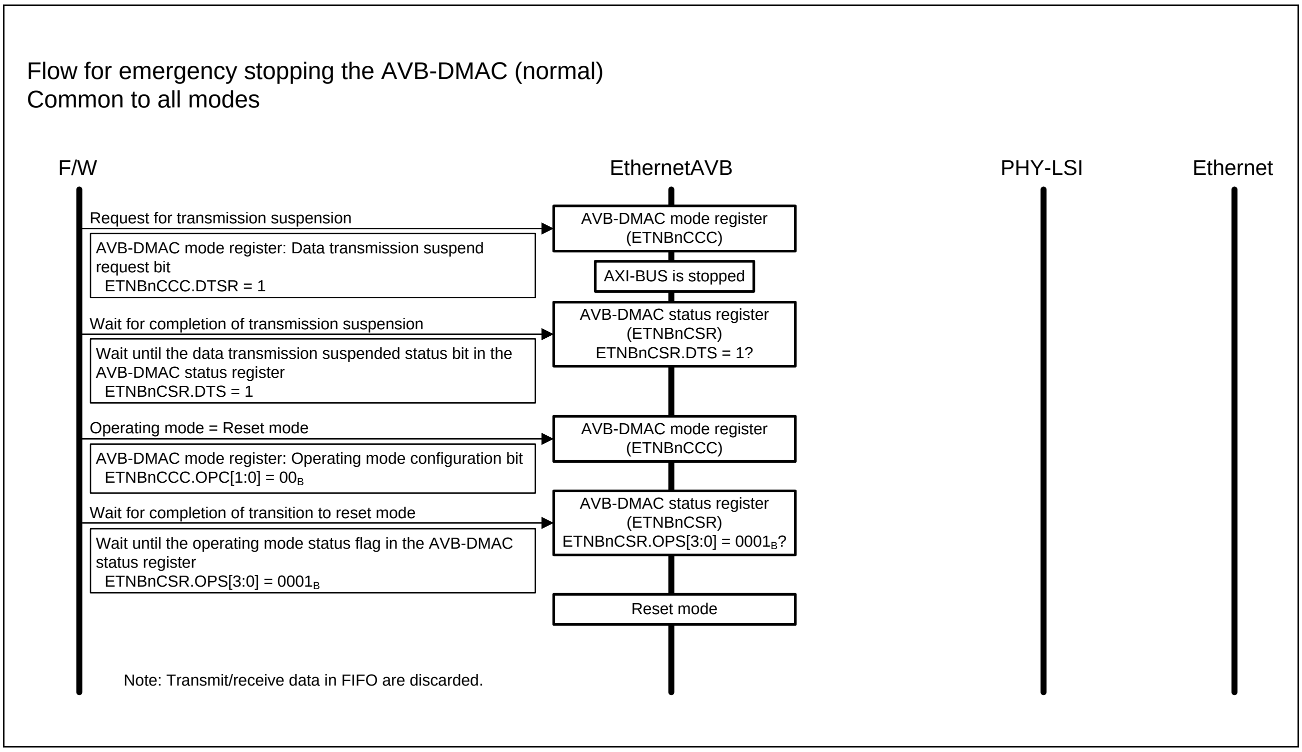


Figure 26.57 Flow for Emergency Stopping the AVB-DMAC (Normal, Common to All Modes)

26.4.11.9 Flow of gPTP Initialization

Figure 26.58, Flow of gPTP Initialization (Normal, Common to All Modes) shows the flow of gPTP initialization (normal, common to all modes).

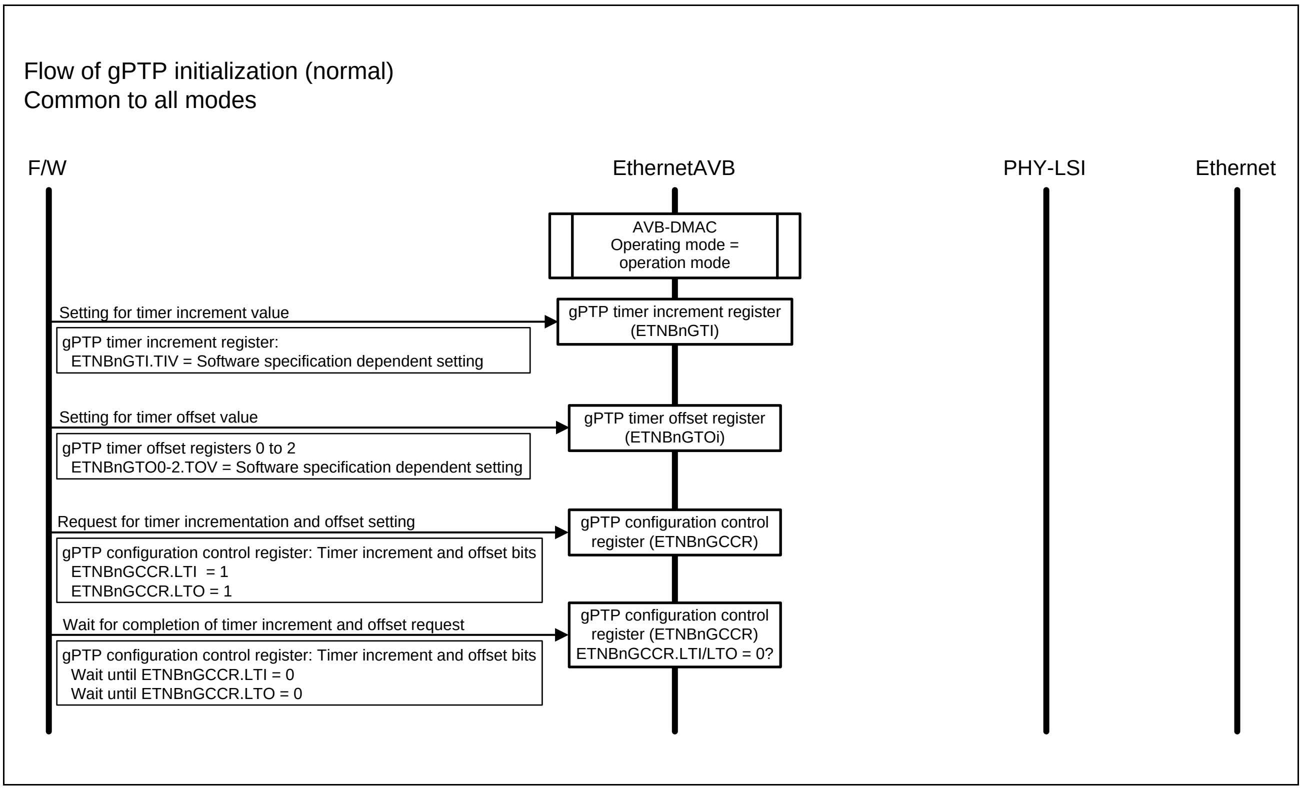


Figure 26.58 Flow of gPTP Initialization (Normal, Common to All Modes)

26.4.11.10 Flow of gPTP Timestamping in Transmission

Figure 26.59, Flow of gPTP Timestamping in Transmission (Normal, Common to All Modes) shows the flow of gPTP timestamping in transmission (normal, common to all modes).

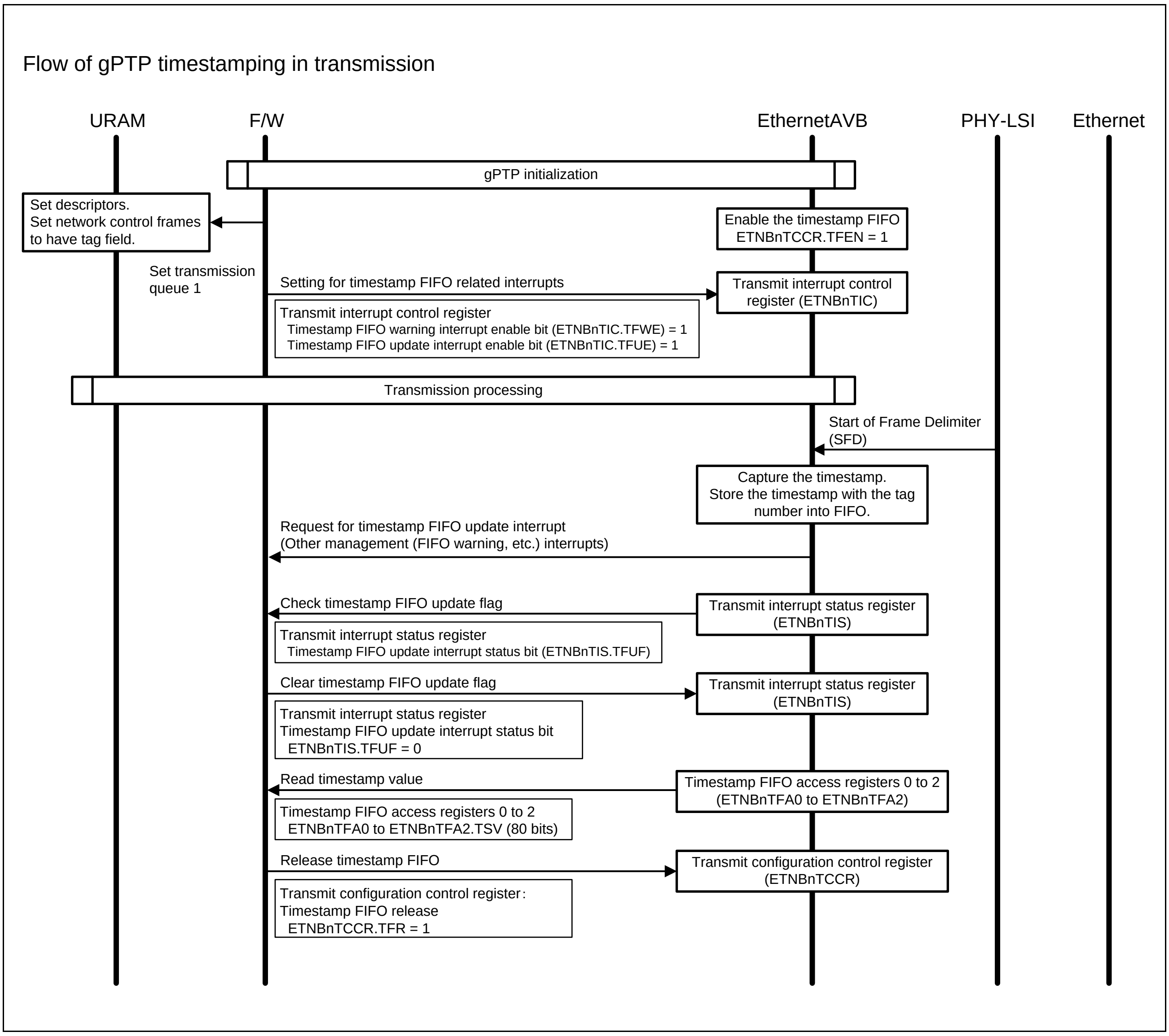


Figure 26.59 Flow of gPTP Timestamping in Transmission (Normal, Common to All Modes)

26.4.11.11 Flow of gPTP Timestamping and Synchronization in Reception

Figure 26.60, Flow of gPTP Timestamping and Synchronization in Reception (Normal, Common to All Modes) shows the flow of gPTP timestamping and synchronization in reception (normal, common to all modes).

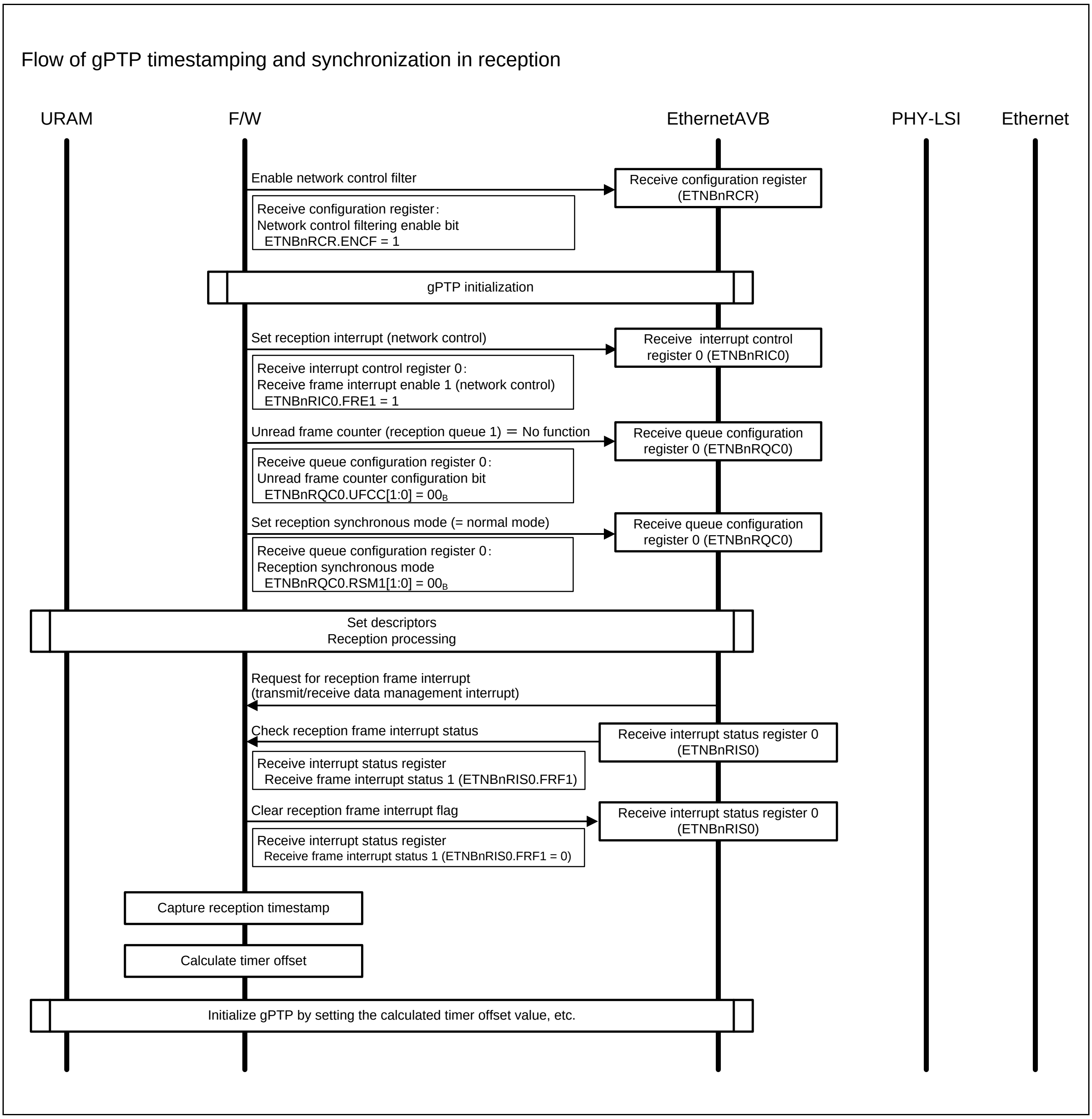


Figure 26.60 Flow of gPTP Timestamping and Synchronization in Reception (Normal, Common to All Modes)

26.4.11.12 Flow of Capturing gPTP Presentation Times

Figure 26.61, Flow of Capturing gPTP Presentation Times (Common to All Modes) shows the flow of capturing gPTP presentation times (common to all modes).

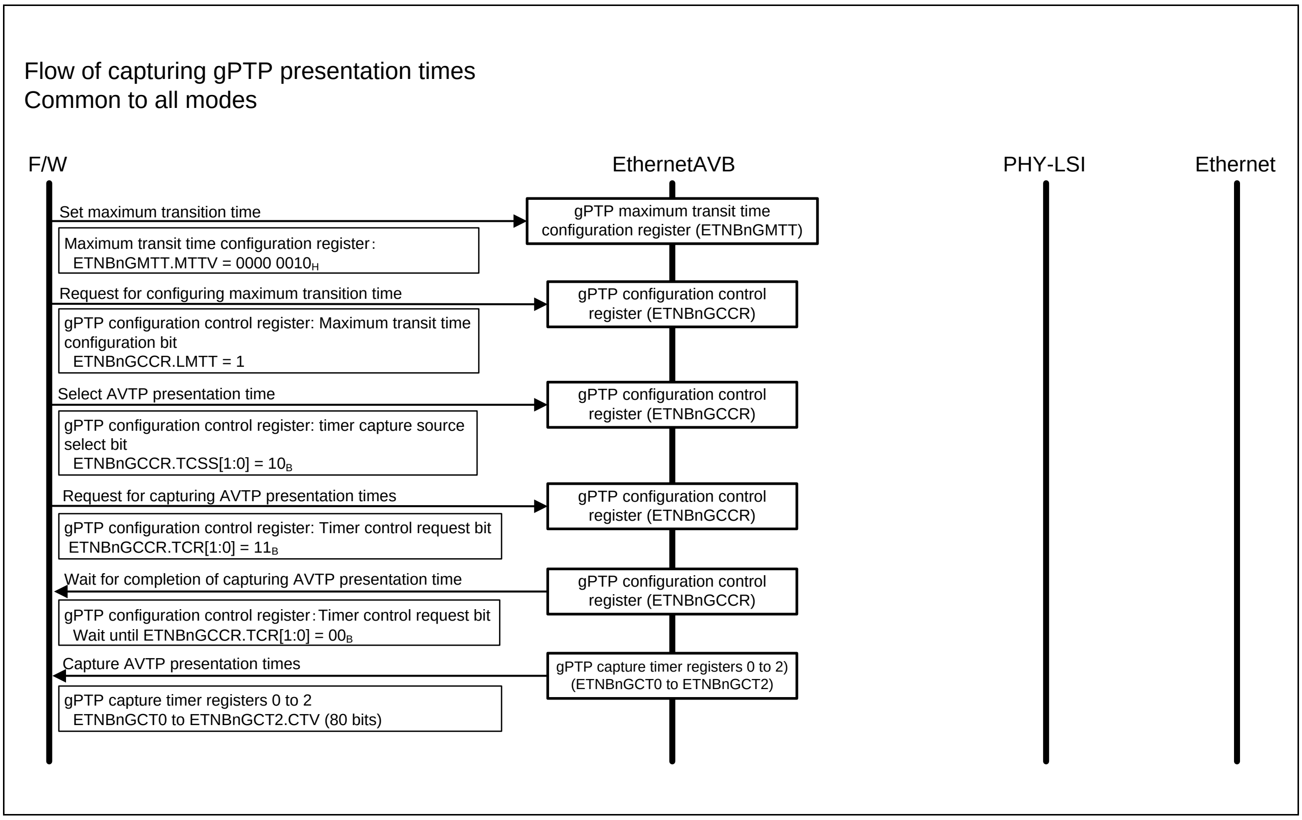


Figure 26.61 Flow of Capturing gPTP Presentation Times (Common to All Modes)

26.4.11.13 Flow of AVTP Presentation Time Comparison

Figure 26.62, Flow of AVTP Presentation Time Comparison (Common to All Modes) shows the flow of AVTP presentation time comparison (common to all modes).

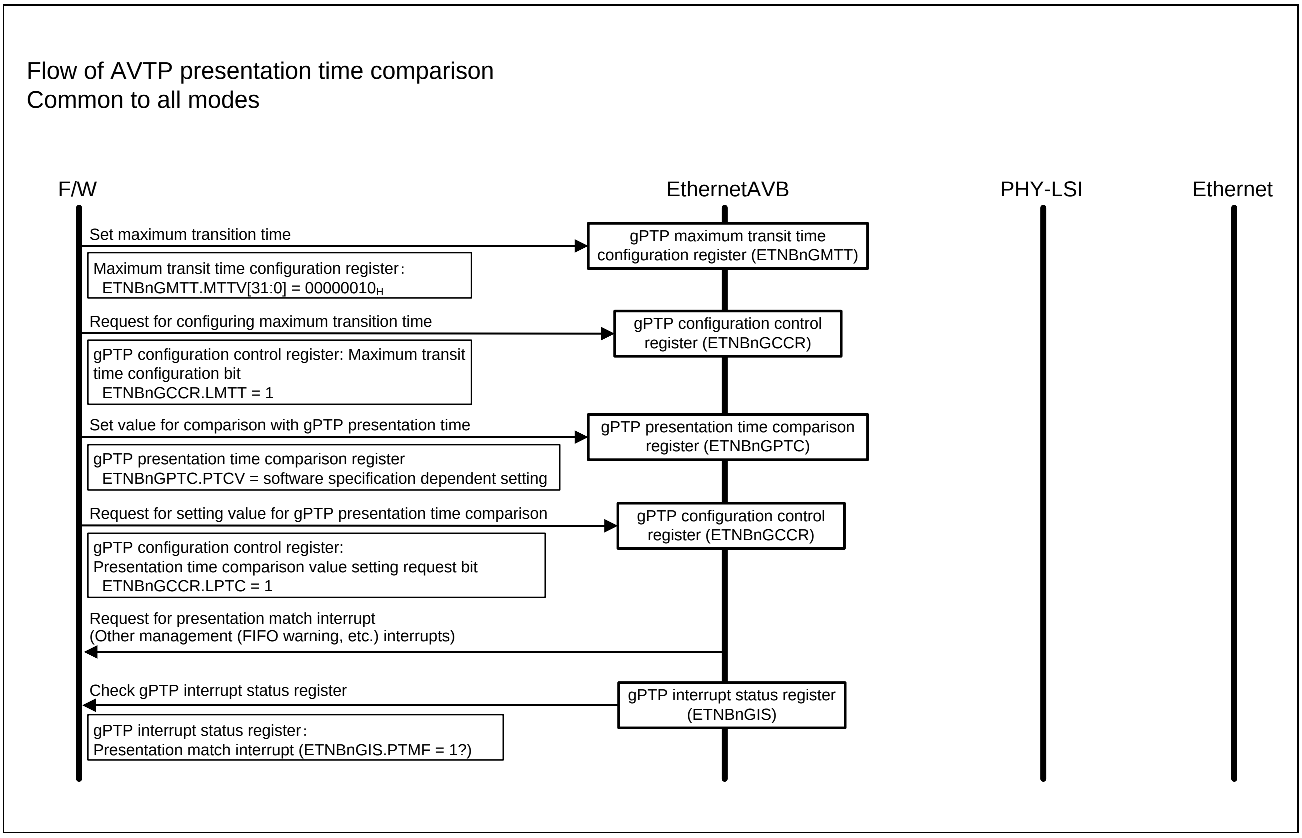


Figure 26.62 Flow of AVTP Presentation Time Comparison (Common to All Modes)

26.4.11.14 Flow of Loopback Mode Operation

Figure 26.63, Flow of Loopback Mode Operation shows the flow of loopback mode operation.

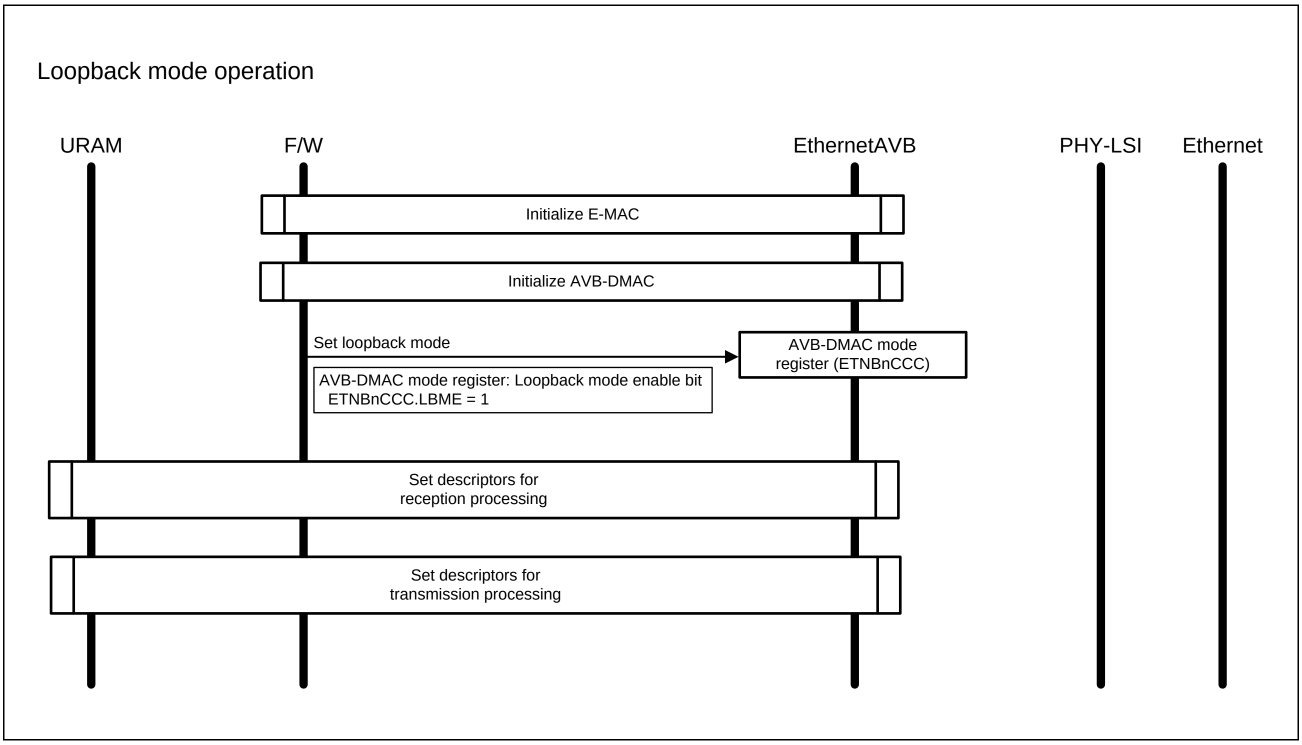


Figure 26.63 Flow of Loopback Mode Operation

26.4.12 Connection to PHY-LSI

26.4.12.1 MII Frame Transmission/Reception Timing

Each MII frame transmission/reception timing is shown in **Figure 26.64, MII Frame Transmit Timing (Normal Transmission)** to **Figure 26.68, MII Fame Receive Timing (Reception Error (2))**.

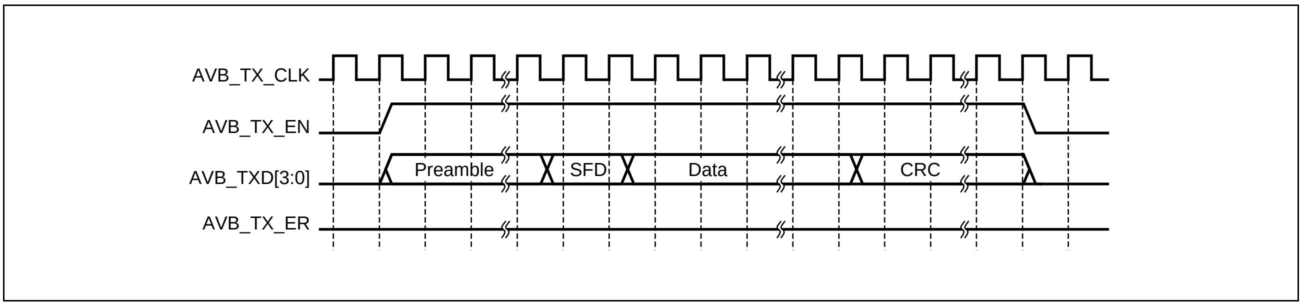


Figure 26.64 MII Frame Transmit Timing (Normal Transmission)

Figure 26.65 Reserved

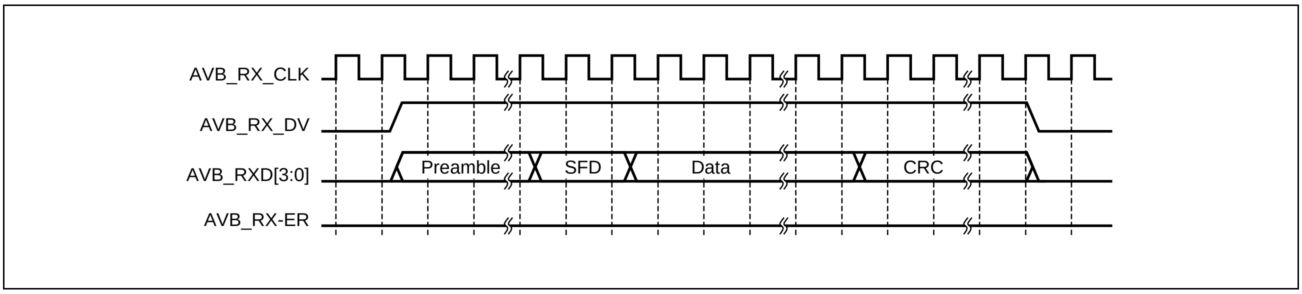


Figure 26.66 MII Frame Receive Timing (Normal Reception)

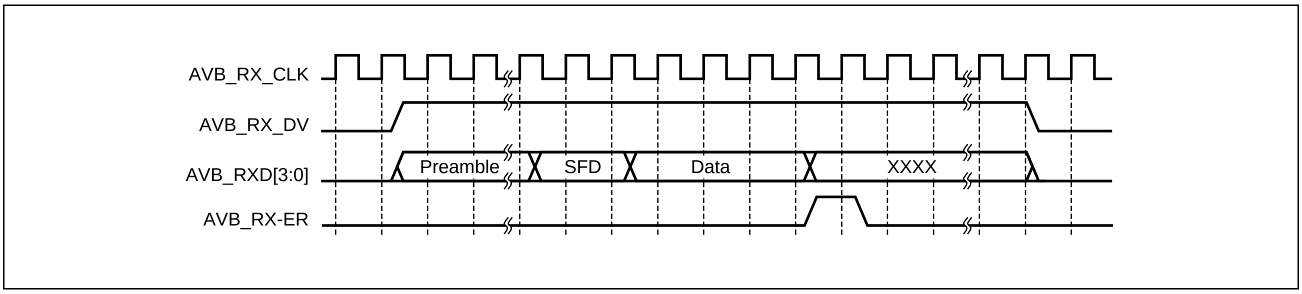


Figure 26.67 MII Frame Receive Timing (Reception Error (1))

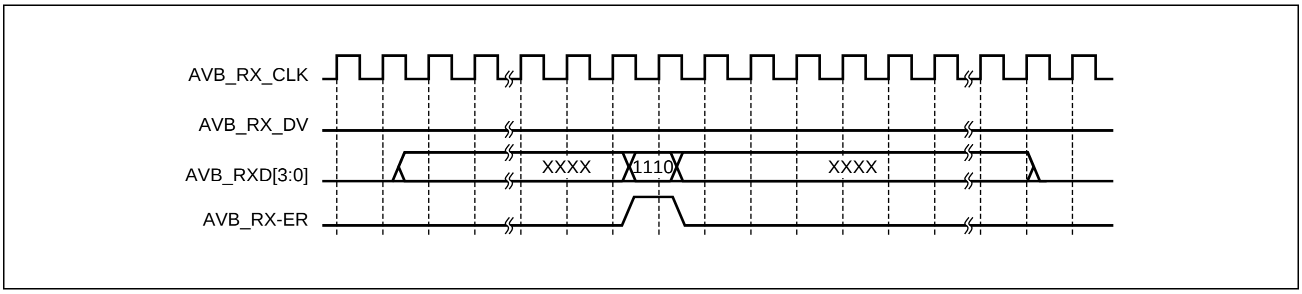


Figure 26.68 MII Fame Receive Timing (Reception Error (2))

26.4.12.2 Accessing MII Registers

MII registers in the PHY-LSI are accessed via ETNBnPIR in this LSI. ETNBnPIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3µ.

(1) MII Management Frame Format

Figure 26.69, MII Management Frame Format shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in **(2), MII Register Access Procedure**.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0*[1]	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

Remark:

PRE: 32 consecutive 1s
ST: Write of 01_B indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 00001_B if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
REGAD: Write of 0001_B if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10_B written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(d) Read: Bus already released in TA: control unnecessary

Note 1. The 2nd bit of TA only the ETNBnPIR.MDC bit should be written and ETNBnPIR.MMD should remain 0.

Figure 26.69 MII Management Frame Format

(2) MII Register Access Procedure

The program accesses MII registers via ETNBnPIR. Access is implemented by a combination of 1-bit- unit data write, 1-bit-unit data read, bus release, and independent bus release. **Figure 26.70, 1-Bit Data Write Flowchart** to **Figure 26.73, Independent Bus Release Flowchart (IDLE in Write in Figure 26.69)** show the MII register access timing. The timing will differ depending on the PHY-LSI type.

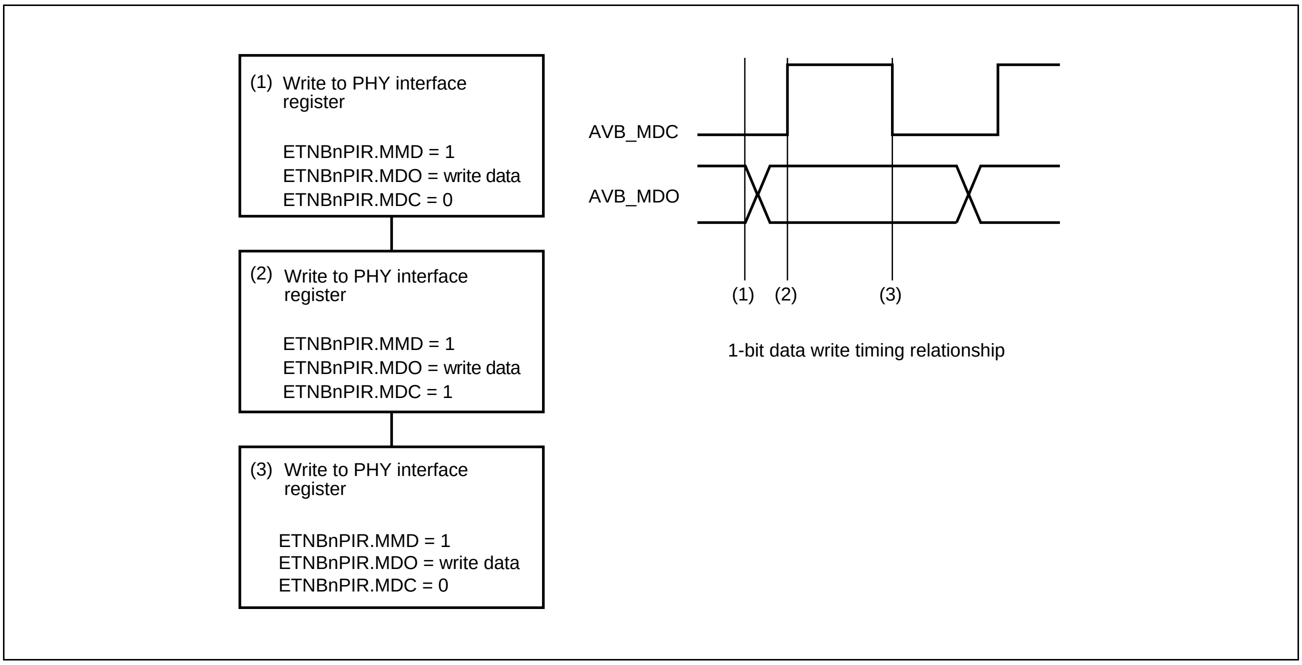


Figure 26.70 1-Bit Data Write Flowchart

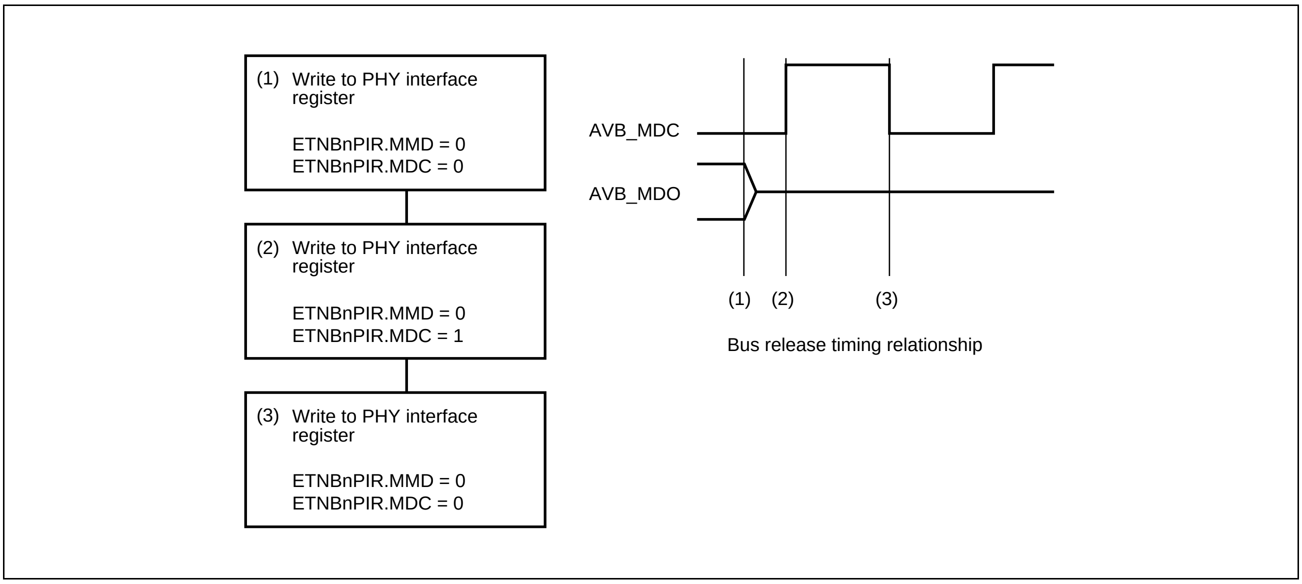


Figure 26.71 Bus Release Flowchart (TA in Read in **Figure 26.69**)

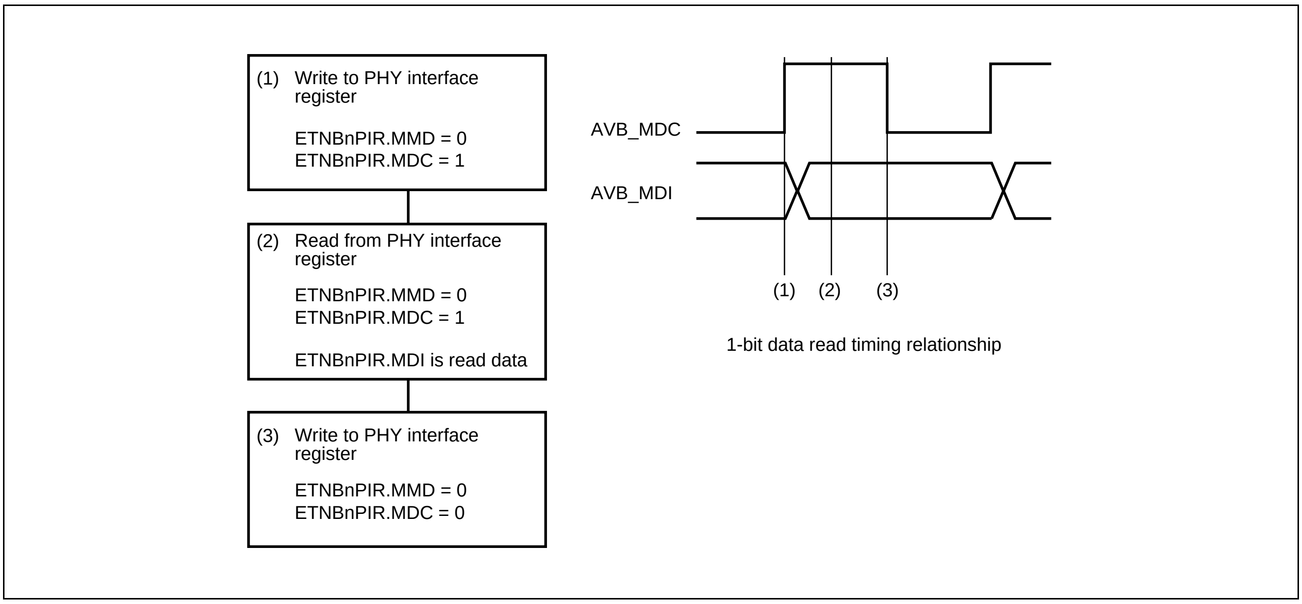


Figure 26.72 1-Bit Data Read Flowchart

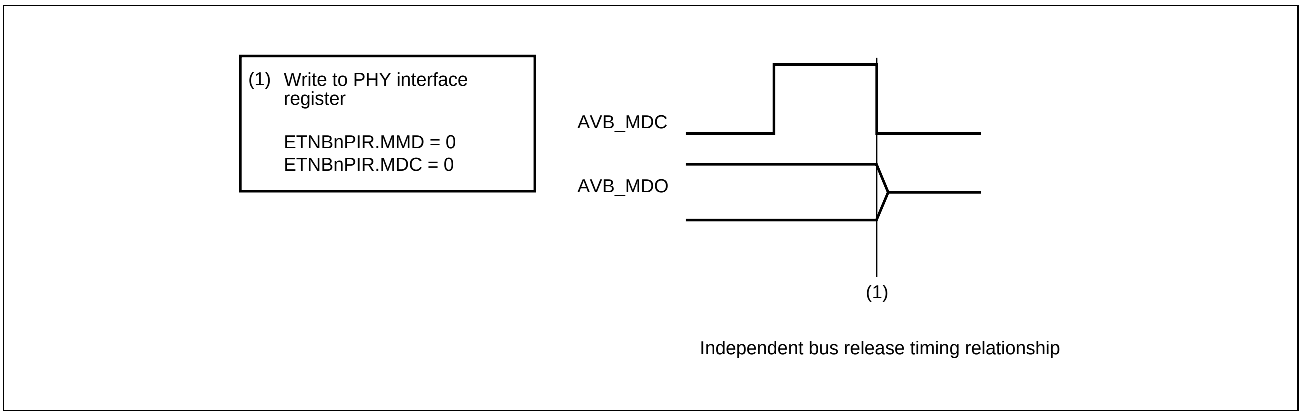


Figure 26.73 Independent Bus Release Flowchart (IDLE in Write in **Figure 26.69**)

26.4.13 Usage Notes

26.4.13.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to checksum calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. **Figure 26.74, Data Subject to Checksum Calculation** shows schematics indicating which parts of the Ethernet frames are calculated. Calculation involves 16-bit addition only; it does not involve bit inversion. Note that when the checksum data is valid, the CRC data (4 bytes) is not transferred as a receive frame, and the checksum data (sum data) is added automatically. **Figure 26.75, Data after Checksum Data Addition** shows schematics of Ethernet frames to which the checksum data has been added.

CAUTION

Also for the frames with VLANtag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

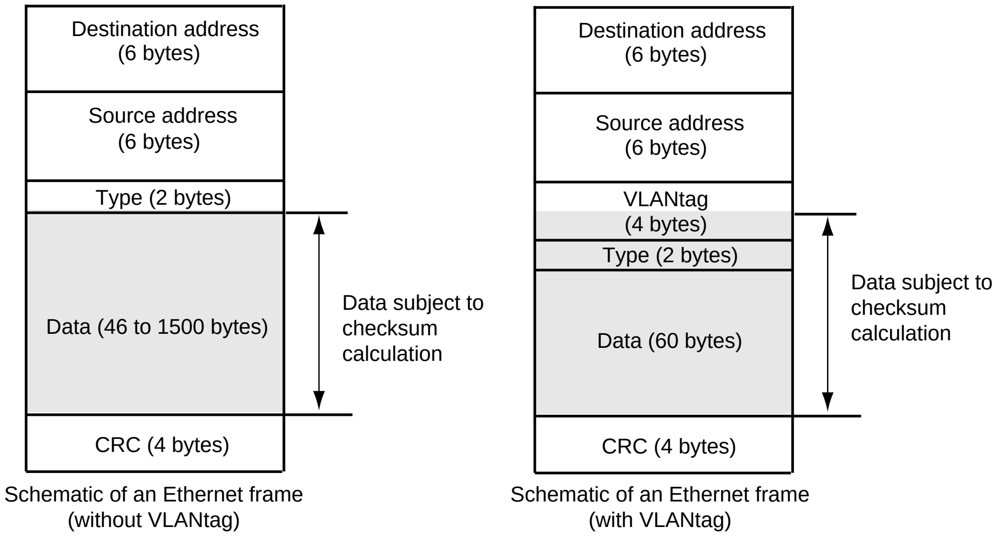


Figure 26.74 Data Subject to Checksum Calculation

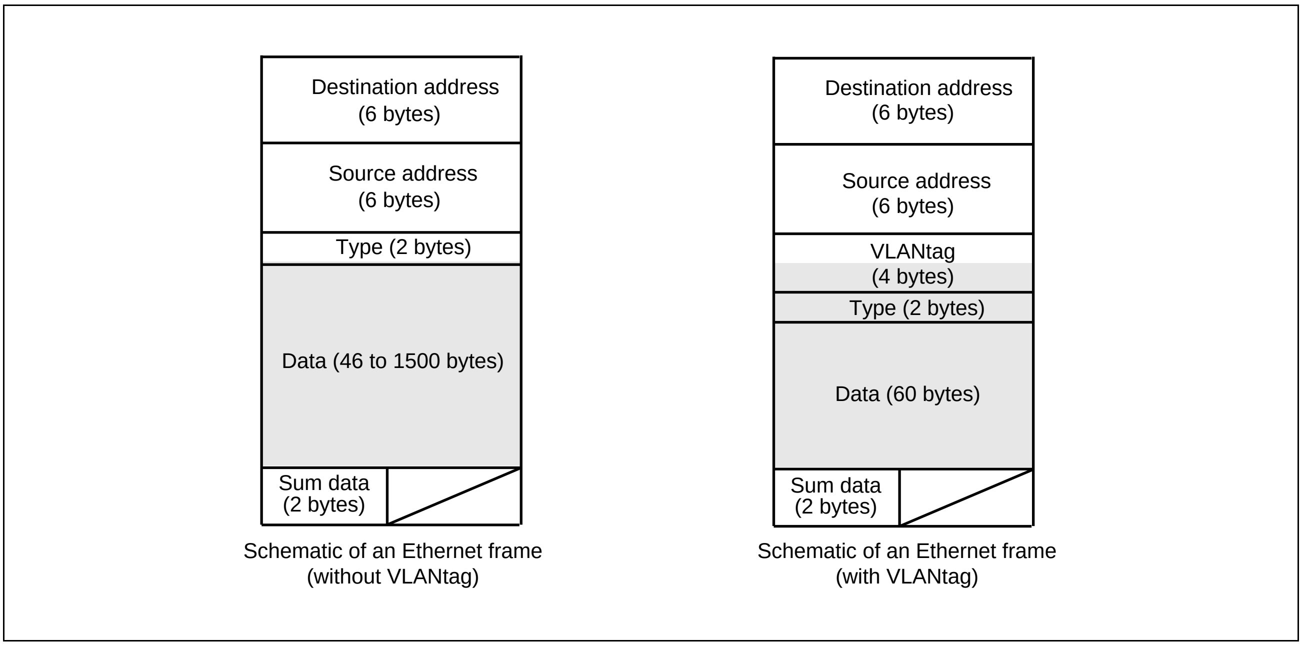


Figure 26.75 Data after Checksum Data Addition

26.4.13.2 Rx-FIFO Read Error May Not be Flagged when Using FEMPTY_ND Descriptor

After reading the last byte targeted to the FEMPTY_ND descriptor, there can be up to 3 bytes read from Rx-FIFO targeted to the next descriptor of the same frame. Rx-FIFO read errors of not stored data due to FEMPTY_ND are neither flagged in DESCR.EI nor by ETNBnEIS.QEF as per target specification.

When external ECC logic flags detect Rx-FIFO read error related to last byte targeted to FEMPTY_ND descriptor, data in the next descriptor may be corrupted without notification.

The issue is limited to implementations where Rx-FIFO error information is provided to Ethernet AVB and also to applications using FEMPTY_ND descriptor.

The issue only occurs when the first byte saved in the descriptor following FEMPTY_ND is not a multiple of 4 bytes inside the received frame.

Use a FEMPTY descriptor to store unwanted data if consistent error flagging is required.

26.4.13.3 When Trying to Release Non-Existing Timestamp FIFO Entry, New FIFO Update Flag May be Lost

When SW releases an entry of timestamp FIFO by writing ETNBnTCCR.TFR while this FIFO is empty (ETNBnTSR.TFFL is set to 0), flagging of next FIFO update may be inconsistent. The next timestamp is correctly stored in FIFO and the fill level is incremented to 1 but ETNBnTIS.TFUF is not set to 1.

The issue is limited to applications releasing FIFO entries without checking if there are entries available.

Do not release not existing FIFO entries.

26.4.13.4 gPTP Compare May Fail for Range of Compare Values

When the comparison value (ETNBnGPTC.PTCV) is in the range of [x – 1 to x + 1] (x is the configured increment value in ETNBnGTI.TIV), it may happen that a comparison match is not detected when Timer wraps around.

The issue is limited to applications using the AVTP comparison function. Do not configure comparison values inside the critical range.

26.4.13.5 UFC Stop Level Triggers ETNBnRIS2.QFFr Even No Received Frame is Lost

When a received frame is dropped due to non-availability of empty descriptor, ETNBnRIS2.QFFr is correctly set to 1.

Additionally, when ETNBnUFCVi.CVr reached the configured stop level (ETNBnUFCS.SLj), the queue full flag (ETNBnRIS2.QFFr) is set to 1 even before any further received frame is dropped.

The issue is limited to applications using unread frame counter with stop level function.

Such application gets information about lost received frames which in fact might not have lost.

26.4.13.6 ETNBnRIS0.FRFr may be lost when Data Processing Stops Close to or Below the Configured Warning Level.

When SW decrements UFC counter value (CV) to WL-1, resulting CV can still be WL due to recent new storage completion, ETNBnRIS0.FRFr is set to 0.

Any further reception will set ETNBnRIS0.FRFr as normal.

The issue is limited to applications using a single frame interrupt triggered by ETNBnRIS0.FRFr and not be able to process each interrupt in time.

The missing interrupt due to this effect is automatically recovered by the next frame reception as usual.

26.4.13.7 Notes on Using the Intelligent Checksum Function

The checksum calculation using the intelligent checksum function is not affected by the padding insertion specified by the receive data padding configuration register (ETNBnRPC). This is because the checksum calculation is performed when transferring the receive data from E-MAC to AVB-DMAC, while the padding of the receive data is performed when transferring the receive data from AVB-DMAC to the receive buffer in memory.

26.4.13.8 Receive Frame Interrupt and Descriptor Interrupt may be issued before Completion of Writing Data

When receive frame interrupt is issued, the software should check that the descriptor type is updated correctly before processing the frame data.

The descriptor type is shown in the descriptor field DESCR.DT (the address of the current descriptor is shown by the register ETNBnCDARq). If DESCR.DT is not yet updated by the DMA hardware (e.g. from FEMPTY to FEND), then the write of data to the memory is not completed. In that case the software should repeat checking the descriptor type until an update has happened.

Alternatively the software can use the Unread Frame Counter (UFC) value to compare with its processed descriptors to confirm there is no missing descriptor from the last received interrupt.

26.5 Detection and Correction of Errors in Ethernet AVB RAM

For details of ECC, see **Section 40A.2.6, ECC for Peripheral RAM**, **Section 40B.2.6, ECC for Peripheral RAM**.

Section 27 Single Edge Nibble Transmission (RSENT)

This section contains a generic description of the Renesas single edge nibble transmission (RSENT).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RSENT.

27.1 Features of RH850/F1KH, RH850/F1KM RSENT

27.1.1 Number of Units

This microcontroller has the following number of RSENT units.

Each RSENT unit has single channel interface.

Table 27.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	2	2	2
Name	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)

Table 27.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	2	2	2	2
Name	RSENTn (n = 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)

Table 27.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	2	2	2	2
Name	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)	RSENTn (n = 0, 1)

Table 27.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RSENT units are identified by the index “n”; for example, RSENTnTSPC (n = 0, 1) indicates the RSENTn time stamp register.

27.1.2 Register Base Addresses

RSENTn base addresses are listed in the following table.

RSENTn register addresses are given as offsets from the base addresses.

Table 27.5 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<RSENT0_base>	FFCF 0000_H
<RSENT1_base>	FFCF 0100_H

27.1.3 Clock Supply

The RSENTn clock supply is shown in the following table.

Table 27.6 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RSENTn	SENT Commutation Clock	CKSCLK_IPERI2	Commutation clock
	Register Access Clock (PCLK)	CPUCLK_L	Bus clock

Note: Commutation clock must be either 16 MHz or in the range of 32 MHz to 40 MHz and register access clock frequency can be in the range 16 MHz to 60 MHz.

27.1.4 Interrupt Requests

RSENTn interrupt requests are listed in the following table.

Table 27.7 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RSENT0			
INTSENT0SI	RSENT status interrupt	347	—
INTSENT0RI	RSENT receive interrupt	348	123
RSENT1			
INTSENT1SI	RSENT status interrupt	349	—
INTSENT1RI	RSENT receive interrupt	350	124

27.1.5 Reset Sources

RSENTn reset sources are listed in the following table. RSENTn is initialized by these reset sources.

Table 27.8 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RSENTn	All reset sources (ISORES)

27.1.6 External Input/Output Signals

External input/output signals of RSENT are listed below.

Table 27.9 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RSENT0		
sent_rx	RSENT Receive Data Input	SENT0RX
sent_spc	RSENT SPC Extension Output	SENT0SPCO
RSENT1		
sent_rx	RSENT Receive Data Input	SENT1RX
sent_spc	RSENT SPC Extension Output	SENT1SPCO

27.2 Overview

Overview of Functions

The RSENT interface supports the following standard specification (SAE J2716 version APR2016) functions (with extensions for support of 8 nibble data):

- Triple speed expansion Tick Time: Clock cycle (1 µs to 90 µs)
- Variable data transmission rate
  - up to 74.9 kbps (based on 8 nibble data at 3us clock rate)
  - up to 224.7 kbps (based on 8 nibble data at 1us clock rate)
- Unidirectional communication: Between the sensor and MCU
- Bidirectional communication: Between the sensor and MCU (supported in SPC mode)
- Single edge data transmission: Coded by the temporal distance of two serially-detected falling edges on a data line
- Transmission frame with up to 8 data nibbles and additionally status and communications nibbles
- Data transmission protected with CRC is available.
  - CRC data can be read with the RSENTnSRXD.SCRC and RSENTnFRXD.FCRC bits.
- Calibration phase in each data frame (RSENTnCPL.CPLV bits)
- Multiple sensors can connect to the RSENT channel that has the standard expansion function.

  Received data from sensors is detected by software or DMA.
- The timestamp function: Master or slave can be selected for the RSENT module. (RSENTnTSPC.TMS bit)
- The RSENT circuit consists of the following functions:
  - Data receive part
  - Clock recovery
  - Register group

27.2.1 Block Diagram

The following figure shows a block diagram of the RSENT module.

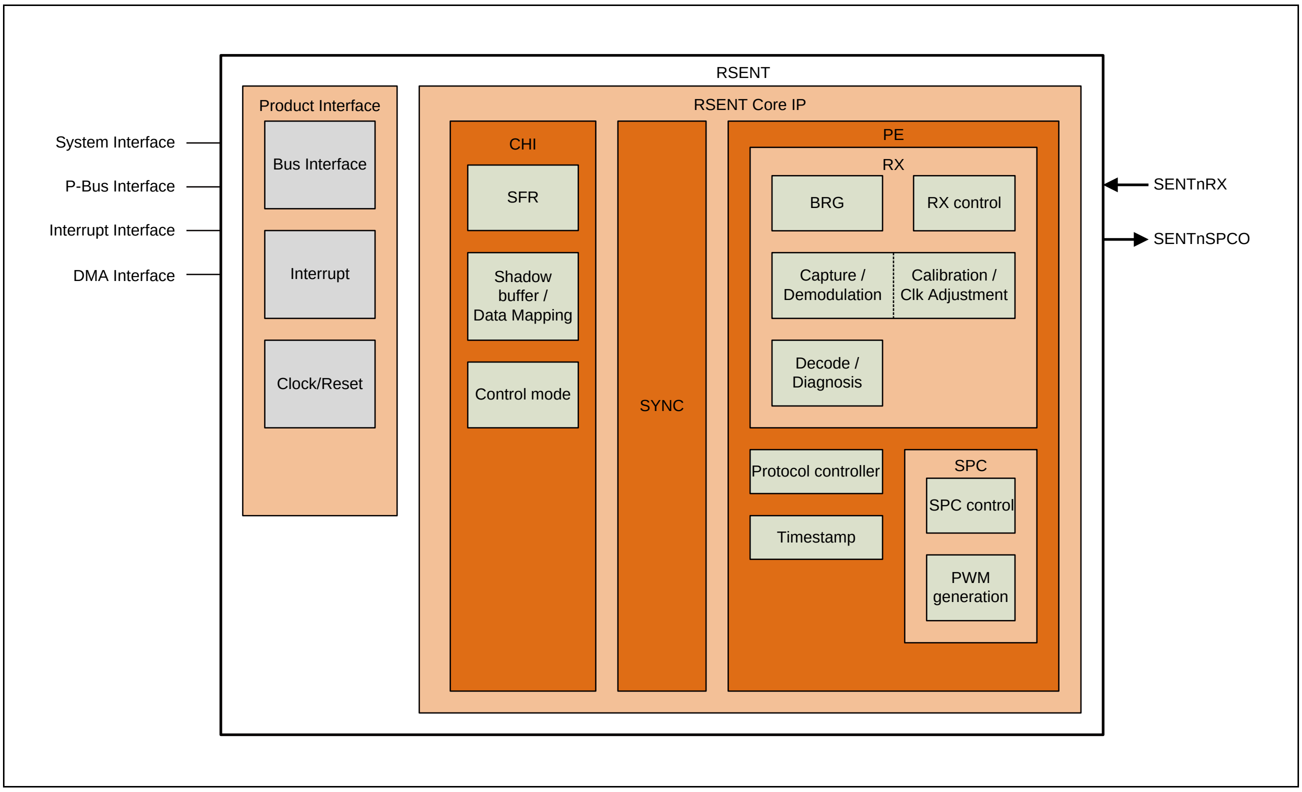


Figure 27.1 Block Diagram of RSENT

27.3 Registers

27.3.1 List of Registers

RSENT registers are listed in the following table.

For details about <RSENTn_base>, see **Section 27.1.2, Register Base Addresses**.

Table 27.10 List of Registers

Module Name	Register Name	Abbreviation	Address
RSENTn	RSENT timestamp register	RSENTnTSPC	<RSENTn_base> + 0000_H
	RSENT timestamp counter	RSENTnTSC	<RSENTn_base> + 0004_H
	RSENT communication configuration register	RSENTnCC	<RSENTn_base> + 0010_H
	RSENT baud rate prescaler register	RSENTnBRP	<RSENTn_base> + 0014_H
	RSENT interrupt/DMA enable register	RSENTnIDE	<RSENTn_base> + 0018_H
	RSENT mode control register	RSENTnMDC	<RSENTn_base> + $001C_H$
	RSENT SPC transmission register	RSENTnSPCT	<RSENTn_base> + 0020_H
	RSENT mode status register	RSENTnMST	<RSENTn_base> + 0024_H
	RSENT communication status register	RSENTnCS	<RSENTn_base> + 0028_H
	RSENT communication status clear register	RSENTnCSC	<RSENTn_base> + $002C_H$
	RSENT slow channel receive timestamp register	RSENTnSRTS	<RSENTn_base> + 0030_H
	RSENT slow channel receive data register	RSENTnSRXD	<RSENTn_base> + 0034_H
	RSENT calibration pulse length register	RSENTnCPL	<RSENTn_base> + 0038_H
	RSENT message length register	RSENTnML	<RSENTn_base> + $003C_H$
	RSENT fast channel receive timestamp register	RSENTnFRTS	<RSENTn_base> + 0040_H
	RSENT fast channel receive data register	RSENTnFRXD	<RSENTn_base> + 0044_H
	RSENT calibration pulse length mirror register	RSENTnCPLM	<RSENTn_base> + 0050_H
	RSENT message length mirror register	RSENTnMLM	<RSENTn_base> + 0054_H
	RSENT fast channel receive timestamp mirror register	RSENTnFRTSM	<RSENTn_base> + 0058_H
	RSENT expanded fast channel receive data register 0	RSENTnEFRD0	<RSENTn_base> + $005C_H$
	RSENT expanded fast channel receive data register 1	RSENTnEFRD1	<RSENTn_base> + 0060_H

27.3.2 RSENTnTSPC — RSENT Timestamp Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TTM[6:0]							—	TTPV[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.11 RSENTnTSPC Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
16	TMS	Timestamp Mode Selection 0: Master mode 1: Slave mode
15	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
14 to 8	TTM[6:0]	Timestamp Tick Multiplier 0000000_B: 1 0000001_B: 2 0000010_B: 3 : 1111111_B: 128
7	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
6 to 0	TTPV[6:0]	Timestamp Tick Prescaler Value 0000000_B: 1 0000001_B: 2 0000010_B: 3 : 1111111_B: 128

RSENTnTSPC.TMS (Timestamp Mode Selection)

This bit defines the timestamp counter synchronization mode.
For information about the timestamp clock settings, see **Section 27.4.2.1, Timestamp**.
When this bit is set to 0, the timestamp counter operates in master mode.
When writing $0000\ 0000_H$ to RSENTnTSC, the timestamp counter is cleared. In addition all RSENT timestamp counters operating as slave of RSENT are also cleared. For master-slave interconnection, see **Section 27.4.2.1(2), Timestamp Counter Operation**.
When this bit is set to 1, the timestamp counter operates in slave mode.
The timestamp counter is only cleared when writing $0000\ 0000_H$ to the timestamp counter of RSENT module that operates in master mode.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The RSENT module operating in slave mode should have the same timestamp counter prescaler settings as the RSENT module that operates in master mode.
The CPU should not set this bit to 1 for RSENT module that operates in master mode.

RSENTnTSPC.TTM (Timestamp Tick Multiplier)

These bits define the multiplication value of the 1-µs time tick used for the timestamp counter.
For timestamp clock configuration, see **Section 27.4.2.1, Timestamp**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode
(the RSENTnMST.OMS bits are 001_B).

RSENTnTSPC.TTPV (Timestamp Tick Prescaler Value)

These bits define the prescaler value to generate a 1-µs clock tick.
For timestamp clock configuration, see **Section 27.4.2.1, Timestamp**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode
(the RSENTnMST.OMS bits are 001_B).
The CPU should configure this value in such a way that, based on the supplied communication clock, a 1-µs clock tick is generated.

27.3.3 RSENTnTSC — RSENT Timestamp Counter

Access:	This register can be read or written in 32-bit units.
Address:	<RSENTn_base> + 0004$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.12 RSENTnTSC Register Contents

Bit Position	Bit Name	Function
31 to 0	TS[31:0]	Timestamp counter value

RSENTnTSC.TS (Timestamp)

These bits indicate the current timestamp counter value.
The CPU can only write desired values to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001$_B$).
When the timestamp counter is configured to operate in slave mode (RSENTnTSPC.TMS = 1), writing to this register has no effect when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011$_B$ or 101$_B$).
The timestamp counter is incremented on every timestamp counter tick (as configured in the RSENTnTSPC.TTPV and RSENTnTSPC.TTM bits) when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011$_B$ or 101$_B$).
When the timestamp counter is configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU writes 0000 0000$_H$ to these bits and RSENTnTSC.TS is set to 0000 0000$_H$.

When the slave mode setting is made for the timestamp counter of channel n (RSENTnTSPC.TMS = 1), writing to the timestamp counter of the channel that is the master for channel n leads to the RSENTnTSC.TS bits being set to 0000 0000$_H$.

For timestamp mode selection, see **Section 27.4.2.1, Timestamp**.

27.3.4 RSENTnCC — RSENT Communication Configuration Register

Access:	This register can be read or written in 32-bit units.
Address:	<RSENTn_base> + 0010_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SOPC	FCM	SCCD	FCCD	DCF	SMF[1:0]		PPTC	PPC	NDN[2:0]			SPCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.13 RSENTnCC Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
12	SOPC	SPC Output Polarity Control 0: SPC pulse active high 1: SPC pulse active low
11	FCM	Frame Check Method 0: Check against next calibration pulse 1: Check against previous calibration pulse
10	SCCD	Slow Channel CRC Check 0: Slow channel CRC check enabled 1: Slow channel CRC check disabled
9	FCCD	Fast Channel CRC Check 0: Fast channel CRC check enabled 1: Fast channel CRC check disabled
8	DCF	Data nibble CRC Format 0: SAE J2716 2010/2016 format 1: pre SAE J2716 2010 format
7, 6	SMF[1:0]	Serial Message Format 00_B: No serial message extraction 01_B: Short serial message format 10_B: Enhanced serial message format 11_B: Setting prohibited
5	PPTC	Pause Pulse Type Configuration 0: Pause pulse for variable message length 1: Pause pulse for fixed message length
4	PPC	Pause Pulse Configuration 0: Pause pulse absent 1: Pause pulse present

Table 27.13 RSENTnCC Register Contents

Bit Position	Bit Name	Function
3 to 1	NDN[2:0]	Number of Data Nibbles 000_B: 1 data nibble 001_B: 2 data nibbles 010_B: 3 data nibbles 011_B: 4 data nibbles 100_B: 5 data nibbles 101_B: 6 data nibbles 110_B: 7 data nibbles 111_B: 8 data nibbles
0	SPCE	SPC Mode Enable 0: SPC mode disabled 1: SPC mode enabled

RSENTnCC.SOPC (SPC Output Polarity Control)

When this bit is set to 0, the SPC pulse is sent as an active high signal. The default output value is low level.
When this bit is set to 1, the SPC pulse is sent as an active low signal. The default output value is high level.
For SPC operation, see also **Section 27.4.4, SPC Function**.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

NOTE

Any change to this bit from the default value becomes effective on the output value when entering the OPERATION ACTIVE mode (the RSENTnMST.OMS bits are 101_B). When entering RESET mode (the RSENTnMST.OMS bits are 000_B), the output level is set to the default value (low level).

RSENTnCC.FCM (Frame Check Method)

When this bit is set to 0, the current calibration pulse is compared to the next received calibration pulse.
The buffer update mechanism is operating according to the preferred option as described in SAE J2716 2016.
When this bit is set to 1, the current calibration pulse is compared to the previously received calibration pulse.
The buffer update mechanism is operating according to the second option as described in SAE J2716 2016 which should be only used if extra latency to process the second calibration pulse cannot be tolerated.
For buffer update timings, see also **Section 27.4.3.2(3), Fast Channel Message Reception**.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SCCD (Slow Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the slow channel is disabled. In this case, messages are stored in the slow channel message reception buffer with the received CRC.
When this bit is set to 1, the RSENTnCS.SCS bit is not set.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.FCCD (Fast Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the fast channel is disabled. In this case, messages are stored in the fast channel message reception buffer with the received CRC.
When this bit is set to 1, the RSENTnCS.FCS bit is not set.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.DCF (Data Nibble CRC Format)

This bit selects between the SAE J2716 2016 data nibble CRC format and the legacy format.
When this bit is set to 0 the recommended CRC implementation according to SAE J2716 2016 *Section 5.4.2.2* is selected.
When this bit is set to 1 the legacy CRC implementation according to SAE J2716 2008 (refer to SAE J2716 2016 *Section 5.4.2.1*) is selected.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (RSENTnMST.OMS = 001_B)

RSENTnCC.SMF (Serial Message Format)

These bits define the serial message format expected to be received for automatic extraction.
When these bits are set to 00_B, no serial message is extracted and the status and communications nibbles are stored in the RSENTnSRXD register.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).
The CPU shall set these bits to 00_B when RSENTnCC.SPCE is set to 1 and more than one sensor is connected to the RSENT module.

RSENTnCC.PPTC (Pause Pulse Type Configuration)

This bit defines the pause pulse type.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).
The CPU should not set this bit to 1 when the RSENTnCC.PPC bit is set to 0.

RSENTnCC.PPC (Pause Pulse Configuration)

This bit defines the presence or absence of the pause pulse.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.NDN (Number of Data Nibbles)

These bits define the number of data nibbles included in an SENT message.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SPCE (SPC Mode Enable)

This bit enables the SPC mode.
For details about SPC mode operation, see also **Section 27.4.4, SPC Function**.
The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

27.3.5 RSENTnBRP — RSENT Baud Rate Prescaler Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0014$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TTF[3:0]				—	TTI[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SCDV[6:0]							—	—	—	SCMV[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 27.14 RSENTnBRP Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
27 to 24	TTF[3:0]	Time Tick Decimal Fraction 0000$_B$: 0.0 µs 0001$_B$: 0.1 µs 0010$_B$: 0.2 µs : 1000$_B$: 0.8 µs 1001$_B$: 0.9 µs Other than above: Setting prohibited
23	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
22 to 16	TTI[6:0]	Time Tick Integer 0000000$_B$: 1 µs 0000001$_B$: 2 µs 0000010$_B$: 3 µs : 1011000$_B$: 89 µs 1011001$_B$: 90 µs Other than above: Setting prohibited
15	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
14 to 8	SCDV[6:0]	Sample Clock Division Value 0000000$_B$: 1 0000001$_B$: 2 0000010$_B$: 3 : 1111110$_B$: 127 1111111$_B$: 128
7 to 5	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.

Table 27.14 RSENTnBRP Register Contents

Bit Position	Bit Name	Function
4 to 0	SCMV[4:0]	Sample Clock Multiplication Value 00000_B: 1 00001_B: 2 00010_B: 3 : 11110_B: 31 11111_B: 32

RSENTnBRP.TTF (Time Tick Decimal Fraction)

These bits define the decimal part of the tick length in 0.1-µs granularity.
For tick length configuration, see **Section 27.4.2.2(2), RX and SPC Tick Settings**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.TTI (Time Tick Integer)

These bits define the integer part of the tick length.
For tick length configuration, see **Section 27.4.2.2(2), RX and SPC Tick Settings**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCDV (Sample Clock Division Value)

These bits define the division value for the sample clock generation logic.
For RSENTnBRP settings, see **Section 27.4.2.2(1), RX BRP Setting**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCMV (Sample Clock Multiplication Value)

These bits define the multiplication value for the sample clock generation logic.
For RSENTnBRP settings, see **Section 27.4.2.2(1), RX BRP Setting**.
The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

27.3.6 RSENTnIDE — RSENT Interrupt/DMA Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0018$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEIE	SMIE	SCIE	NRIE	CVIE	CLIE	FNIE	FEIE	FMIE	FCIE	FRIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.15 RSENTnIDE Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
10	SEIE	Slow Channel Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
9	SMIE	Slow Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
8	SCIE	Slow Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
7	NRIE	No Response Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
6	CVIE	Calibration Pulse Length Variation Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
5	CLIE	Calibration Pulse Length Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
4	FNIE	Fast Channel Nibble Count Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
3	FEIE	Fast Channel Nibble Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
2	FMIE	Fast Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
1	FCIE	Fast Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled

Table 27.15 RSENTnIDE Register Contents

Bit Position	Bit Name	Function
0	FRIE	Fast Channel Receive Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled

RSENTnIDE.SEIE (Slow Channel Encoding Error Interrupt Enable)

This bit enables the generation of the slow channel encoding error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SMIE (Slow Channel Message Lost Interrupt Enable)

This bit enables the generation of the slow channel message lost interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SCIE (Slow Channel CRC Error Interrupt Enable)

This bit enables the generation of the slow channel CRC error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.NRIE (No Response Error Interrupt Enable)

This bit enables the generation of the no response error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).
The CPU should not set this bit when the SPC mode is disabled (RSENTnCC.SPCE set to 0).

RSENTnIDE.CVIE (Calibration Pulse Length Variation Error Interrupt Enable)

This bit enables the generation of the calibration pulse length variation error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.CLIE (Calibration Pulse Length Error Interrupt Enable)

This bit enables the generation of the calibration pulse length error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FNIE (Fast Channel Nibble Count Error Interrupt Enable)

This bit enables the generation of the fast channel nibble count error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FEIE (Fast Channel Nibble Encoding Error Interrupt Enable)

This bit enables the generation of the fast channel nibble encoding error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FMIE (Fast Channel Message Lost Interrupt Enable)

This bit enables the generation of the fast channel message lost interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FCIE (Fast Channel CRC Error Interrupt Enable)

This bit enables the generation of the fast channel CRC error interrupt.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FRIE (Fast Channel Receive Interrupt Enable)

This bit enables the generation of the fast channel receive interrupt.
The fast channel receive interrupt can be also used to notify a DMA request.
The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

27.3.7 RSENTnMDC — RSENT Mode Control Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + $001C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 27.16 RSENTnMDC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
2 to 0	OMC[2:0]	Operation Mode Control 000_B: RESET 001_B: CONFIGURATION 011_B: OPERATION IDLE 101_B: OPERATION ACTIVE Other than above: Setting prohibited

RSENTnMDC.OMC (Operation Mode Control)

These bits are used to control the operation mode of the RSENT module.

- 000_B: RESET
  In RESET mode, the mode can only be changed to CONFIGURATION mode.
- 001_B: CONFIGURATION
  In CONFIGURATION mode, the mode can only be changed to RESET mode or OPERATION ACTIVE mode.
- 011_B: OPERATION IDLE
  In OPERATION IDLE mode, the mode can be changed to OPERATION ACTIVE mode, CONFIGURATION mode, or RESET mode.
- 101_B: OPERATION ACTIVE
  In OPERATION ACTIVE mode, the mode can be changed to OPERATION IDLE mode, CONFIGURATION mode, or RESET mode. However, it is recommended to process to the OPERATION IDLE mode first.
  For the recommended methods to change between operation modes, see **Section 27.4.3.1, Changing Operation Modes**.
- Other than above: Setting prohibited
  The CPU should not write any other value than listed above into this register.
  The CPU should follow the mode change flows as shown in **Section 27.4.3.1, Changing Operation Modes**.

27.3.8 RSENTnSPCT — RSENT SPC Transmission Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0020_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TLL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.17 RSENTnSPCT Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
6 to 0	TLL[6:0]	Length of the Trigger Low Phase in Ticks 0000000_B: 1 tick 0000001_B: 2 ticks 0000010_B: 3 ticks : 1111110_B: 127 ticks 1111111_B: 128 ticks

RSENTnSPCT.TLL (Trigger Low Length)

These bits define the length of the SPC trigger pulse.

When the CPU writes to these bits, an SPC trigger pulse with the configured length is sent starting from the next SPC trigger tick. In case RSENTnCS.NRS is set by the RSENT module following a write to these bits, no SPC trigger pulse is sent.

For details about SPC communication, see **Section 27.4.4, SPC Function**.

The CPU can only write to these bits if the RSENT module is in the OPERATION ACTIVE mode (the RSENTnMST.OMS bits are 101_B) and SPC communication is enabled (RSENTnCC.SPCE is 1_B).

It is important to note that two consecutive write access might not cause a no response error as the previous request might not have started yet.

After writing to this register, the CPU should wait for at least one SPC trigger tick before writing again to this register.

27.3.9 RSENTnMST — RSENT Mode Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMS[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.18 RSENTnMST Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is read.
2 to 0	OMS[2:0]	Operation Mode 000_B: RESET 001_B: CONFIGURATION 011_B: OPERATION IDLE 101_B: OPERATION ACTIVE Other than above: Reserved

RSENTnMST.OMS (Operation Mode Status)

These bits indicate the current operation mode. These bits are read only.
These bits are updated after a mode change request is made in the RSENTnMDC.OMC register.

- 000_B: RESET mode
  When in RESET mode, all registers are set to their reset values and write access to all registers except the RSENTnMDC register is disabled.
  When in RESET mode, SENT communication is disabled.
- 001_B: CONFIGURATION mode
  When in CONFIGURATION mode, write access to the timestamp registers (RSENTnTSPC and RSENTnTSC register), configuration registers (RSENTnCC and RSENTnBRP register), RSENTnIDE register, and mode control register (RSENTnMDC.OMC) is enabled.
  When in CONFIGURATION mode, SENT communication is disabled.
  When entering CONFIGURATION mode, all status registers and receive buffer registers are set to their reset values.
- 011_B: OPERATION IDLE mode
  In OPERATION IDLE mode, no reception or SPC trigger transmission is possible.
  When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.
- 101_B: OPERATION ACTIVE mode
  In OPERATION ACTIVE mode, reception and SPC trigger transmission are possible.
- Other than above: Reserved

27.3.10 RSENTnCS — RSENT Communication Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0028$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SES	SMS	SCS	NRS	CVS	CLS	FNS	FES	FMS	FCS	FRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.19 RSENTnCS Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is read.
10	SES	Slow channel Encoding Error Interrupt Status 0: Not detected 1: Detected
9	SMS	Slow Channel Message Lost Interrupt Status 0: Not detected 1: Detected
8	SCS	Slow Channel CRC Error Interrupt Status 0: Not detected 1: Detected
7	NRS	No Response Error Interrupt Status 0: Not detected 1: Detected
6	CVS	Calibration Pulse Length Variation Error Interrupt Status 0: Not detected 1: Detected
5	CLS	Calibration Pulse Length Error Interrupt Status 0: Not detected 1: Detected
4	FNS	Fast Channel Nibble Count Error Interrupt Status 0: Not detected 1: Detected
3	FES	Fast Channel Nibble Encoding Error Interrupt Status 0: Not detected 1: Detected
2	FMS	Fast Channel Message Lost Interrupt Status 0: Not detected 1: Detected
1	FCS	Fast Channel CRC Error Interrupt Status 0: Not detected 1: Detected

Table 27.19 RSENTnCS Register Contents

Bit Position	Bit Name	Function
0	FRS	Fast Channel Receive Interrupt Status 0: Not detected 1: Detected

RSENTnCS.SES (Slow Channel Encoding Error Status)

This bit represents the slow channel encoding error status.
This bit is read only.
In the short serial message format (RSENTnCC.SMF = 01_B), this bit is set when the sequence on serial start bit (bit #3) is different from “1000 0000 0000 0000_B” (a single 1 and 15 0s).
In the enhanced serial message format (RSENTnCC.SMF = 10_B), this bit is set when following the reception of a start sequence (“0111 1110_B”) on the serial data bit 3, bit 13 or bit 18 are not received as “0”.
When this bit is set in the short serial message format the received communication and status nibble is used to assemble a serial message.
When this bit is set in the enhanced serial message format the RSENT module checks the presence of a new start sequence at the same time and uses the received communication and status nibble to assemble a serial message.
This bit is cleared when writing 1 to RSENTnCSC.SEC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SMS (Slow Channel Message Lost Status)

This bit represents the slow channel message lost status.
This bit is read only.
This bit is set when there is an attempt to update the slow channel message reception buffer, but the previous message has not been read yet.
This bit is cleared when writing 1 to RSENTnCSC.SEC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SCS (Slow Channel CRC Error Status)

This bit represents the slow channel CRC error status.
This bit is read only.
This bit is set when a CRC error is detected on the slow channel and the slow channel CRC detection is enabled (RSENTnCC.SCCD is set to 0).
This bit is cleared when writing 1 to RSENTnCSC.SCC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.NRS (No Response Error Status)

This bit represents the no response error status.
This bit is read only.
This bit is set when

- The CPU writes to the RSENTnSPCT.TLL bits and
- SPC mode enabled (RSENTnCC.SPCE set to 1) and
- No complete response was received from the sensor for the previous SPC trigger.

This bit is set after 4 PCLK + 5 clkc (Maximum time) from the CPU writes to the RSENTnSPCT.TLL.
This bit is cleared when writing 1 to RSENTnCSC.NRC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CVS (Calibration Pulse Length Variation Error Status)

This bit represents the calibration pulse length variation error status.
This bit is read only.
When RSENTnCC.PPTC is 0, then this bit is set when two successive calibration pulses differ by more than 1.5625 %.
When RSENTnCC.PPTC is 1, this bit is never set. In this mode (pause pulse with fixed message length), the CPU needs to check the variation of the ratio of calibration pulse to message length by reading the RSENTnCPL and RSENTnML registers.
This bit is cleared when writing 1 to RSENTnCSC.CVC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CLS (Calibration Pulse Length Error Status)

This bit represents the calibration pulse length error status.
This bit is read only.
This bit is set when the measured calibration pulse length is less than 42 clock ticks or more than 70 clock ticks (deviation of 25 % from nominal length).
This bit is cleared when writing 1 to RSENTnCSC.CLC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FNS (Fast Channel Nibble Count Error Status)

This bit represents the fast channel nibble count error status.
This bit is read only.
This bit is set when there is an unexpected number of falling edges between two calibration pulses.
This bit is cleared when writing 1 to RSENTnCSC.FNC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FES (Fast Channel Nibble Encoding Error Status)

This bit represents the fast channel nibble encoding error status.
This bit is read only.
This bit is set when on the fast channel a measured nibble period is less than 12 clock ticks or more than 27 clock ticks.
This bit is cleared when writing 1 to RSENTnCSC.FEC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FMS (Fast Channel Message Lost Status)

This bit represents the fast channel message lost status.
This bit is read only.
This bit is set when the fast channel message reception buffer is updated, but the previous messages in the foreground and background buffer have not been read yet.
This bit is cleared when writing 1 to RSENTnCSC.FMC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FCS (Fast Channel CRC Error Status)

This bit represents the fast channel CRC error status.
This bit is read only.
This bit is set when a CRC error is detected on the fast channel and the fast channel CRC detection is enabled (RSENTnCC.FCCD is set to 0).
This bit is cleared when writing 1 to RSENTnCSC.FCC.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FRS (Fast Channel Receive Status)

This bit represents the fast channel receive status.
This bit is read only.
This bit is set when the fast channel message reception buffer was updated.
This bit is cleared when the CPU reads the RSENTnFRXD or RSENTnEFRD1 register.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).
If the set condition occurs simultaneously with the clear condition, the bit is set.

27.3.11 RSENTnCSC — RSENT Communication Status Clear Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + $002C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEC	SMC	SCC	NRC	CVC	CLC	FNC	FEC	FMC	FCC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 27.20 RSENTnCSC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.
10	SEC	Slow Channel Encoding Error Clear 0: — 1: Clear
9	SMC	Slow Channel Message Lost Clear 0: — 1: Clear
8	SCC	Slow Channel CRC Error Clear 0: — 1: Clear
7	NRC	No Response Error Clear 0: — 1: Clear
6	CVC	Calibration Pulse Length Variation Error Clear 0: — 1: Clear
5	CLC	Calibration Pulse Length Error Clear 0: — 1: Clear
4	FNC	Fast Channel Nibble Count Error Clear 0: — 1: Clear
3	FEC	Fast Channel Nibble Encoding Error Clear 0: — 1: Clear
2	FMC	Fast Channel Message Lost Clear 0: — 1: Clear
1	FCC	Fast Channel CRC Error Clear 0: — 1: Clear
0	Reserved	When read, the value after reset is read. When writing, write the initial value after reset.

RSENTnCSC.SEC (Slow Channel Encoding Error Clear)

Writing 1 sets RSENTnCS.SES to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.SMC (Slow Channel Message Lost Clear)

Writing 1 sets RSENTnCS.SMS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.SCC (Slow Channel CRC Error Clear)

Writing 1 sets RSENTnCS.SCS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.NRC (No Response Error Clear)

Writing 1 sets RSENTnCS.NRS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.CVC (Calibration Pulse Length Variation Error Clear)

Writing 1 sets RSENTnCS.CVS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.CLC (Calibration Pulse Length Error Clear)

Writing 1 sets RSENTnCS.CLS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.FNC (Fast Channel Nibble Count Error Clear)

Writing 1 sets RSENTnCS.FNS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.FEC (Fast Channel Nibble Encoding Error Clear)

Writing 1 sets RSENTnCS.FES to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.FMC (Fast Channel Message Lost Clear)

Writing 1 sets RSENTnCS.FMS to 0. Writing 0 has no effect. This bit is always read as 0.

RSENTnCSC.FCC (Fast Channel CRC Error Clear)

Writing 1 sets RSENTnCS.FCS to 0. Writing 0 has no effect. This bit is always read as 0.

27.3.12 RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<RSENTn_base> + 0030_H
Value after reset:	$0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.21 RSENTnSRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	STS[31:0]	Slow Channel Receive Timestamp

RSENTnSRTS.STS (Slow Channel Timestamp)

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated with the timestamp counter value of the last frame provided to the slow channel message.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.13 RSENTnSRXD — RSENT Slow Channel Receive Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SND	—	SCRC[5:0]						—	—	—	SMGC	IDD[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.22 RSENTnSRXD Register Contents

Bit Position	Bit Name	Function
31	SND	Slow Channel New Data 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	Reserved	When read, the value after reset is read.
29 to 24	SCRC[5:0]	Slow Channel CRC Data
23 to 21	Reserved	When read, the value after reset is read.
20	SMGC	Slow Channel Configuration Bit Data
19 to 0	IDD[19:0]	Slow Channel Data / ID Information

RSENTnSRXD.SND (Slow Channel New Data)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.
This bit is read only.
This bit is set when the slow channel message reception buffer is updated.
This bit is cleared automatically whenever it is read.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SCRC (Slow Channel CRC)

These bits are representing the slow channel CRC data.
These bits are read only.
These bits are updated when the slow channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SMGC (Slow Channel Configuration Bit)

This bit represents the slow channel configuration bit.
This bit is read only.
This bit is updated when the slow channel message reception buffer is updated.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.IDD (ID/Data)

These bits are representing the slow channel data and ID information.
The alignment within this register depends on the message format. For details, see **Section 27.4.3.2(5), Slow Channel Message Reception**.
These bits are read only.
These bits are updated when the slow channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.14 RSENTnCPL — RSENT Calibration Pulse Length Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0038$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLV [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.23 RSENTnCPL Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is read.
16 to 0	CPLV[16:0]	Calibration Pulse Length Value of Received Message

RSENTnCPL.CPLV (Calibration Pulse Length Value)

These bits are used by the CPU to calculate the ratio of two consecutive calibration pulses or the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.
For details, refer to *Section 5.3.3* of the SAE J2716 2016 specification.
These bits are read only.
In modes other than pause pulse with fixed message length mode (RSENTnCC.PPTC = 1$_{B}$) or SPC mode (RSENTnCC.SPCE = 1$_{B}$), these bits are invalid and should not be used.
Updating of the fast channel message reception buffer is storage of the value counted over the calibration pulse length (one tick time × the number of ticks between calibration pulses) based on the sample clock (f_{SAMPLE} = 16 MHz).
These bits are cleared when the RSENTnMST.OMS bits are changed to 001$_{B}$ (CONFIGURATION).

27.3.15 RSENTnML — RSENT Message Length Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + $003C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.24 RSENTnML Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is read.
20 to 0	MLV[20:0]	Message Length Value of Received Message

RSENTnML.MLV (Message Length Value)

These bits are used by the CPU to calculate the ratio of the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

In modes other than pause pulse with fixed message length, these bits are invalid and should not be used.

These bits are updated with the measured message length in sample clock ticks when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.16 RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.25 RSENTnFRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	FTS[31:0]	Fast Channel Receive Timestamp

RSENTnFRTS.FTS (Fast Channel Receive Timestamp)

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.17 RSENTnFRXD — RSENT Fast Channel Receive Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SNDM	FND	FCCN[1:0]		FCRC[3:0]				ND[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.26 RSENTnFRXD Register Contents

Bit Position	Bit Name	Function
31	SNDM	Slow Channel New Data Mirror 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	FND	Fast Channel New Data 0: Fast channel frame data is not updated since last read. 1: Fast channel frame data is updated since last read.
29, 28	FCCN[1:0]	Fast Channel Communication Nibble
27 to 24	FCRC[3:0]	Fast Channel CRC Data
23 to 0	ND[23:0]	Fast Channel Nibble Data

RSENTnFRXD.SNDM (Slow Channel New Data Mirror)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.
This bit is read only.
This bit is set when the slow channel message reception buffer is updated.
This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FND (Fast Channel New Data)

This bit indicates that the fast channel message reception buffer is holding data that has not been read.
This bit is read only.
This bit is set when the fast channel message reception buffer is updated.
This bit is cleared automatically whenever RSENTnFRXD or RSENTnEFRD1 is read.
This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCCN (Fast Channel Communication Nibble)

These bits are representing the fast channel communication nibble bits [1:0].
These bits are read only.
These bits are updated when the fast channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCRC (Fast Channel CRC)

These bits are representing the fast channel CRC data.
These bits are read only.
These bits are updated when the fast channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.ND (Fast Channel Nibble Data)

These bits are representing the fast channel nibble data.
The alignment of the nibble data depends on nibble count. For details, see **Section 27.4.3.2(3), Fast Channel Message Reception**.
These bits are read only.
These bits are updated when the fast channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.18 RSENTnCPLM — RSENT Calibration Pulse Length Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLVM [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.27 RSENTnCPLM Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is read.
16 to 0	CPLVM[16:0]	Calibration Pulse Length Value (Mirror) of Received Message

RSENTnCPLM.CPLVM (Calibration Pulse Length Value (Mirror))

These bits are mirror bit of the RSENTnCPL.CPLV.
These bits are read only.
In modes other than pause pulse with fixed message length mode (RSENTnCC.PPTC = 1_B) or SPC mode (RSENTnCC.SPCE = 1_B), these bits are invalid and should not be used.
Updating of the fast channel message reception buffer is storage of the value counted over the calibration pulse length (one tick time × the number of ticks between calibration pulses) based on the sample clock (f_{SAMPLE} = 16 MHz).
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.19 RSENTnMLM — RSENT Message Length Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0054$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLVM[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.28 RSENTnMLM Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is read.
20 to 0	MLVM[20:0]	Message Length Value (Mirror) of Received Message

RSENTnMLM.MLVM (Message Length Value (Mirror))

These bits are mirror bit of the RSENTnML.MLV.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001$_B$ (CONFIGURATION).

27.3.20 RSENTnFRTSM — RSENT Fast Channel Receive Timestamp Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0058$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTSM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTSM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.29 RSENTnFRTSM Register Contents

Bit Position	Bit Name	Function
31 to 0	FTSM[31:0]	Fast Channel Receive Timestamp (Mirror)

RSENTnFRTSM.FTSM (Fast Channel Receive Timestamp (Mirror))

These bits are mirror bit of the RSENTnFRTS.FTS.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001$_B$ (CONFIGURATION).

27.3.21 RSENTnEFRD0 — RSENT Expanded Fast Channel Receive Data Register 0

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + $005C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SNDM	FND	FCCN[1:0]		FCRC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.30 RSENTnEFRD0 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is read.
7	SNDM	Slow Channel New Data (Mirror) 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
6	FND	Fast Channel New Data 0: Fast channel frame data is not updated since last read. 1: Fast channel frame data is updated since last read.
5 to 4	FCCN[1:0]	Fast Channel Communication Nibble Bits [1:0]
3 to 0	FCRC[3:0]	Fast Channel CRC Data

RSENTnEFRD0.SNDM (Slow Channel New Data (Mirror))

This bit indicates that the slow channel message reception buffer is holding data that has not been read.
This bit is read only.
This bit is set when the slow channel message reception buffer is updated.
This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnEFRD0.FND (Fast Channel New Data)

This bit indicates that the fast channel message reception buffer is holding data that has not been read.
This bit is read only.
This bit is set when the fast channel message reception buffer is updated.
This bit is cleared automatically whenever RSENTnFRXD or RSENTnEFRD1 is read.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnEFRD0.FCCN (Fast Channel Communication Nibble)

These bits are representing the fast channel communication nibble bits [1:0].
These bits are read only.
These bits are updated when the fast channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnEFRD0.FCRC (Fast Channel CRC)

These bits are representing the fast channel CRC data.
These bits are read only.
These bits are updated when the fast channel message reception buffer is updated.
These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.3.22 RSENTnEFRD1 — RSENT Expanded Fast Channel Receive Data Register 1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.31 RSENTnEFRD1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ND[31:0]	Fast Channel Nibble Data

RSENTnEFRD1.ND (Nibble Data)

These bits are representing the fast channel nibble data.

The alignment of the nibble data depends on nibble count. For details, see **Section 27.4.3.2(3), Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

27.4 Operation

27.4.1 Modes of Operation

The RSENT module can be in one of the following modes:

- RESET mode
- CONFIGURATION mode
- OPERATION IDLE mode
- OPERATION ACTIVE mode

The CPU should follow the mode change flows as shown in **Section 27.4.3.1, Changing Operation Modes**.

The current operation mode status can be seen in the RSENTnMST.OMS bits.

Figure 27.2, Transition between Operation Modes shows the possible transitions between the channel modes:

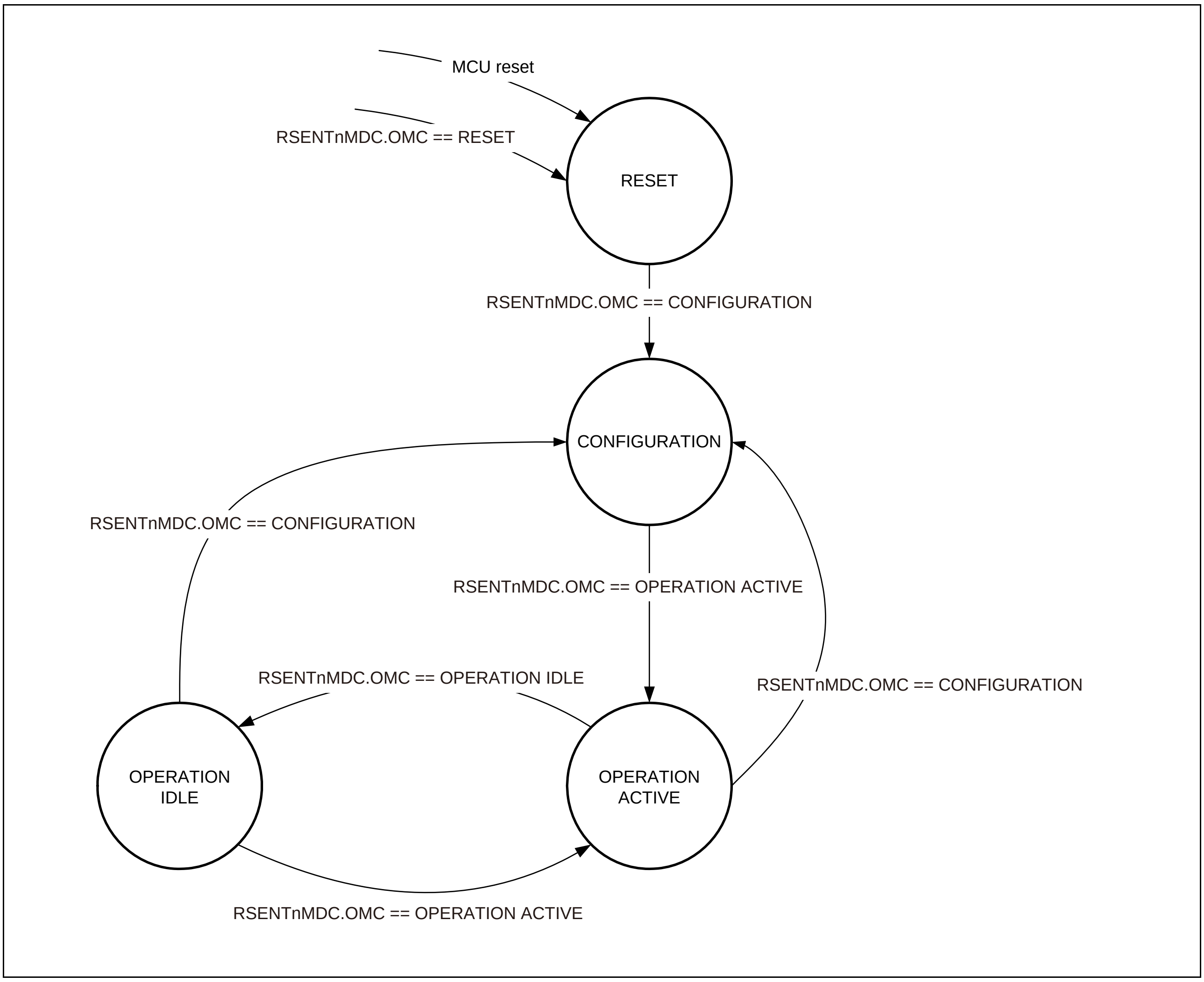


Figure 27.2 Transition between Operation Modes

27.4.1.1 RESET Mode

This mode is the initial mode that the RSENT module automatically enters after the hardware reset (MCU reset) is cleared. Its purpose is to provide a clean reset of the registers in the RSENT module.

The RESET mode is also entered after the RSENTnMDC.OMC bits have been set to 000_B. In this state, all, configuration, control (except RSENTnMDC.OMC bits), and status registers are set to their reset value. Any on-going transmission or reception process is stopped immediately and the interface pins of the RSENT module are set to their default values.

Read access to all registers is possible in this state. Write access is limited to the RSENTnMDC register.

27.4.1.2 CONFIGURATION Mode

The CONFIGURATION mode is entered after the RSENTnMDC.OMC bits have been set to 001_B.

The interface pins of the RSENT module are set to their default values.

Regarding the output polarity setting of SENTnSPCO pin and the timing that becomes effective, please refer to the explanation of “RSENTnCC.SOPC (SPC Output Polarity Control)” in **Section 27.3.4, RSENTnCC — RSENT Communication Configuration Register**.

In this state, all status registers (RSENTnCS) and the receive buffer registers (RSENTnSRTS, RSENTnSRXD, RSENTnCPL, RSENTnML, RSENTnFRTS, and RSENTnFRXD) are set to their default value.

Read access to all registers is possible in this state.

Write access is limited to both timestamp registers (RSENTnTSPC and RSENTnTSC) and configuration registers (RSENTnCC, RSENTnBRP, RSENTnIDE, RSENTnMDC, and RSENTnCSC).

27.4.1.3 OPERATION IDLE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 011_B.

In OPERATION IDLE mode, no reception and transmission are done.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, and RSENTnCSC.

27.4.1.4 OPERATION ACTIVE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 101_B.

In OPERATION ACTIVE mode, transmission and reception can take place.

Frame reception and status flagging starts after a valid calibration pulse (including the falling edge at the beginning) was detected.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, RSENTnSPCT, and RSENTnCSC.

27.4.1.5 Register Behavior in Operation Modes

Table 27.32, Register Behavior in Operation Modes shows the register behavior when the RSENT module transitions to the indicated operation modes. The table also gives an overview about the access restriction in each operation mode.

Table 27.32 Register Behavior in Operation Modes

		MCU Reset	RESET		CONFIGURATION		OPERATION IDLE		OPERATION ACTIVE	
Register Name	Symbol	Change	Change	R/W	Change	R/W	Change	R/W	Change	R/W
Timestamp register	RSENTnTSPC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Timestamp counter register	RSENTnTSC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W *1	Unchanged	R/W *1
Communication configuration register	RSENTnCC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Baud rate prescaler register	RSENTnBRP	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
Interrupt/DMA enable register	RSENTnIDE	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
Mode control register	RSENTnMDC	0000 0000$_H$	0000 0000$_H$	R/W	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
SPC transmission register	RSENTnSPCT	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R	Unchanged	R	Unchanged	R/W
Mode status register	RSENTnMST	0000 0000$_H$	0000 0000$_H$	R	0000 0001$_H$	R	0000 0003$_H$	R	0000 0005$_H$	R
Communication status register	RSENTnCS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Communication status clear register	RSENTnCSC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
Slow channel receive timestamp register	RSENTnSRTS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Slow channel receive data register	RSENTnSRXD	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Calibration pulse length register	RSENTnCPL	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Message length register	RSENTnML	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Fast channel receive timestamp register	RSENTnFRTS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Fast channel receive data register	RSENTnFRXD	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Calibration pulse length mirror register	RSENTnCPLM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Message length mirror register	RSENTnMLM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Fast channel receive timestamp mirror register	RSENTnFRTSM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Expanded fast channel receive data register 0	RSENTnEFRD0	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
Expanded fast channel receive data register 1	RSENTnEFRD1	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R

Note 1. Means write restriction exists.

27.4.2 Clock Configuration

27.4.2.1 Timestamp

(1) Timestamp Clock Configuration

RSENT incorporates the timestamp counter.

The minimum required resolution of the timestamp is 1 µs. Depending on the applied communication frequency (clkc), the user should configure the RSENTnTSPC.TTPV bits to achieve the 1-µs resolution. The input frequency is divided by the configured timestamp prescaler value RSENTnTSPC.TTPV.

Depending on the configured tick lengths, the resolution can be decreased by configuring the RSENTnTSPC.TTM bits. The already divided input frequency is divided further by the value of the RSENTnTSPC.TTM bits.

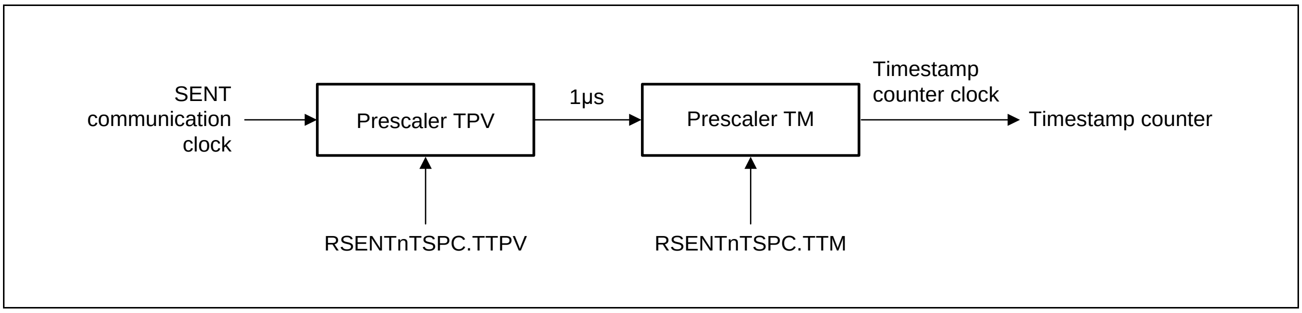


Figure 27.3 Timestamp Counter Clock Generation

(2) Timestamp Counter Operation

The timestamp counter value can be initialized to any value by writing to the RSENTnTSC.TS bits only when the RSENT module is in CONFIGURATION mode.

When timestamp counters are configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU can reset the timestamp counter by writing $0000\ 0000_H$ to the RSENTnTSC.TS bits when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode.

When timestamp counters are configured to operate in slave mode (RSENTnTSPC.TMS = 1), the timestamp counter is cleared when the CPU writes $0000\ 0000_H$ to the RSENTnTSC.TS bits of the channel set in the master when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode. The RSENT module operating in slave mode should have the same timestamp counter prescaler settings as master RSENT module. When timestamp counter synchronization occurs, the internal timestamp counter prescalers are also synchronized.

The current timestamp counter value can be read from the RSENTnTSC.TS bits.

When the RSENT module is in OPERATION ACTIVE mode, each received message is stored with its related timestamp. Timestamp values are taken for fast channel and slow channel data.

The timestamp value is captured when the calibration pulse is detected.

The timestamp value for the fast channel is stored in the RSENTnFRTS.FTS or RSENTnFRTSM.FTSM register field when the fast channel message reception buffer is updated.

The timestamp value for the slow channel is stored in the RSENTnSRTS.STS bits. The timestamp value for the slow channel is identical to the timestamp value of the last fast channel message contributing to the slow channel message.

In case timestamp counter synchronization is required, the following flow should be used.

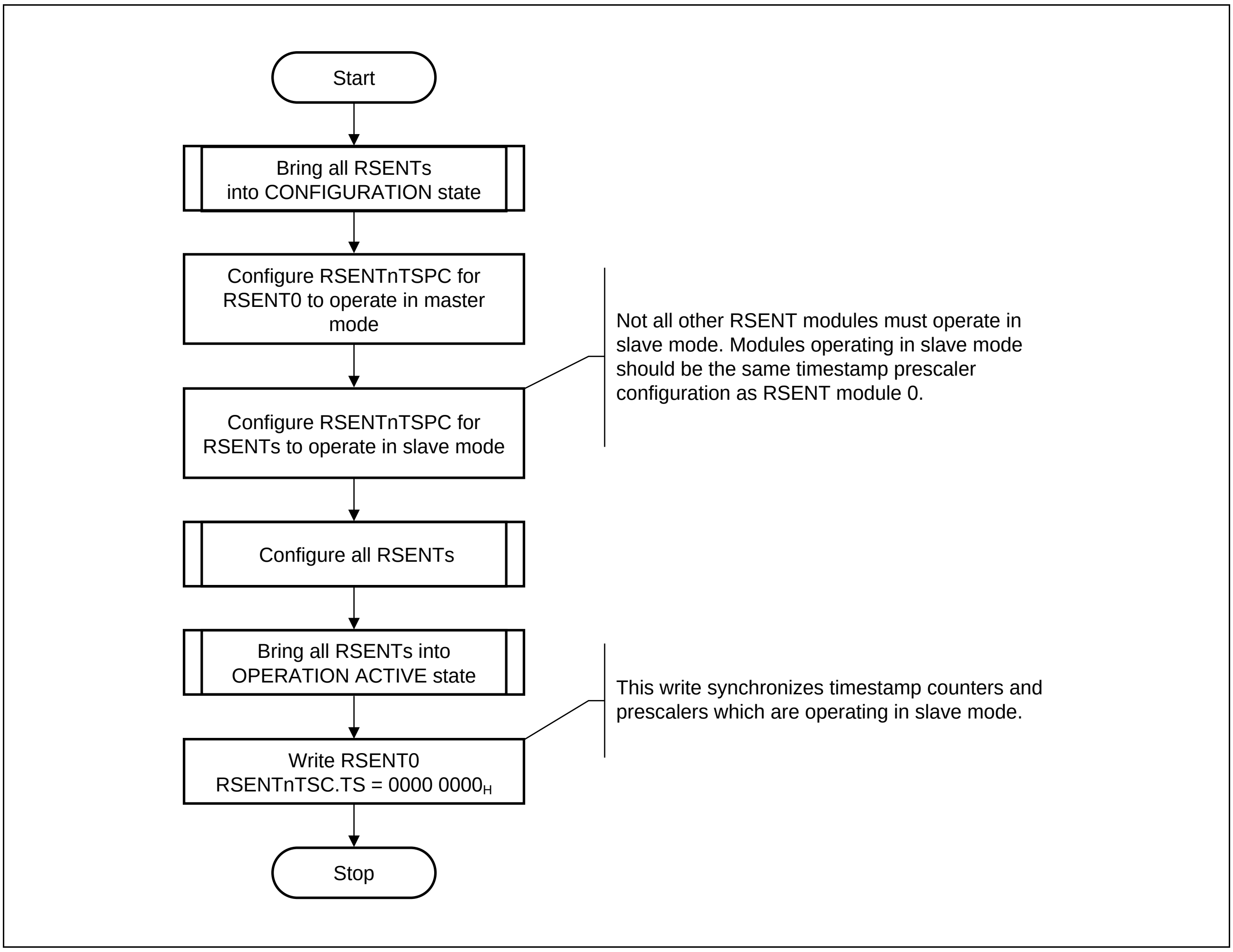


Figure 27.4 Timestamp Counter Synchronization

Further synchronization can be done as long as master RSENT module is in either OPERATION ACTIVE or OPERATION IDLE state.

27.4.2.2 Communication Clock Configuration

(1) RX BRP Setting

Use the formula below to obtain the settings for RSENTnBRP.SCDV and RSENTnBRP.SCMV in accord with $f_{COMMUNICATION}$ to be used (the frequency of the clkc clock) so that the sample clock frequency is 16 MHz (for example, if the settings are for RSENTnBRP.SCDV = 4/RSENTnBRP.SCMV = 6, replace this with RSENTnBRP.SCDV = 2/RSENTnBRP.SCMV = 3). The input clock (clkc clock) is frequency-divided according to the settings of RSENTnBRP.SCDV and RSENTnBRP.SCM to generate the sample clock.

Set $f_{COMMUNICATION}$ (the frequency of the clkc clock) so that it falls within the range from 32 to 40 MHz (or at precisely 16 MHz). Use the formula below to obtain the values for RSENTnBRP.SCDV and RSENTnBRP.SCM such that the sample clock frequency becomes 16 MHz.

$$f_{SAMPLE} = 16\ MHz = f_{COMMUNICATION} \times \frac{Sample\ Clock\ Multiplication\ Value\ (RSENTnBRP.SCMV + 1)}{Sample\ Clock\ Division\ Value\ (RSENTnBRP.SCDV + 1)}$$

$$\begin{aligned} Where\ Sample\ Clock\ Multiplication\ Value\ &= 2\ (RSENTnBRP.SCMV\ =\ 5'd1), \\ Sample\ Clock\ Division\ Value\ &=\ 5\ (RSENTnBRP.SCDV\ =\ 7'd4), \\ f_{COMMUNICATION}\ &=\ 40MHz \\ f_{SAMPLE} &= 40 \times 2/5 = 16MHz \end{aligned}$$

(2) RX and SPC Tick Settings

The used tick length in RX and SPC function can be configured with the RSENTnBRP.TTI and RSENTnBRP.TTF bits. Tick lengths from 1.0 μs to 90.0 μs with a resolution of 0.1 μs can be configured.

The RSENTnBRP.TTI holds the integer part of the tick length and the RSENTnBRP.TTF bits hold the fractional part of the tick length. The tick length is then calculated by:

$$\begin{aligned} &T_{TICK} = T_{RSENTnBRP.TTI} + T_{RSENTnBRP.TTF} \\ &Where\ RSENTnBRP.TTI\ =\ 0, RSENTnBRP.TTF\ =\ 3 \\ &T_{TICK} = 1 + 0.3 = 1.3\ \mu s \end{aligned}$$

27.4.3 RSENT Operation

27.4.3.1 Changing Operation Modes

Once initialization has been completed in CONFIGURATION mode, operation can be enabled by entering OPERATION ACTIVE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION ACTIVE and waiting for the RSENTnMST.OMS to transition to OPERATION ACTIVE.

Once in OPERATION ACTIVE mode the RSENT module begins to receive messages or SPC communication can be started depending on the configuration.

Figure 27.5, Communication Enable Flow shows the communication enabled flow assuming that the RSENT module is in RESET mode:

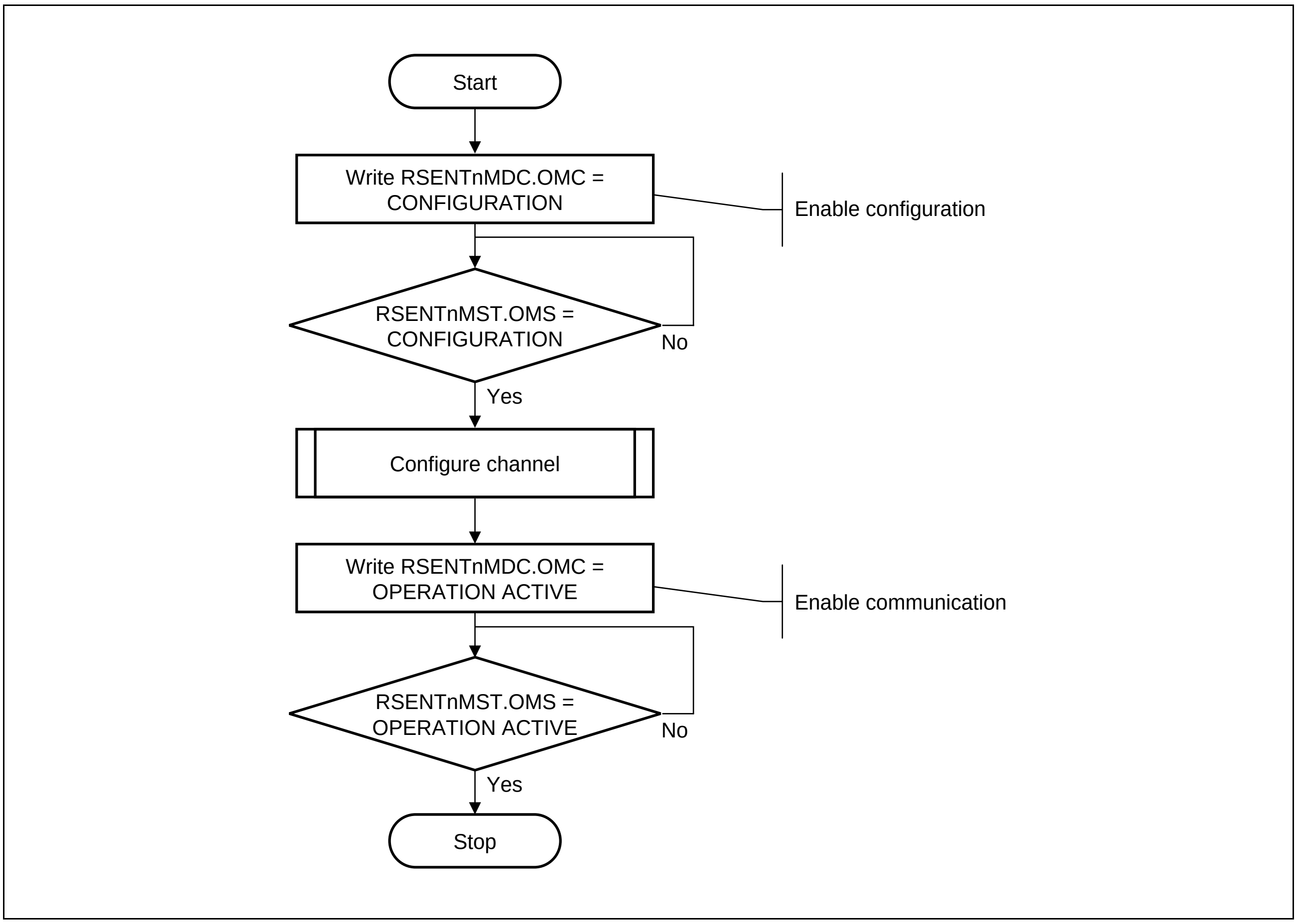


Figure 27.5 Communication Enable Flow

To leave OPERATION ACTIVE mode, communication should be disabled first by entering OPERATION IDLE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION IDLE and waiting for the RSENTnMST.OMS bits to transition to OPERATION IDLE.

However, when the SPC mode is enabled (RSENTnCC.SPCE set to 1) and the SPC trigger transmission has not been requested after the previous SPC communication has been completed (e.g. successful reception for the previous SPC trigger transmission), the RSENT module can directly enter the CONFIGURATION mode.

The transition between OPERATION ACTIVE and OPERATION IDLE depends on the setting of the RSENTnCC.SPCE bit.

(1) RSENTnCC.SPCE = 0

In case a reception is currently ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the receive buffer was updated or error was flagged (see **Section 27.4.3.2(3), Fast Channel Message Reception**).

In case no reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place immediately.

(2) RSENTnCC.SPCE = 1

In case a reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

In case a no response error is flagged, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place at the same time as the error flagging.

The mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the sequence of making a SPC trigger and receiving the response has been completed. This means when a response was already received, the transition takes place immediately. When the response is still pending, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

CONFIGURATION mode can be entered by writing CONFIGURATION to the RSENTnMDC.OMC bits and waiting for the RSENTnMST.OMS to transition to CONFIGURATION.

Once CONFIGURATION mode is entered, the remaining status and message information stored in the RSENT module is lost since status and message information is cleared in CONFIGURATION mode.

Figure 27.6, Communication Disable Flow shows the communication disable flow assuming that the RSENT module is in OPERATION ACTIVE mode.

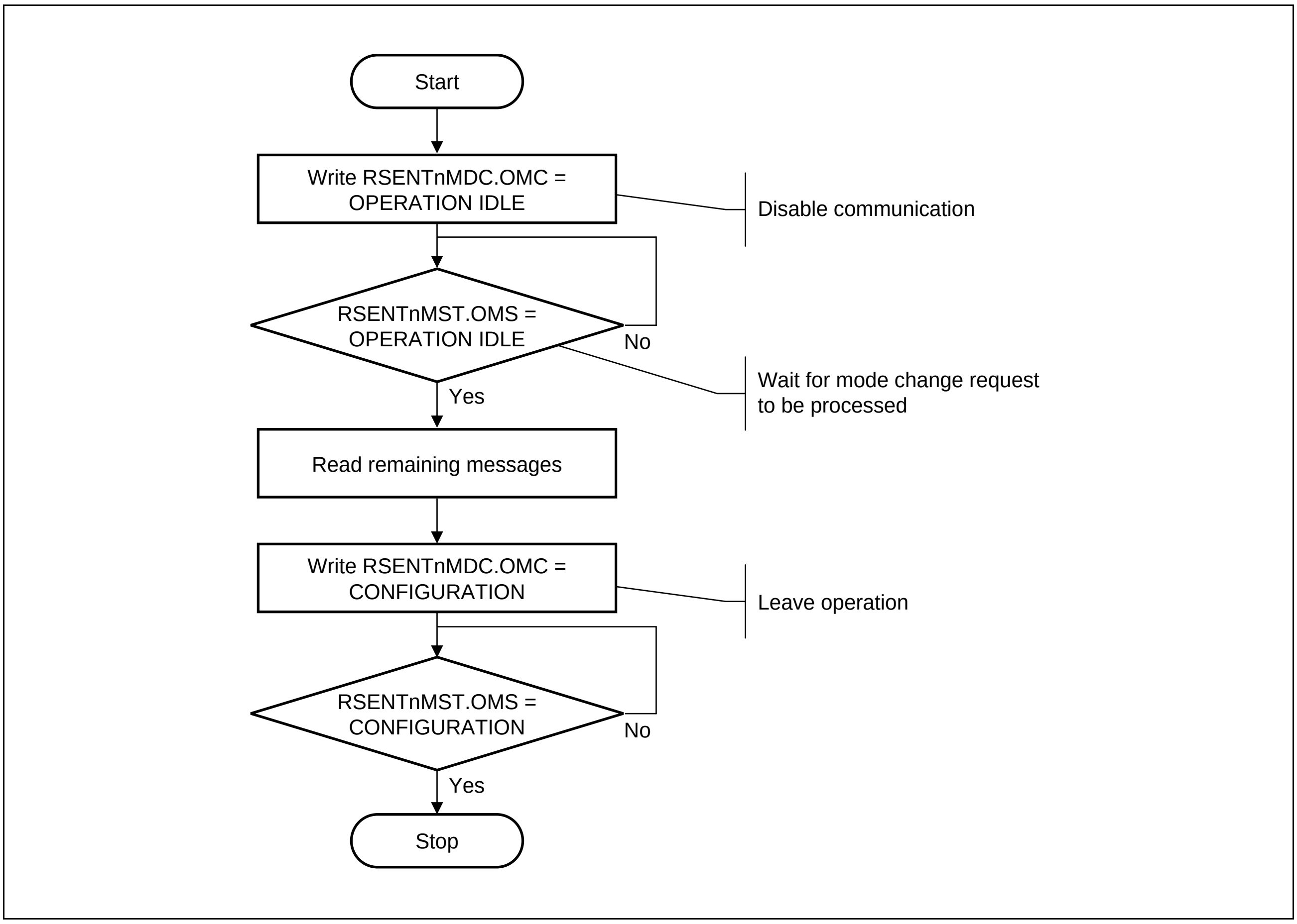


Figure 27.6 Communication Disable Flow

27.4.3.2 Message Reception

SENT message reception is composed of the calibration pulse reception followed by the data nibble pulse reception.

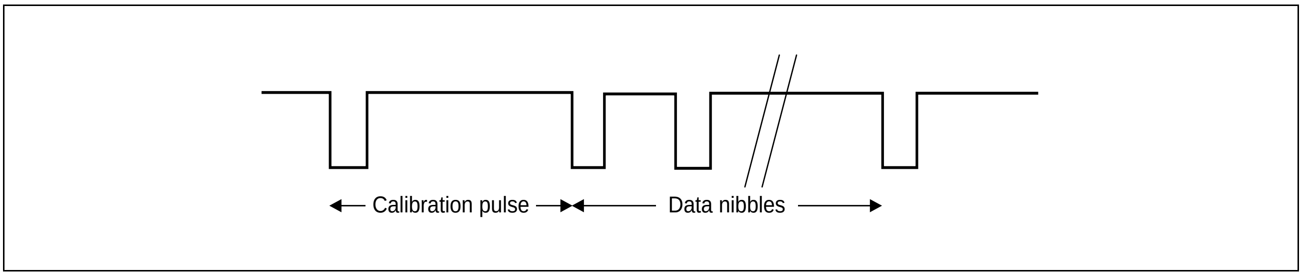


Figure 27.7 SENT Received Message Structure

(1) Calibration Pulse Reception

Within the calibration pulse reception phase the internally generated clock tick is adjusted to the transmit clock speed.

In addition, the calibration pulse is used to end the previous message and perform message diagnostics. The RSENT module supports automatic calibration pulse length diagnostics in variable message length modes (RSENTnCC.PPTC = 0). In case the calibration pulse ratio check fails, the calibration pulse length variation error flag (RSENTnCS.CVS) is set to 1.

(2) Data Nibble Reception

The receive function of the RSENT module is a straightforward capture and compare function. The RSENT module receives sensor information encoded by the temporal distance of two consecutive falling edges on the data line. The temporal distance (the Number of Clock Ticks) is captured and compared against a set of values to determine the actual nibble value. The data encoding is illustrated in **Table 27.33, Data Nibble Encoding** below.

Table 27.33 Data Nibble Encoding

Nibble Period (Number of Clock Ticks)	Nibble Value (Binary)
12	0000_B
13	0001_B
14	0010_B
15	0011_B
16	0100_B
17	0101_B
18	0110_B
19	0111_B
20	1000_B
21	1001_B
22	1010_B
23	1011_B
24	1100_B
25	1101_B
26	1110_B
27	1111_B

The received data nibbles are composed into an SENT message which is then stored in the fast channel message reception buffer.

Any other received nibble period during the reception of data nibbles will cause a fast channel nibble encoding error.

(3) Fast Channel Message Reception

Messages received on the fast message channel are stored in a receive buffer.

A fast channel message reception buffer is composed of the calibration pulse length register (RSENTnCPL), the message length register (RSENTnML), the fast channel receive timestamp register (RSENTnFRTS), and the fast channel receive data register (RSENTnFRXD).

All registers are placed on successive addresses that allows transferring the register content into memory using DMA.

The RSENT module is equipped with a double receive buffer structure that allows the storage of two complete SENT messages including the related timestamp and message length information. Message decoding and assembling are done in a separate register stage.

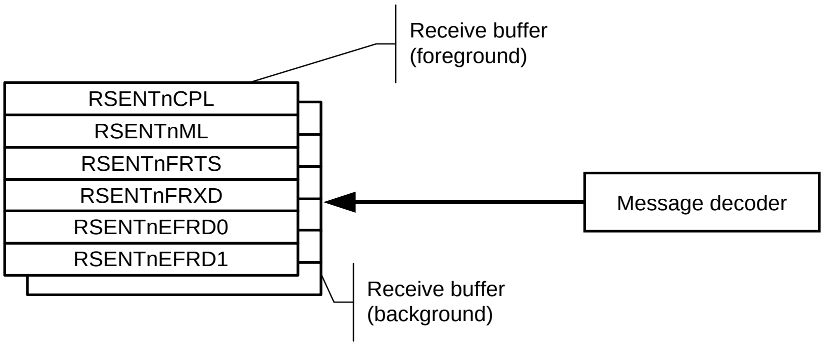


Figure 27.8 Fast Channel Message Reception Buffer

The first received message is placed into the message buffer that can be accessed by the CPU. This buffer (except the RSENTnFRXD.SNDM and RSENTnEFRD0.SNDM bit) is not updated any more until the RSENTnFRXD or RSENTnEFRD1 register was read.

When a new message is placed into a receive buffer, the RSENTnFRXD.FND and RSENTnEFRD0.FND bit is set. At the same time, the RSENTnCS.FRS bit is set and, if enabled, a receive interrupt request is generated.

When the foreground receive buffer is holding an unprocessed message (the RSENTnFRXD.FND and RSENTnEFRD0.FND bit is 1), any further incoming message is placed in the background buffer. The background buffer is updated with any further incoming messages. In case an unprocessed background message buffer message is overwritten, the RSENTnCS.FMS bit is set to 1.

When the CPU reads the RSENTnFRXD or RSENTnEFRD1 register and there is valid data in the background buffer, the data previously located in the background buffer becomes available in the receive buffer and is accessible by the CPU. If enabled, a new interrupt request for fast channel data is generated and RSENTnCS.FRS is set.

When the RSENTnFRXD.FND and RSENTnEFRD0.FND/RSENTnCS.FRS bit is not set, the data in the receive buffer is not defined and the CPU should not access the receive buffer.

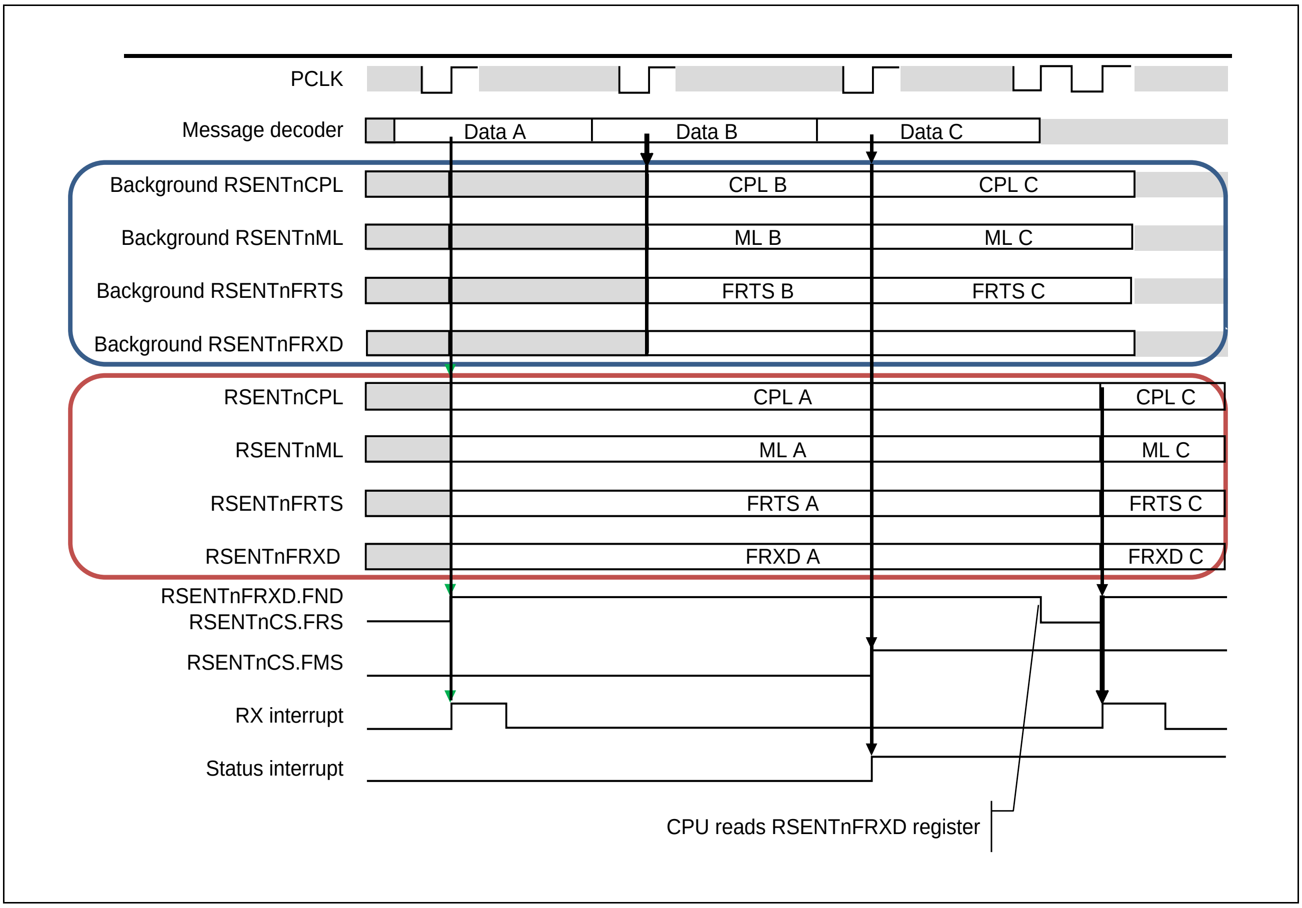


Figure 27.9 Fast Channel Message Reception Buffer Update Timing

The update timing of the receive buffer depends on the applied configuration as depicted in **Figure 27.10, Buffer Update in Variable Message Length Mode and Preferred Check Method** to **Figure 27.13, Buffer Update in SPC Mode**.

The RSENTnFRTS register is updated with the current timestamp counter register value when the calibration pulse is detected.

The data alignment in the RSENTnFRXD register depends on the nibble data count (RSENTnCC.NDN).

Table 27.34 Data Nibble Alignment in RSENTnFRXD Register

RSENTnFRXD RSENTnCC.NDN	23:20	19:16	15:12	11:8	7:4	3:0
000_B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001_B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010_B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011_B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100_B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101_B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6

Table 27.35 Data Nibble Alignment in RSENTnEFRD1 Register

RSENTnEFRD1 RSENTnCC.NDN	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
000_B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001_B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010_B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011_B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100_B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101_B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6
110_B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7
111_B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7	Nibble 8

(a) SAE Operation with Variable Message Length and Preferred Check Method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 0)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the preferred option in the J2716 2016 specification. In this mode, message diagnostics is done after the calibration pulse was received following a message.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

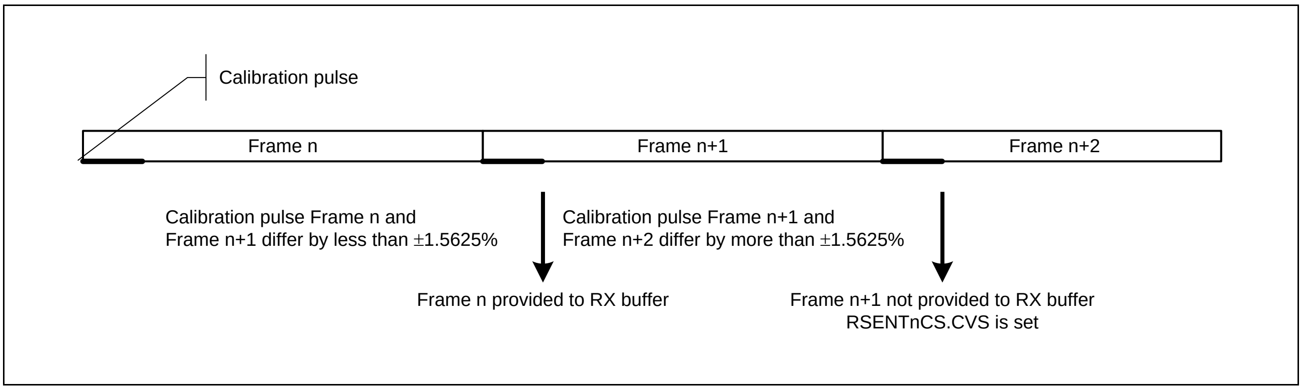


Figure 27.10 Buffer Update in Variable Message Length Mode and Preferred Check Method

(b) SAE Operation with Variable Message Length and Optional Check Method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 1)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the optional frame check method as described in the J2716 2016 specification. In this mode, the calibration pulse of the current frame is compared to the calibration pulse of the last valid preceding frame.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

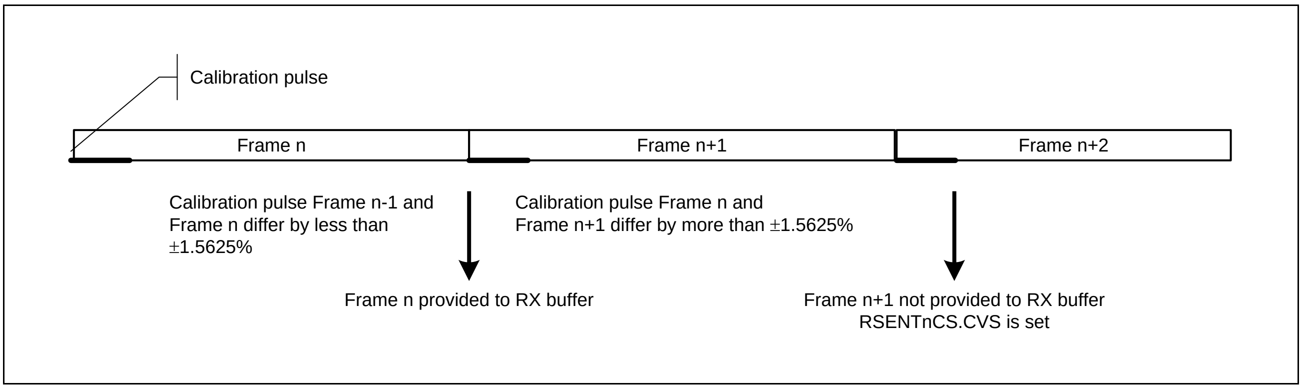


Figure 27.11 Buffer Update in Variable Message Length Mode and Optional Check Method

(c) SAE Operation with Fixed Message Length (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 1)

In this mode, the RSENT module does not perform the check for calibration pulse and message length ratio according to the preferred option in the J2716 2016 specification. In this mode, the RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples.

The message buffer is updated at the beginning of the following calibration pulse irrespective of the values in the RSENTnCPL and RSENTnML registers. The CPU needs to calculate the ratio and either accept or discard the message.

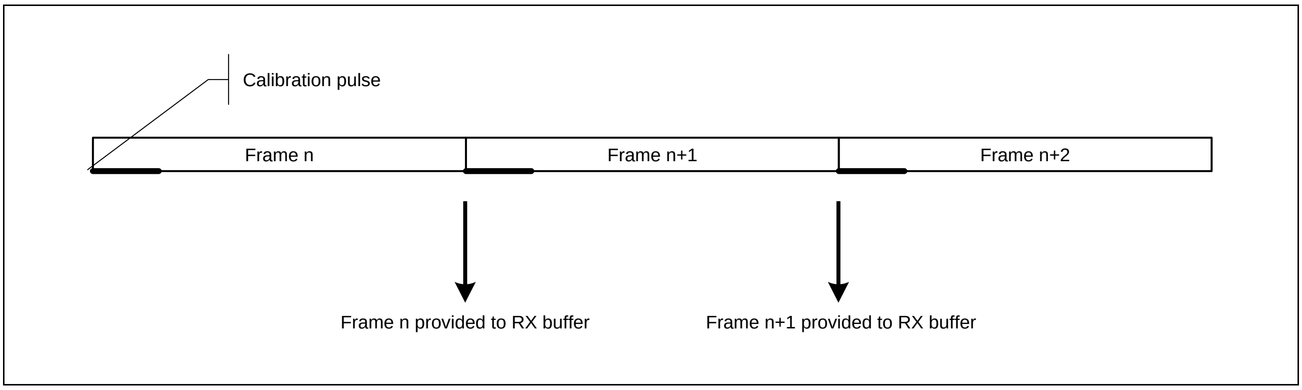


Figure 27.12 Buffer Update in Fixed Message Length Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

(d) SPC Operation (RSENTnCC.SPCE = 1)

In this operation mode, sensor data transmission is done following a SPC master trigger pulse. Within SAE SENT communication, the calibration pulse or pause pulse is terminating the previous message. In SPC communication, the sensor is only sending data following a SPC trigger request. An end pulse sent by the sensor is terminating the message. The message buffer is updated at the beginning of the end pulse.

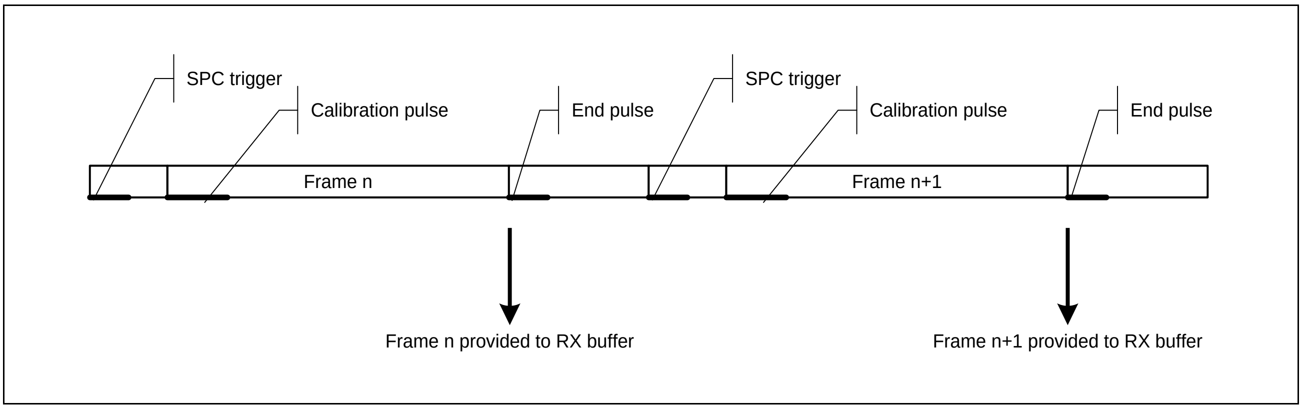


Figure 27.13 Buffer Update in SPC Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

The RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples. The CPU needs to calculate the ratio of calibration pulses and/or message length and either accept or discard the message.

In case of variable message length mode, the RSENT module cannot perform this check because the receive timing of the next calibration pulse depends on the next SPC trigger timing.

(4) Fast Channel Reception Flow

In **Figure 27.14, Fast Channel Reception Flow**, the recommended reception flow for the fast channel message reception buffer is shown.

When using a polling or event driven method, the CPU should only read the setting of the RSENTnCS.FRS bit to check the availability of new fast channel data.

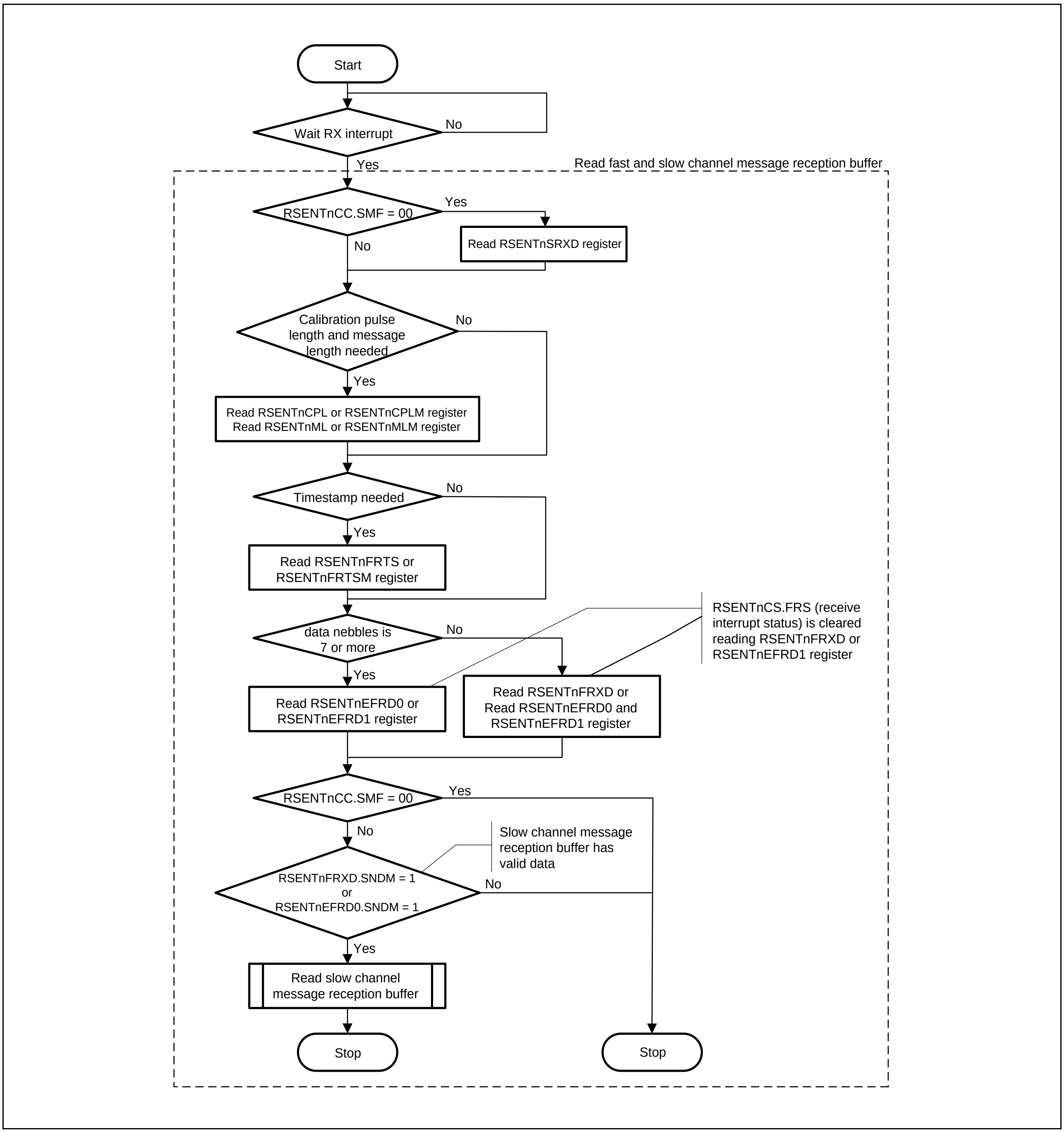


Figure 27.14 Fast Channel Reception Flow

In any case, the CPU should keep the order in reading the receive buffer registers as shown in the flow. The RSENTnFRXD or RSENTnEFRD0 register should be the last register to be accessed.

The handling of the slow channel message reception buffer is described in **Section 27.4.3.2(6), Slow Channel Reception Flow**.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

(5) Slow Channel Message Reception

The RSENT module supports extraction of the slow message out of the fast channel messages by using the bits 3 and 2 out of the status and communication nibble. In order to enable the slow channel extraction, the CPU should set the RSENTnCC.SMF bits to the expected serial message format.

When no serial message extraction is selected (RSENTnCC.SMF = 00_B), the RSENTnSRXD register becomes part of the fast channel message reception buffer structure (including background buffer) and RSENTnSRTS register should be ignored. The status and communications nibbles is placed in the RSENTnSRXD.IDD bits. Furthermore no slow channel new data and slow channel message lost flags are generated.

In order to receive the slow channel serial message, all fast channel serial messages contributing to a slow channel serial message must be received successfully and the received slow channel serial message must comply with the selected serial message format.

A message lost on the fast channel does not impact the reception on the slow channel.

A slow channel message reception buffer is composed of the slow channel receive timestamp register (RSENTnSRTS) and the slow channel receive data register (RSENTnSRXD).

In opposite to the fast channel message reception buffer, the slow channel message reception buffer does not support a double receive buffer structure; only a single receive buffer structure is available.

Message decoding and assembling is done in a separate register stage.

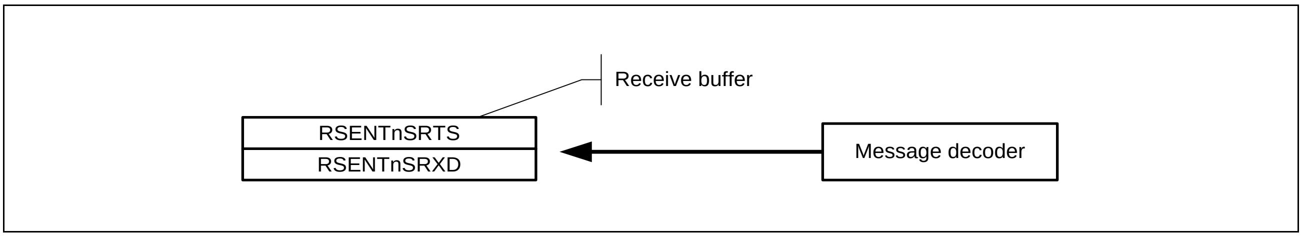


Figure 27.15 Slow Channel Message Reception Buffer

The slow channel message reception buffer is updated at the same time as the fast channel message reception buffer that holds the last status and communication nibble required for the slow channel message. The RSENTnSRXD.SND bit is set to 1 at the same time.

Further updates to the buffer are not carried out until after the RSENTnSRXD.SND bit has been read.

When the receive buffer is holding an unprocessed message (RSENTnSRXD.SND is 1), any further incoming message is lost (the slow channel message reception buffer is not updated) and RSENTnCS.SMS is set to 1.

When the CPU reads the RSENTnSRXD register, RSENTnSRXD.SND is automatically cleared.

The RSENTnSRTS register is updated with the current timestamp counter register value of the last frame contributing to the slow channel message.

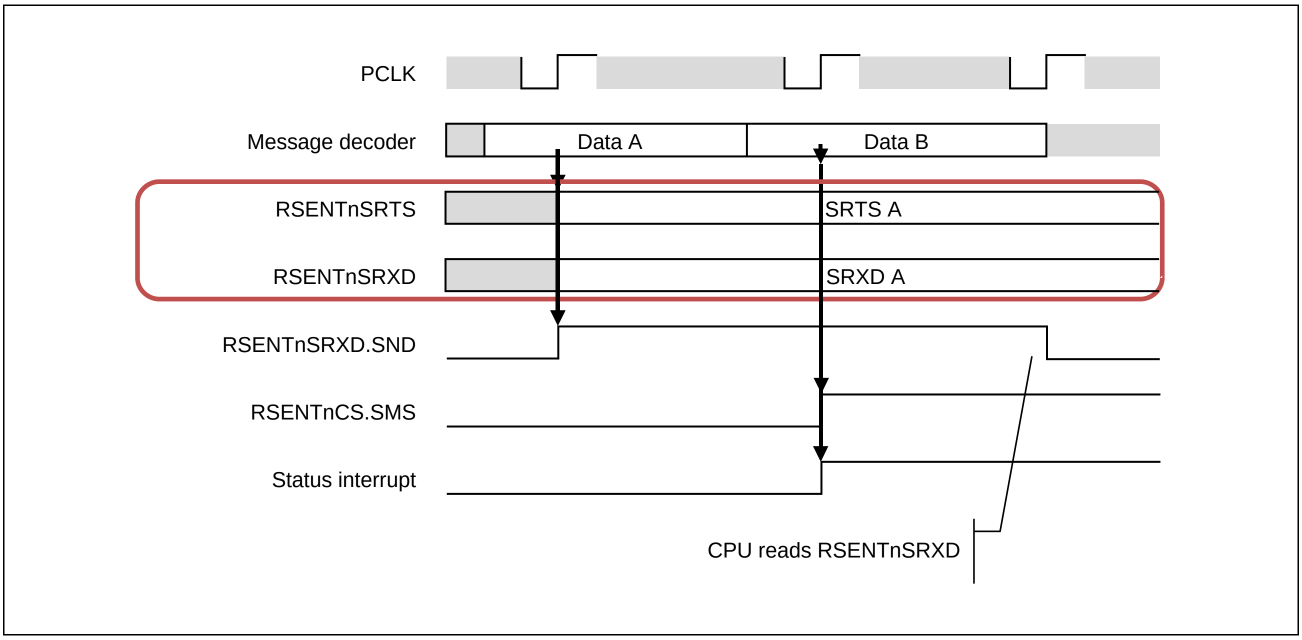


Figure 27.16 Slow Channel Message Reception Buffer Update Timing

The data alignment in the RSENTnSRXD register depends on the slow channel message format (RSENTnCC.SMF) and the received configuration bit.

Table 27.36 Data Alignment in RSENTnSRXD Register

RSENTn CC.SMF	RSENTnSRXD. SMGC	RSENTnSRXD. IDD[19:16]	RSENTnSRXD. IDD[15:12]	RSENTnSRXD. IDD[11:8]	RSENTnSRXD. IDD[7:4]	RSENTnSRXD. IDD[3:0]
00_B	Undefined	Undefined	Undefined	Undefined	Undefined	Status and communication nibble
01_B	Undefined	Undefined	Undefined	Message ID[3:0]	DATA[7:4]	DATA[3:0]
10_B	0	Message ID[7:4]	Message ID[3:0]	DATA[11:8]	DATA[7:4]	DATA[3:0]
10_B	1	Message ID[3:0]	DATA[15:12]	DATA[11:8]	DATA[7:4]	DATA[3:0]

(6) Slow Channel Reception Flow

In **Figure 27.17, Slow Channel Reception Flow**, the recommended reception flow for the slow channel message reception buffer is shown. When the slow channel receive data is required, this process should be executed as part of the fast channel reception flow.

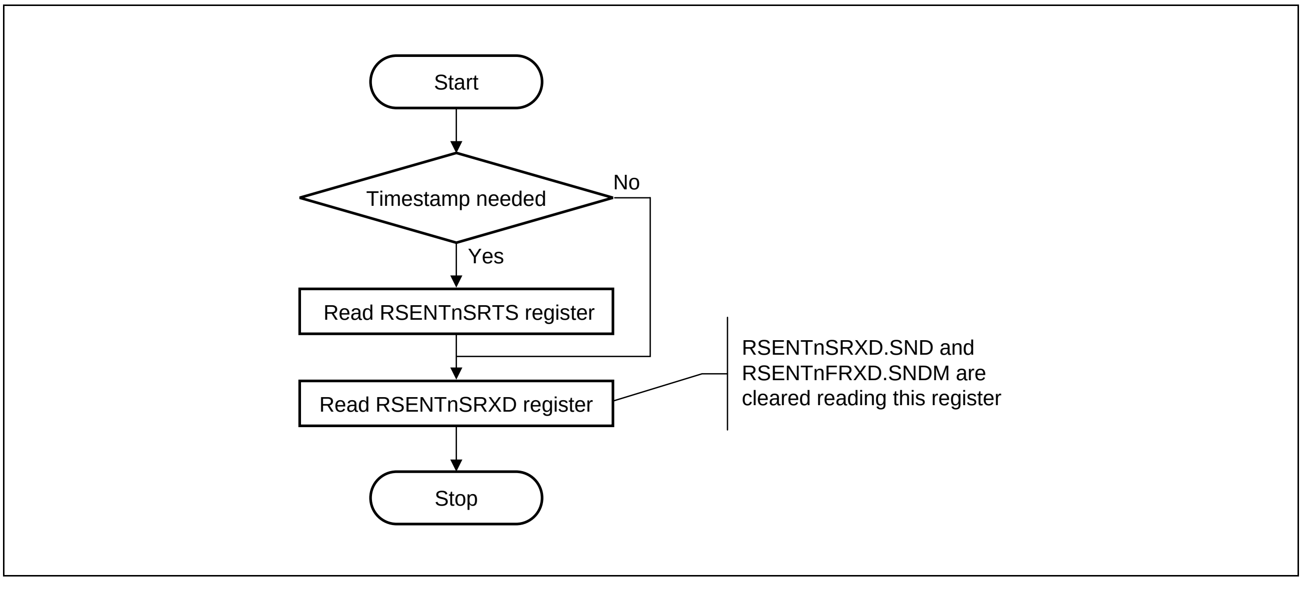


Figure 27.17 Slow Channel Reception Flow

In any case, the CPU should keep the order in reading the slow channel message reception buffer registers as shown in the flow. The RSENTnSRXD.SND bit should be accessed as last.

(7) DMA Flow

In case of DMA usage, the start address for the DMA usage and the number of transfers define which part of the receive buffer will be transferred. The RSENTnFRXD or RSENTnEFRD1 register should be the last register to be accessed using a 32 bit access method.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

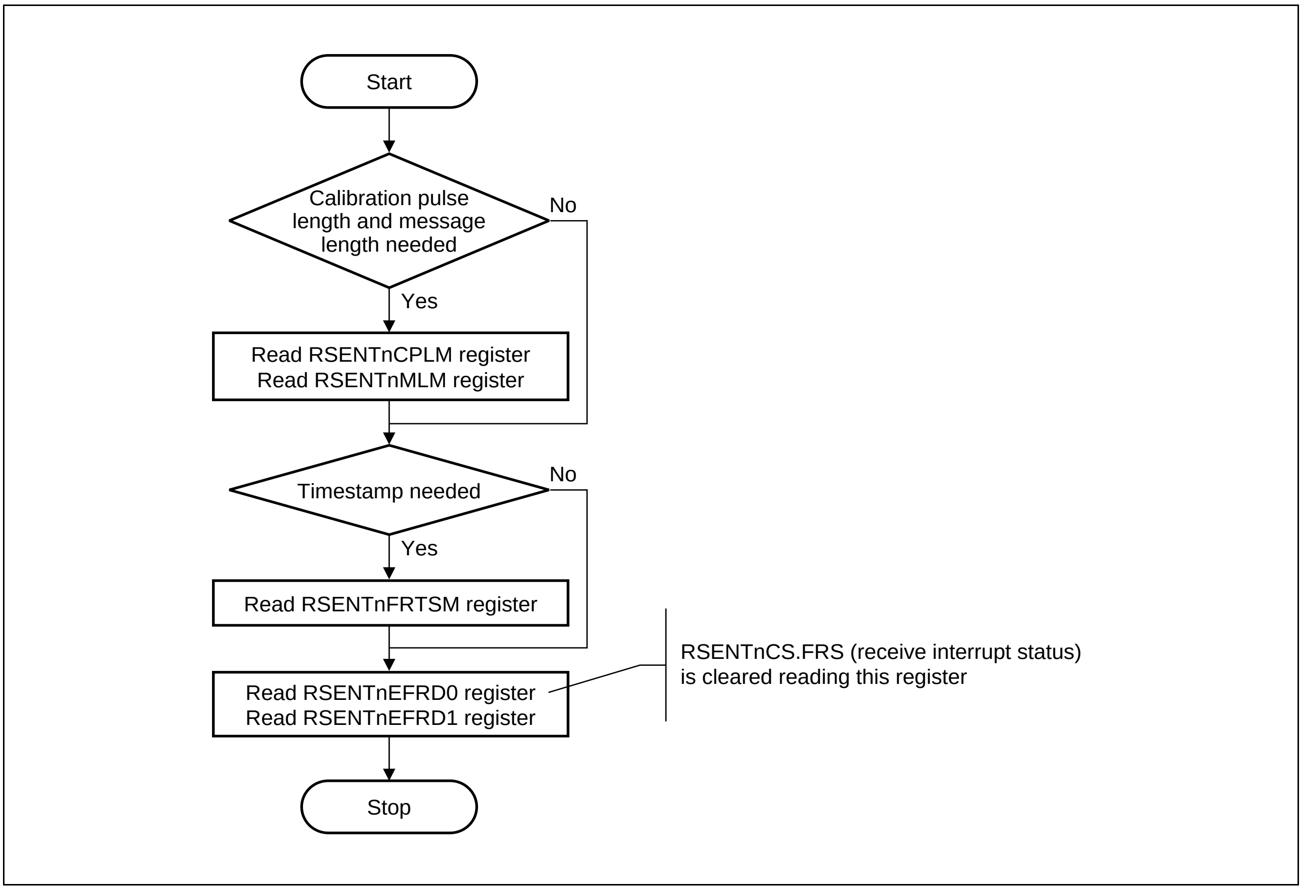


Figure 27.18 DMA Reception Flow

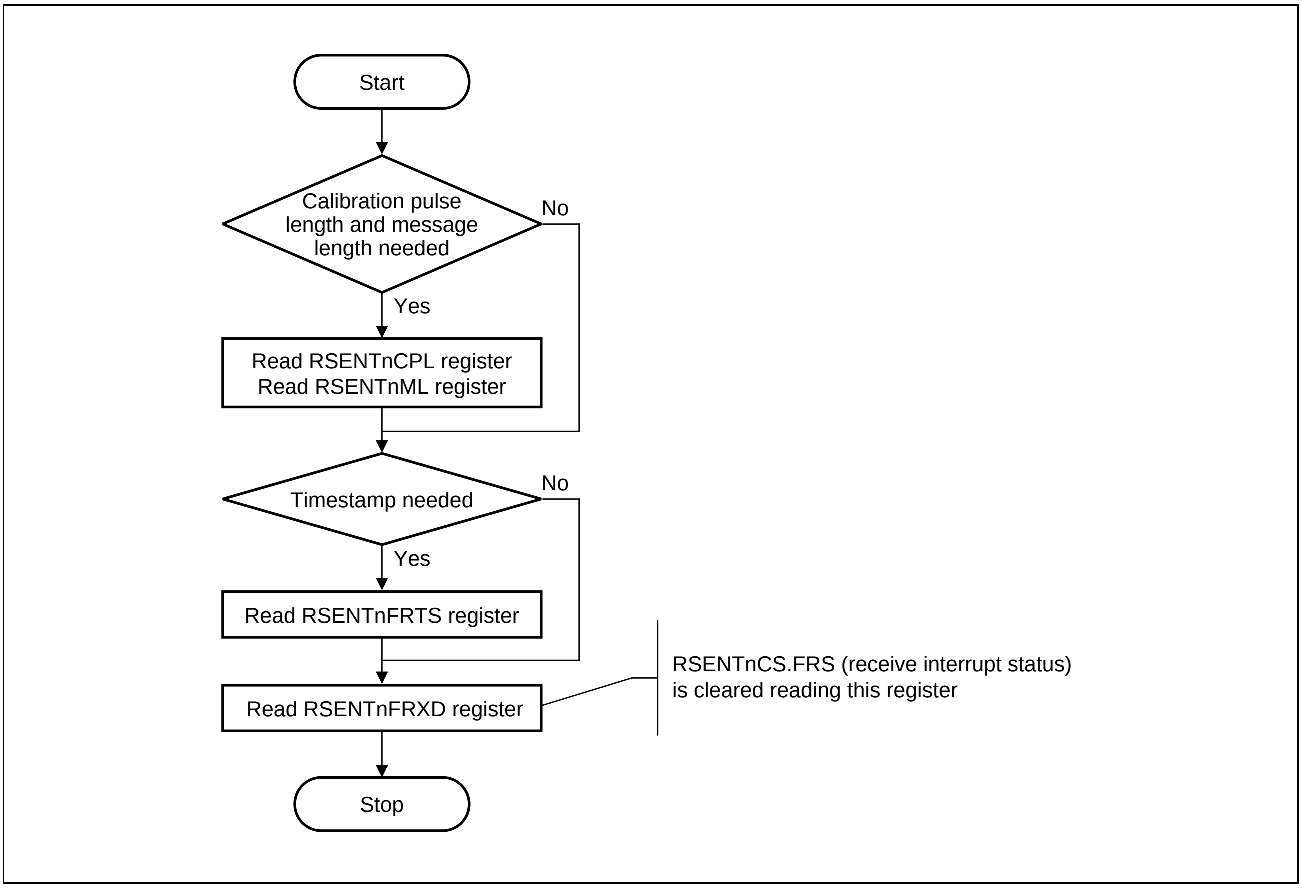


Figure 27.19 DMA Reception Flow (Data nibbles are 6 or less)

In the software processing, when the transferred data set, the CPU should check the status of the transferred RSENTnFRXD.SNDM bit. If this bit is set to 1, then the user needs to read the slow channel message reception buffer if needed.

(8) Error Flagging

The message lost error flag (RSENTnCS.SMS or RSENTnCS.FMS) is set when a new message's diagnostics pass before the previous message is read.

The SPC communication error (shown in RSENTnCS.NRS) is flagged when the CPU writes to RSENTnSPCT.TLL before/during response reception.

The timing at which the fast channel reception error flags (RSENTnCS.CVS, RSENTnCS.CLS, RSENTnCS.FNS, RSENTnCS.FES, and RSENTnCS.FCS) and the slow channel reception error flags (RSENTnCS.SCS, and RSENTnCS.SES) are updated varies with the setting of bits in the communications configuration register (RSENTnCC.SPCE, RSENTnCC.FCM, RSENTnCC.PPC, and RSENTnCC.PPTC).

Table 27.37, Timing to Set Error Flag (RSENTnCC.SPCE = 0) and **Table 27.38, Timing to Set Error Flag (RSENTnCC.SPCE = 1)** list the timings with which the error flags corresponding to each setting are updated.

In case a nibble encoding error or calibration pulse length error is detected, message reception is terminated immediately. No further error flagging for this message is done. Message decoding starts again after a calibration pulse without calibration length error (RSENTnCS.CLS) is detected.

Table 27.37 Timing to Set Error Flag (RSENTnCC.SPCE = 0)

RSENTnCC.SPCE	0							
RSENTnCC.FCM	0				1			
RSENTnCC.PPC	0		1		0		1	
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	EC	X	EC	IM	IM	X	IM	IM
RSENTnCS.FES	EC	X	EC	IM	IM	X	IM	IM
RSENTnCS.FNS	EC	X	EC	—	—	X	—	—
RSENTnCS.SCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.SES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CLS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CVS	EC	X	EC	—	EC	X	EC	—

Note: EC: End of calibration pulse (on the falling edge of a valid calibration pulse)
IM: Immediately upon detection (immediately when the error is detected)
—: Error flag not detected
X: Setting prohibited

NOTE

When RSENTnCC.PPC and RSENTnCC.PPTC are 1, the mode is pause pulse with fixed message length.

When this mode, this diagnostic can be used as the receiver does not need to wait for the next calibration pulse to diagnose the current received frame, so RSENTnCS.FNS and RSENTnCS.CVS are not set.

Table 27.38 Timing to Set Error Flag (RSENTnCC.SPCE = 1)

RSENTnCC.SPCE	1							
RSENTnCC.FCM	0				1			
RSENTnCC.PPC	0		1		0		1	
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.FES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.FNS	—	X	—	—	—	X	—	—
RSENTnCS.SCS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.SES	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CLS	IM	X	IM	IM	IM	X	IM	IM
RSENTnCS.CVS	—	X	—	—	—	X	—	—

Note: IM: Immediately upon detection (immediately when the error is detected)
—: Error flag not detected
X: Setting prohibited

NOTE

In case the sensor stops communication, no buffer update or status update for last message takes place. The SW should take care of this by timeout checks.

SPC mode is a communication of only one frame that is start the SPC trigger to the starting point. The Error flags are immediately set when they are detected. So RSENTnCS.FNE and RSENTnCS.CVS are not set.

When a transition to OPERATION IDLE is configured in RSENTnMDC.OMC and an error for the calibration or the fast channel reception is detected in the message in which the mode transition was requested, the error is not flagged and the message is aborted.

In case of a fast channel encoding error or a calibration pulse length error the OPERATION IDLE mode is entered immediately.

In case of a fast channel nibble count error, fast channel CRC error, or fast channel calibration pulse variation error the OPERATION IDLE state is entered at the end of the next status and communications nibble.

RERSENTnCS.FNS is only set after a valid calibration pulse was detected and all following nibbles have a valid length (≥ 12 ticks and ≤ 27 ticks) or no nibble was received between two valid calibration pulses.

RSENTnCS.FES is only set if the nibble with an encoding error occurred in the communication and status nibble, CRC nibble or in one of the expected data nibbles.

If SPC is enabled (RSENTnCC.SPCE = 1), RSENTnCS.CLS is set if a calibration pulse was expected but the pulse length does not meet the calibration pulse range. If SPC is disabled (RSENTnCC.SPCE = 0), RSENTnCS.CLS is set only after a valid calibration pulse has been received and a calibration pulse was expected but the pulse length does not meet the calibration pulse range.

During re-synchronization additional error flags might be set which is not affecting the reception of the following frame.

27.4.4 SPC Function

The RSENT module supports an extension of the J2716 specification known as SPC, whereby the RSENT module can pull down the RX line to initiate SENT message transmission.

The user can configure the polarity of the sent_spc port.

The text below describes the behavior of the sent_spc port with the default settings of RSENTnCC.SOPC. When the value of RSENTnCC.SOPC after a reset is changed, the sent_spc port operates with inverted polarity.

The user can enable or disable the SPC extension by setting the RSENTnCC.SPCE bit. When the RSENTnCC.SPCE bit is set to 0, SPC is disabled. The sent_spc port is driven low by the RSENT module, allowing normal SENT reception to take place. When the RSENTnCC.SPCE bit is set to 1, SPC is enabled and the sent_spc port can be driven high by the RSENT module to request a frame transmission by the sensor.

The transmission function of the RSENT module is a straightforward PWM function. The purpose of this function is to communicate in direction to the sensor by the output sent_spc. With the sent_spc output, the RSENT module can pull down the signal line by an external transistor.

The signal line will be held low for a configured length of tick time specified in the RSENTnSPCT.TLL bits.

The Tick time is configured with the RSENTnBRP.TTI bits and the RSENTnBRP.TTF bits which are equal to the transmission tick time. For details, see **Section 27.4.2.2(2), RX and SPC Tick Settings**.

In a single sensor system, this function can be used to trigger data transmission from the sensor. Further data can be sent to the sensor by varying the trigger pulse length. In a multi sensor system, this function can be used to address a dedicated sensor and request a data transmission.

Once RSENT SPC initialization is complete, transmission can be triggered by writing the trigger pulse width to the RSENTnSPCT.TLL register. When a transmission is triggered, the trigger pulse with the configured length is sent. Then a frame reception is expected. After frame reception was done, a new trigger pulse can be sent.

Writing to RSENTnSPCT.TLL requests a SPC trigger transmission. After writing to RSENTnSPCT.TLL, the CPU should read RSENTnCS.NRS to check whether the previous request was completed or not.

In case RSENTnCS.NRS is set, no SPC trigger is sent and any potentially ongoing reception at this time is aborted. The CPU should clear RSENTnCS.NRS by writing 1 to RSENTnCSC.NRC. The CPU can write again to RSENTnSPCT.TLL to request a SPC trigger transmission.

In case RSENTnCS.NRS is not set, the CPU should set a reception timeout counter in software. If a reception occurs before the timeout counter elapses, the user should process the received slow and fast channel data as shown in the fast channel reception flow (**Figure 27.14, Fast Channel Reception Flow**) and slow channel reception flow (**Figure 27.17, Slow Channel Reception Flow**).

When the timeout counter elapses without any successful reception, the addressed sensor seems not to send any valid response. The CPU should analyze the RSENTnCS register to analyze the reason for no successful reception. A new request can be made considering that when RSENTnCS.NRS gets set no SPC trigger is sent.

Purpose of the timeout function is to define a timeout window for response reception in software.

Figure 27.20, Transmission Flow shows a transmission flow with a timeout function implemented in software. The timeout function is optional and can be omitted if not needed.

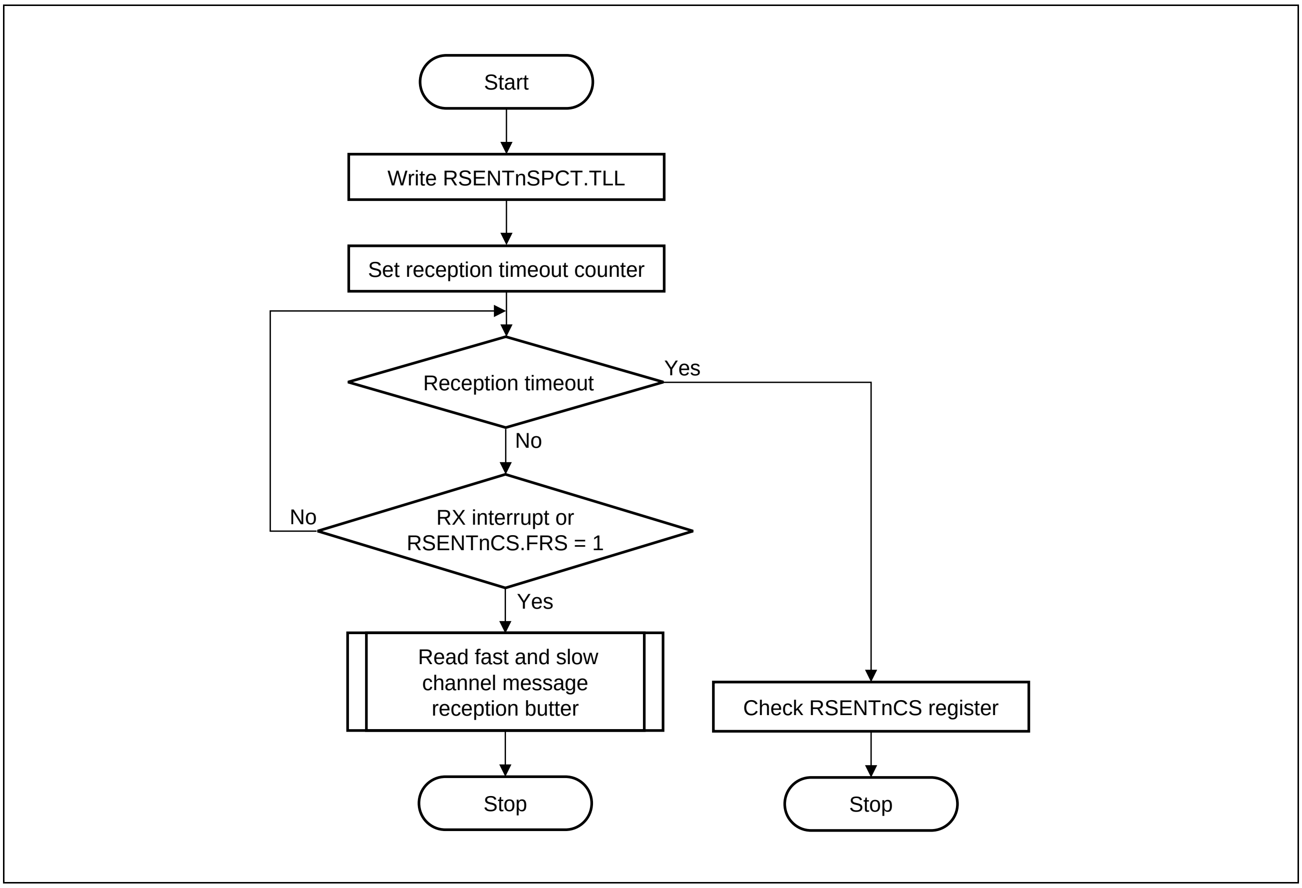


Figure 27.20 Transmission Flow

27.4.5 Interrupts and Checks

The RSENT module provides two interrupt lines.

The successful fast channel receive interrupt notifies the CPU that the fast channel message reception buffer was updated and is holding a set of valid received data. Also, the reception status bit is set (RSENTnCS.FRS).

The status interrupt notifies the CPU that at least one of the error flags or message lost flags in the RSENTnCS register is set.

Whether a status flag in the RSENTnCS register is contributing to the generation of an interrupt event or not can be set individually.

The execution of the CRC checks can be disabled for the slow channel and fast channel individually. In case a check is disabled, the CRC of the received message is not checked and the related error flag is never set.

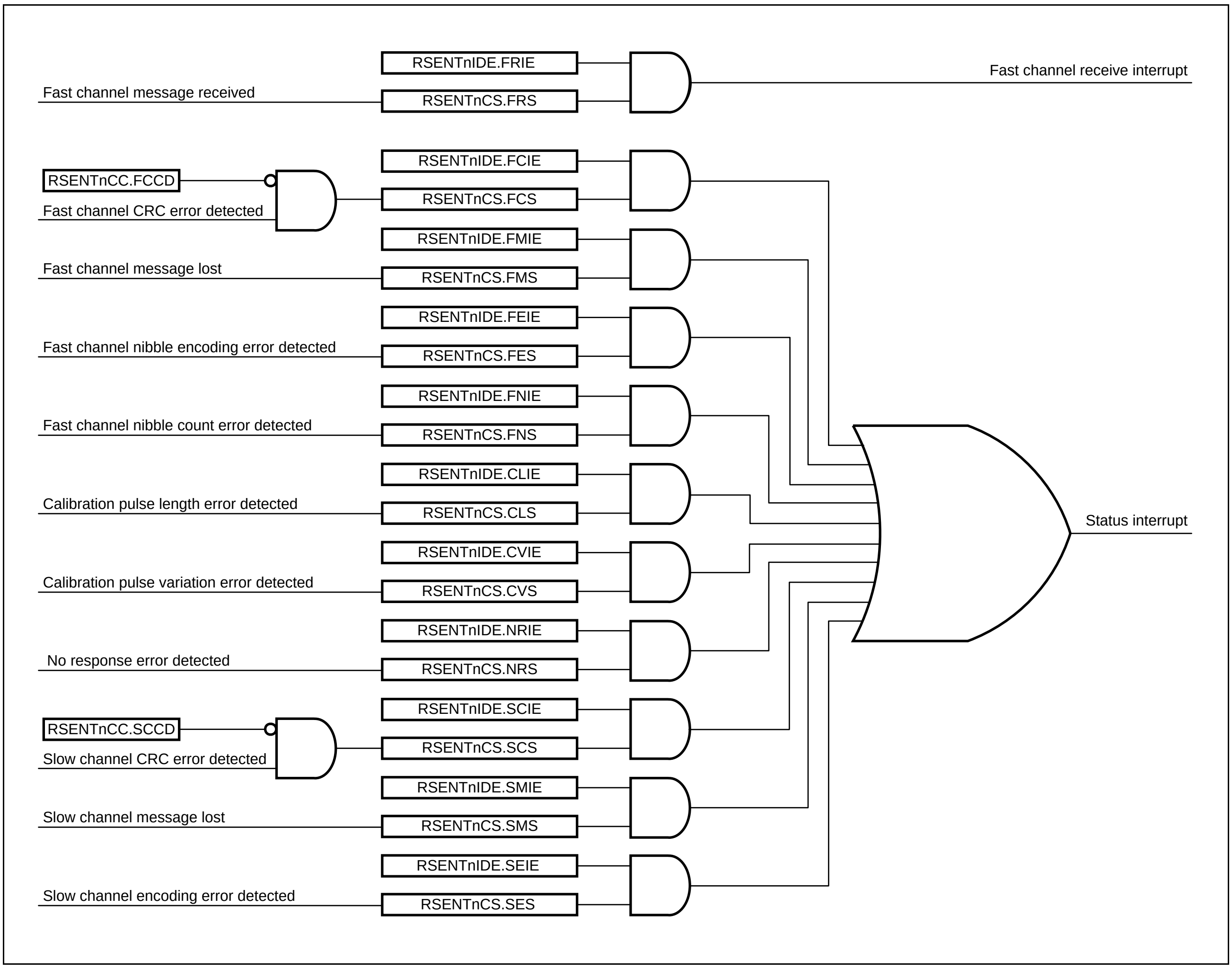


Figure 27.21 Interrupt Structure

Table 27.39, Status Flag Influence to Receive Buffer Behavior gives an overview about the relationship between set status flags and the buffer update.

Table 27.39 Status Flag Influence to Receive Buffer Behavior

RSENTnCS	Fast Channel Message Reception Buffer	Slow Channel Message Reception Buffer
FRS	Updated	Updated if all status and communications nibbles of slow channel messages are received and RSENTnCS.SES = 0 and RSENTnCS.SCS = 0
FCS	Not updated	Receive process aborted. Search for new start condition
FMS	Message lost	Not impacted
FES	Not updated	Receive process aborted. Search for new start condition
FNS	Not updated	Receive process aborted. Search for new start condition
CLS	Not updated	Receive process aborted. Search for new start condition
CVS	Not updated	Receive process aborted. Search for new start condition
NRS	Not updated	Receive process aborted. Search for new start condition
SCS	Not impacted	Not updated
SMS	Not impacted	Message lost
SES	Not impacted	Receive process aborted. Search for new start condition

Section 28 Reserved

Section 29 Window Watchdog Timer (WDTA)

This section contains a generic description of the Window Watchdog Timer (WDTA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of WDTA.

29.1 Features of RH850/F1KH, RH850/F1KM WDTA

29.1.1 Number of Units

This microcontroller has the following number of WDTA units.

Each WDTA unit has single channel interface.

Table 29.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	3	3	3
Name	WDTAn (n = 0 to 2)	WDTAn (n = 0 to 2)	WDTAn (n = 0 to 2)

Table 29.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	2	2	2	2	2
Name	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)

Table 29.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	2	2	2	2
Name	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)	WDTAn (n = 0, 1)

Table 29.4 Index (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual window watchdog timer units are identified by the index “n”: for example, WDTAnWDTE (n = 0 to 2) is the WDTAn enable register.

Table 29.5 Index (RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual window watchdog timer units are identified by the index “n”: for example, WDTAnWDTE (n = 0, 1) is the WDTAn enable register.

29.1.2 Register Base Addresses

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses.

Table 29.6 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<WDTA0_base>	FFED 0000_H
<WDTA1_base>	FFED 1000_H
<WDTA2_base>	FFED 2000_H

Table 29.7 Register Base Addresses (RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<WDTA0_base>	FFED 0000_H
<WDTA1_base>	FFED 1000_H

29.1.3 Clock Supply

The WDTAn clock supply is shown in the following table.

Table 29.8 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name	Description
WDTA0	WDTATCKI	CKSCLK_AWDTA	Timer count clock
	Register access clock	CPUCLK_L	Bus clock
WDTA1	WDTATCKI	LS IntOSC f_{RL}	Timer count clock
	Register access clock	CPUCLK_L	Bus clock
WDTA2	WDTATCKI	LS IntOSC f_{RL}	Timer count clock
	Register access clock	CPUCLK_L	Bus clock

Table 29.9 Clock Supply (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
WDTA0	WDTATCKI	CKSCLK_AWDTA	Timer count clock
	Register access clock	CPUCLK_L	Bus clock
WDTA1	WDTATCKI	LS IntOSC f_{RL}	Timer count clock
	Register access clock	CPUCLK_L	Bus clock

29.1.4 Interrupt Requests

WDTAn interrupt requests are listed in the following table.

Table 29.10 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
WDTA0			
INTWDTAn	WDTA0 75% interrupt	40	—
WDTA1			
INTWDTAn	WDTA1 75% interrupt	41	—
WDTA2			
INTWDTAn	WDTA2 75% interrupt	42	—

Table 29.11 Interrupt Requests (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
WDTA0			
INTWDTAn	WDTA0 75% interrupt	40	—
WDTA1			
INTWDTAn	WDTA1 75% interrupt	41	—

Table 29.12 Interrupt Request (FE Level Non-Maskable Interrupt) (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
WDTA0			
WDTAnTNMI	WDTA0 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA0NMI	—
WDTA1			
WDTAnTNMI	WDTA1 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA1NMI	—
WDTA2			
WDTAnTNMI	WDTA2 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA2NMI	—

Table 29.13 Interrupt Request (FE Level Non-Maskable Interrupt) (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
WDTA0			
WDTAnTNMI	WDTA0 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA0NMI	—
WDTA1			
WDTAnTNMI	WDTA1 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA1NMI	—

29.1.5 Reset Sources

WDTAn reset sources are listed in the following table. WDTAn is initialized by these reset sources.

Table 29.14 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
WDTA0	Reset sources (AWORES)
WDTA1	All reset sources (ISORES)
WDTA2	All reset sources (ISORES)

Note: WDTA1 and WDTA2 are stopped in STOP mode.

Table 29.15 Reset Sources (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
WDTA0	Reset sources (AWORES)
WDTA1	All reset sources (ISORES)

Note: WDTA1 is stopped in STOP mode.

29.1.6 Internal Input/Output Signals

The internal input/output signals of WDTA are listed in the following table.

Table 29.16 Internal Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Connection
WDTAnTRES	Watchdog timer reset	Reset controller (WDTA0RES)
WDTAnTRES	Watchdog timer reset	Reset controller (WDTA1RES)
WDTAnTRES	Watchdog timer reset	Reset controller (WDTA2RES)

Table 29.17 Internal Input/Output Signals (H850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connection
WDTAnTRES	Watchdog timer reset	Reset controller (WDTA0RES)
WDTAnTRES	Watchdog timer reset	Reset controller (WDTA1RES)

29.1.7 State in RUN Mode and Stand-by Modes

The state of the WDTA in respective RUN mode and stand-by modes are listed in the following table.

Table 29.18 States in Respective RUN Mode and Stand-by Modes (RH850/F1KH-D8)

Run Mode/Stand-by Mode	State of WDTA0	State of WDTA1	State of WDTA2
RUN mode (HALT state)	Operation enabled	Operation enabled	Operation enabled
STOP mode	Operation enabled	Operation stopped	Operation stopped
DeepSTOP mode	Operation enabled	Power off	Power off
Cyclic RUN mode	Operation enabled	Operation prohibited	Operation prohibited
Cyclic STOP mode	Operation enabled	Operation stopped	Operation stopped

Table 29.19 States in Respective RUN Mode and Stand-by Modes (H850/F1KM-S4, RH850/F1KM-S1)

Run Mode/Stand-by Mode	State of WDTA0	State of WDTA1
RUN mode (HALT state)	Operation enabled	Operation enabled
STOP mode	Operation enabled	Operation stopped
DeepSTOP mode	Operation enabled	Power off
Cyclic RUN mode	Operation enabled	Operation prohibited
Cyclic STOP mode	Operation enabled	Operation stopped

29.2 Overview

29.2.1 Functional Overview

WDTA has the following functions:

- Selection of the operation mode after reset, by using the option bytes

  Enabling/disabling of WDTA, starting/stopping of the counter after reset, setting of the counter overflow time, and enabling/disabling of the VAC function can be selected. WDTA startup options to be set by the option bytes are described in **Table 29.20, WDTA Start-Up Options (RH850/F1KH-D8)** and **Table 29.21, WDTA Start-Up Options (RH850/F1KM-S4, RH850/F1KM-S1)**.

- WDTA trigger function

  Writing an activation code to the WDTA trigger register starts WDTA and restarts the counter. Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTA trigger register, which causes the counter to be restarted.

- 75% interrupt request signal

  An interrupt request signal can be generated when the WDTA counter reaches 75% of the overflow interval time (this function can be enabled or disabled by the setting of WDTAnMD.WDTAnWIE).

- Window function

  The period during which writing to the WDTA trigger register is valid (window-open period) can be set. Writing to the WDTA trigger register during a period other than the window-open period causes an error.

- WDTA error detection function

  When an error is detected, a non-maskable interrupt request or an internal reset is generated.

  For details about the error sources, see **Section 29.5.3, WDTA Error Detection**.

Table 29.20 WDTA Start-Up Options (RH850/F1KH-D8)

Start-Up Option	Function	Description	Option Byte
OPWDEN	WDTA setting	Enables/disables the WDTA. 0: WDTA is disabled 1: WDTA is enabled	● WDTA0: OPBT0.WDT0_0 ● WDTA1: OPBT0.WDT1_0 ● WDTA2: OPBT0.WDT2_0
OPWDOVF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	WDTA0/WDTA1/WDTA2: OPBT0.WDT_[2:0]
OPWDRUN	Start mode setting	Specifies the start mode. 0: Software trigger start mode 1: Default start mode For details, see **Section 29.5.1, WDTA after Reset Release**.	● WDTA0: OPBT0.WDT0_1 ● WDTA1: OPBT0.WDT1_1 ● WDTA2: OPBT0.WDT2_1
OPWDVAC	Variable activation code selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (ACH). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, see **Section 29.5.2, WDTA Trigger** and **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.	● WDTA0: OPBT0.WDT0_3 ● WDTA1: OPBT0.WDT1_3 ● WDTA2: OPBT0.WDT2_3

Table 29.21 WDTA Start-Up Options (RH850/F1KM-S4, RH850/F1KM-S1)

Start-Up Option	Function	Description	Option Byte
OPWDEN	WDTA setting	Enables/disables the WDTA. 0: WDTA is disabled 1: WDTA is enabled	• WDTA0: OPBT0.WDT0_0 • WDTA1: OPBT0.WDT1_0
OPWDOVF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	WDTA0/WDTA1: OPBT0.WDT_[2:0]
OPWDRUN	Start mode setting	Specifies the start mode. 0: Software trigger start mode 1: Default start mode For details, see **Section 29.5.1, WDTA after Reset Release**.	• WDTA0: OPBT0.WDT0_1 • WDTA1: OPBT0.WDT1_1
OPWDVAC	Variable activation code selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (AC_H). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, see **Section 29.5.2, WDTA Trigger** and **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.	• WDTA0: OPBT0.WDT0_3 • WDTA1: OPBT0.WDT1_3

NOTE

For the option byte settings, see **Section 44.9, Option Bytes**.

29.2.2 Block Diagram

Figure 29.1, Block Diagram of the Window Watchdog Timer A shows the main components of the WDTA.

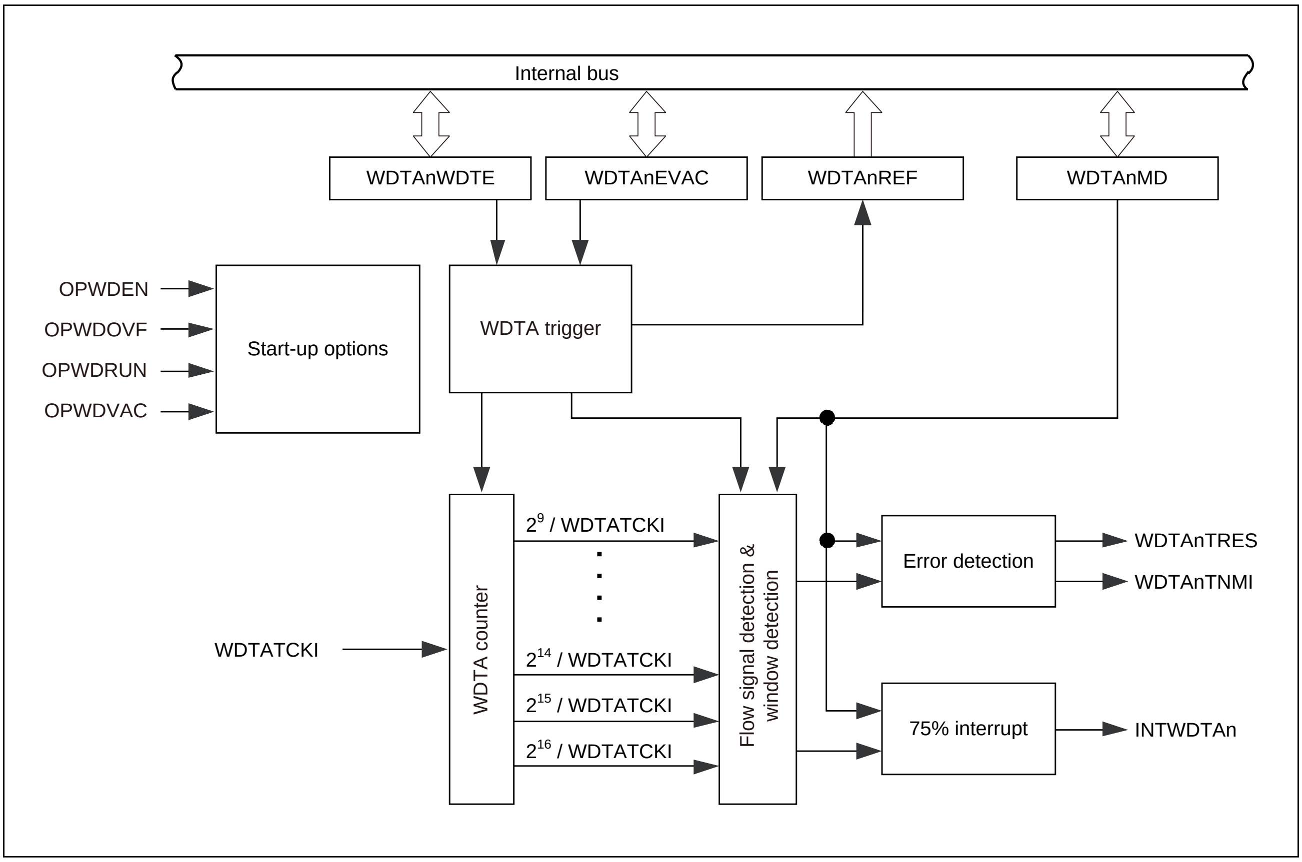


Figure 29.1 Block Diagram of the Window Watchdog Timer A

29.3 Registers

29.3.1 List of Registers

WDTA registers are listed in the following table.

For details about <WDTAn_base>, see **Section 29.1.2, Register Base Addresses**.

Table 29.22 List of Registers

Module Name	Register Name	Symbol	Address
WDTAn	WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000_H
	WDTA enable VAC register	WDTAnEVAC	<WDTAn_base> + 0004_H
	WDTA reference value register	WDTAnREF	<WDTAn_base> + 0008_H
	WDTA mode register	WDTAnMD	<WDTAn_base> + $000C_H$

29.3.2 WDTAnWDTE — WDTA Enable Register

This register is the WDTA trigger register when the VAC function is not used (start-up option OPWDVAC = 0).

Writing AC_H to this register generates a WDTA trigger and starts or restarts the WDTA counter.
See **Section 29.5.2, WDTA Trigger**, for details.

The behavior of this register depends on the setting of the start-up option (OPWDVAC), see **Table 29.25, WDTAnWDTE Behavior**.

WDTA0 is initialized by AWORES.
WDTA1 and WDTA2 are initialized by a reset of any type.

Access:	This register can be read or written in 8-bit units.
Address:	<WDTAn_base> + 0000_H
Value after reset:	The initial value depends on the start-up options (OPWDEN, OPWDRUN and OPWDVAC). See **Table 29.24, Values of WDTAnRUN7 after Reset**.

Bit	7	6	5	4	3	2	1	0
	WDTAnRUN[7:0]							
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.23 WDTAnWDTE Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnRUN[7:0]	Writing the fixed activation code (AC_H) generates the WDTA trigger and starts/restarts the WDTAn counter. Writing a value other than AC_H generates an error. The WDTAn cannot be stopped once it is started. See **Table 29.25, WDTAnWDTE Behavior**, when reading from or writing to these bits.

The WDTAnRUN7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is disabled (OPWDVAC = 0). **Table 29.24, Values of WDTAnRUN7 after Reset** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

Table 29.24 Values of WDTAnRUN7 after Reset

Start-Up Options				
OPWDEN	OPWDVAC	OPWDRUN	Start Mode	Value of WDTAnRUN7 after Reset
1	0	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnWDTE during read/write accesses depends on the OPWDVAC setting, as shown in **Table 29.25, WDTAnWDTE Behavior**.

Table 29.25 WDTAnWDTE Behavior

		WDTAnWDTE	
OPWDVAC	Description	Read	Write
0	The VAC function is disabled. WDTAnWDTE is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). AC_H is returned (after the activation of WDTAn).	WDTA trigger Write AC_H*1.
1	The VAC function is enabled. WDTAnWDTE is disabled.	$2C_H$ is returned.	Writing is ignored.

Note 1. Any other write value will cause an error.

29.3.3 WDTAnEVAC — WDTA Enable VAC Register

This register is the WDTA trigger register when the VAC function is used (start-up option OPWDVAC = 1).

Writing a correct activation code to this register generates a WDTA trigger and starts or restarts the WDTA counter. For details, see **Section 29.5.2, WDTA Trigger**. For details about the activation codes when the VAC function is used, see **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

The behavior of this register depends on the setting of the start-up option (OPWDVAC). See **Table 29.28, WDTAnEVAC Behavior**.

WDTA0 is initialized by AWORES.
WDTA1 and WDTA2 are initialized by a reset of any type.

Access: This register can be read or written in 8-bit units.

Address: <WDTAn_base> + 0004_H

Value after reset: The initial value depends on the start-up options (OPWDEN, OPWDRUN and OPWDVAC). See **Table 29.27, Values of WDTAnEVAC7 after Reset**.

Bit	7	6	5	4	3	2	1	0
	WDTAnEVAC[7:0]							
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.26 WDTAnEVAC Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnEVAC [7:0]	Writing a variable activation code generates the WDTA trigger and starts/restarts the WDTAn counter. Writing an incorrect activation code generates an error. The WDTAn cannot be stopped once it is started. See **Table 29.28, WDTAnEVAC Behavior**, when reading from or writing to these bits.

The WDTAnEVAC7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is enabled (OPWDVAC = 1). **Table 29.27, Values of WDTAnEVAC7 after Reset** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

Table 29.27 Values of WDTAnEVAC7 after Reset

Start-Up Options				
OPWDEN	OPWDVAC	OPWDRUN	Start Mode	Value of WDTAnEVAC7 after Reset
1	1	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnEVAC during read/write accesses depends on the OPWDVAC setting, as shown in **Table 29.28, WDTAnEVAC Behavior**.

Table 29.28 WDTAnEVAC Behavior

OPWDVAC	Description	WDTAnEVAC	
		Read	Write
0	The VAC function is disabled. WDTAnEVAC is disabled.	$2C_H$ is returned.	Writing is ignored.
1	The VAC function is enabled. WDTAnEVAC is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). The variable activation code written last is read (after the activation of WDTAn).	Write the variable activation code*[1] For details, see **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

Note 1. Any other write value will cause an error.

29.3.4 WDTAnREF — WDTA Reference Value Register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. See **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

If the VAC function is disabled (OPWDVAC = 0), reading this register returns 00_H.

WDTA0 is initialized by AWORES.
WDTA1 and WDTA2 are initialized by a reset of any type.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	<WDTAn_base> + 0008_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	WDTAnREF[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 29.29 WDTAnREF Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation for the VAC function.

29.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, 75% interrupt enable/disable, the error mode, and the window-open period.

The value of this register can be updated only once after reset release and before the first trigger is generated. The updated value will be effective after a maximum of 3 x Timer count clock cycles once the WDTA trigger register is written.

Updating this register after the first WDTA trigger is generated generates an error, but an error does not occur if the same value has been written to it.

WDTA0 is initialized by AWORES.

WDTA1 and WDTA2 are initialized by a reset of any type.

Access:	This register can be read or written in 8-bit units.
Address:	<WDTAn_base> + $000C_H$
Value after reset:	The initial value depends on the start-up options (OPWDOVF[2:0]). See **Table 29.21, WDTA Start-Up Options**.

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>—</td><td colspan="3">WDTAnOVF[2:0]</td><td>WDTAnWIE</td><td>WDTAnERM</td><td colspan="2">WDTAnWS[1:0]</td></tr>
<tr><td>Value after reset</td><td>0</td><td>*1</td><td>*1</td><td>*1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>R/W</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>

Note 1. The WDTAnOVF[2:0] value after reset can be set by the start-up options OPWDOVF[2:0].

Table 29.30 WDTAnMD Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>6 to 4</td><td>WDTAnOVF[2:0]</td><td>Selects the overflow interval time.
<table>
<tr><th>WDTAn OVF2</th><th>WDTAn OVF1</th><th>WDTAn OVF0</th><th>Overflow Interval Time</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^9 / WDTATCKI</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{10} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{11} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{12} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{13} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{14} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{15} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{16} / WDTATCKI</td></tr>
</table></td></tr>
<tr><td>3</td><td>WDTAnWIE</td><td>Enables/disables the 75% interrupt request INTWDTAn.
0: INTWDTAn is disabled.
1: INTWDTAn is enabled.</td></tr>
<tr><td>2</td><td>WDTAnERM</td><td>Specifies the error mode.
0: NMI request mode
1: Reset mode</td></tr>
<tr><td>1, 0</td><td>WDTAnWS[1:0]</td><td>Selects the window-open period.
<table>
<tr><th>WDTAn WS1</th><th>WDTAn WS0</th><th>Window-Open Period</th></tr>
<tr><td>0</td><td>0</td><td>25%</td></tr>
<tr><td>0</td><td>1</td><td>50%</td></tr>
<tr><td>1</td><td>0</td><td>75%</td></tr>
<tr><td>1</td><td>1</td><td>100%</td></tr>
</table></td></tr>
</table>

29.4 Interrupt Sources

WDTA checks the status of the WDTA counter value, detects illegal accesses to the WDTA-related registers, and generates an interrupt request. The following are WDTA interrupt requests:

(1) INTWDTAn (WDTA timer count 75% interrupt request)
An interrupt request signal is generated at 75% of the counter overflow time of the WDTA timer.
An interrupt request signal can be enabled or disabled by using the WDTA mode register (WDTAnMD).

(2) WDTAnTNMI (WDTA error detection interrupt)

Detection of a WDTA error to generation of an NMI interrupt request. The WDTA mode register (WDTAnMD) can be used to switch between an NMI interrupt and a reset. For details about WDTA errors, see **Section 29.5.3, WDTA Error Detection**.

29.5 Functions

29.5.1 WDTA after Reset Release

29.5.1.1 Start Modes

There are two start modes (software start mode and default start mode) when WDTAn starts after reset release. The start mode can be selected by the start-up option.

The start mode selection is listed in **Table 29.31, Start Modes**.

Table 29.31 Start Modes

Start-Up Option OPWDRUN	Start Mode	Description
0	Software trigger	• The WDTA counter stops (0000_H) after reset release. • Writing an activation code to the WDTA trigger register starts WDTA.
1	Default	The WDTA counter starts after reset release.

29.5.1.2 WDTA Settings after Reset Release

(1) **Table 29.32, WDTA Settings after Reset Release** shows the WDTA settings after reset release.

Table 29.32 WDTA Settings after Reset Release

Function	Setting	Remark
WDTA enable/disable	Specified by start-up options	
Start mode		
VAC function		
WDTA overflow interval time	Specified by start-up options	Modification is possible only once by the setting of the WDTA mode register (WDTAnMD).
75% interrupt mode	75% interrupt disabled	
Behavior on error detection	Reset generation	
Window-open period	100%	

The setting of the WDTA mode register (WDTAnMD) is enabled behind a maximum of 3 x Timer count clock cycles when the first WDTA trigger is generated (writing an activation code to WDTAnWDTE and WDTAnEVAC). Perform the WDTAnMD register setting before a WDTA trigger is generated.

Setting of WDTA by using WDTAnMD is possible only once. If the value set to WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value is set.

29.5.1.3 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 29.2, Timing Diagram of WDTA Start in Default Start Mode**.

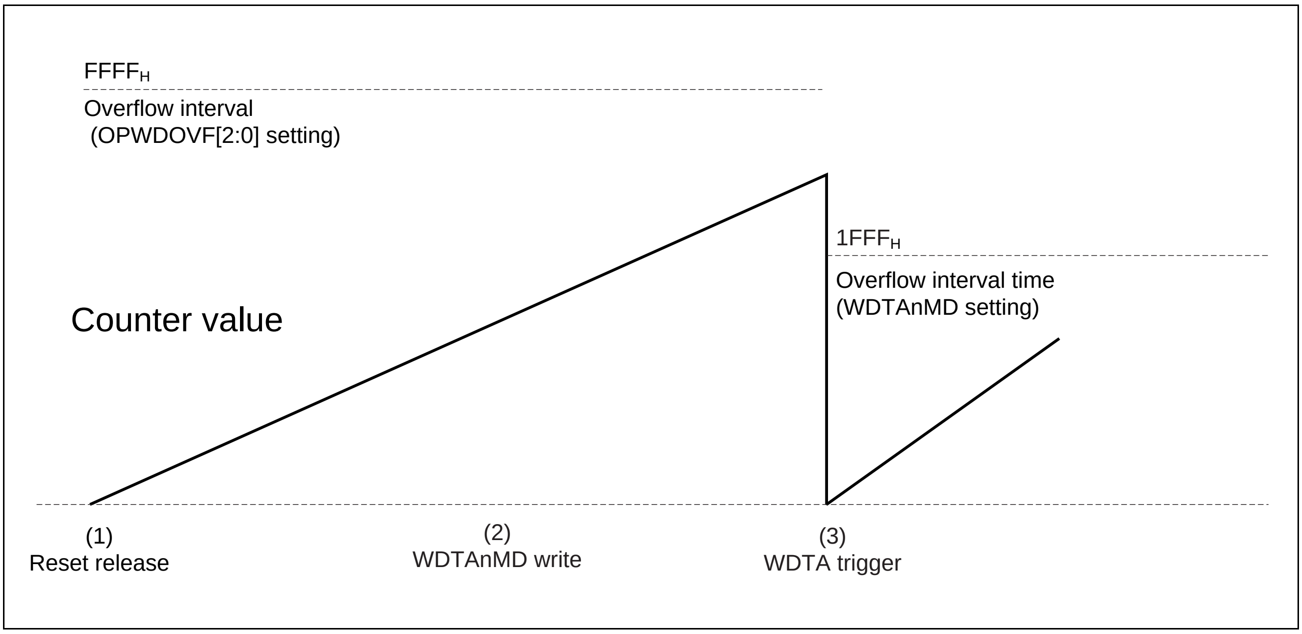


Figure 29.2 Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 29.2, Timing Diagram of WDTA Start in Default Start Mode** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

Example: Overflow interval time after reset release = 2^{16}/WDTATCKI (OPWDOVF[2:0] = 111_B)

(2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.

(3) Write to the WDTA trigger register before the WDTA counter overflows.
The WDTAnMD setting is applied by the generation of a WDTA trigger behind a maximum of 3 x Timer count clock.

Example: Overflow interval time after a WDTA trigger is generated = 2^{13}/WDTATCKI

29.5.1.4 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 29.3, Timing Diagram of WDTA Start in Software Trigger Start Mode**.

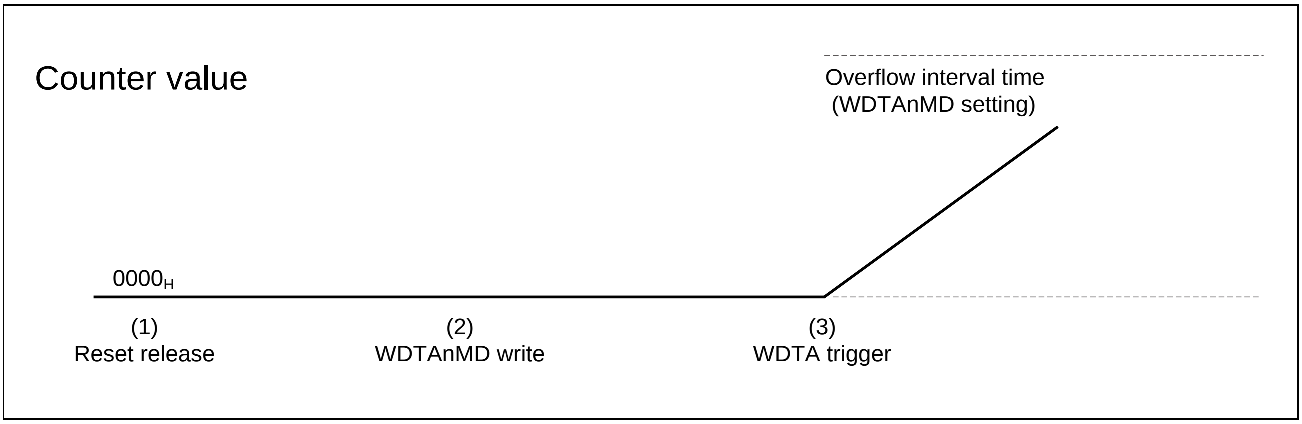


Figure 29.3 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram shown in **Figure 29.3, Timing Diagram of WDTA Start in Software Trigger Start Mode** shows the following behaviors:

(1) The WDTA counter value remains 0000_H until the first trigger is generated after reset release. The overflow interval time is set by start-up options, but this setting has no effect because the counter is not operating.

(2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.

(3) The WDTA counter starts by the generation of a WDTA trigger.
After the generation of a WDTA trigger behind a maximum of 3 x Timer count clock, the overflow interval time specified in WDTAnMD and other settings are applied.

29.5.2 WDTA Trigger

Writing a special value called an activation code to the WDTA enable register (WDTAnWDTE) and the WDTA enable VAC register (WDTAnEVAC) leads to generation of a WDTA trigger.

The WDTA trigger has the following functions:

- Starting the WDTA counter in software trigger start mode
- Restarting the WDTA counter
- Setting the WDTA mode specified by the WDTAnMD register (only for the first WDTA trigger after reset release)

The WDTA trigger register, which generates a WDTA trigger, is specified by the start-up option OPWDVAC.

Table 29.33, WDTA Trigger Registers and Activation Codes lists the WDTA trigger registers and activation codes.

Table 29.33 WDTA Trigger Registers and Activation Codes

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC = 0)	WDTAnWDTE	AC_H
Variable (OPWDVAC = 1)	WDTAnEVAC	For details, see **Section 29.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

CAUTION

- When successively writing the processing to clear WDTA, the clear processing below is not acknowledged for the following period:
  "12 × CPU clock*[1] cycles + 6 × WDT clock*[2] cycles"
- After writing the processing to clear WDTA and then changing to standby mode during the above-mentioned period, the clear processing below is not acknowledged for the following period after return from stand-by mode:
  "6 × CPU clock*[1] cycles + 3 × WDT clock*[2] cycles"

Note 1. CPU clock: Clock selected by CKSC_CPUCLKS_CTL and CKSC_CPUCLKD_CTL

Note 2. WDT clock: Clock selected by CKSC_AWDTAD_CTL is used for WDTA0.
LS IntOSC is used for WDTA1 and WDTA2.

29.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set in the WDTA trigger register (WDTAnEVAC) when the VAC function is used, by using the WDTA reference value register (WDTAnREF):

ExpectWDTE = AC_H – WDTAnREF (previous)

Note that the value in the WDTAnREF register is updated every time a start-code is written to the trigger register WDTAnEVAC. Use the following expression to calculate the updated value of the WDTAnREF register:

WDTAnREF (following) = (rotate the value of ExpectWDTE to the left by 1 bit)

Table 29.34, Expected Variable Activation Code Development lists the variable activation codes according to the number of WDTA triggers.

Table 29.34 Expected Variable Activation Code Development

No*1	WDTAnREF (Previous)		ExpectWDTE (AC_H - WDTAnREF)		WDTAnREF (Following)	
0	0000 0000	00_H	1010 1100	AC_H	0101 1001	59_H
1	0101 1001	59_H	0101 0011	53_H	1010 0110	$A6_H$
2	1010 0110	$A6_H$	0000 0110	06_H	0000 1100	$0C_H$
…	…	…	…	…	…	…

Note 1. Number of triggers after reset

NOTE

Writing an incorrect activation code generates an error.

29.5.3 WDTA Error Detection

WDTA detects an error, including generation of the WDTA count overflow or illegal operations.

The following events are detected as errors:

- WDTA counter overflow
- Incorrect activation code is written to the WDTA trigger register
- Writing to the trigger register at a time outside the window-open period
- When the setting value in the WDTA mode register (WDTAnMD) is changed after the first WDTA trigger is generated
- When the value of the WDTA mode register (WDTAnMD) is changed twice before the first WDTA trigger is generated*[1]

Note 1. In this case, WDTAnRES is generated regardless setting of WDTA error mode bit (WDTAnMD.WDTAnERM).

29.5.3.1 WDTA Error Mode

When a WDTA error is detected, either an NMI interrupt or a reset is generated according to the setting of the WDTA error mode bit (WDTAnMD.WDTAnERM). The error mode bit after reset release is set to the reset mode.

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: Reset mode

Figure 29.4, Timing Diagram of WDTA NMI Request or Reset Generation shows the reset or NMI request generation when the counter overflows and default start mode is selected.

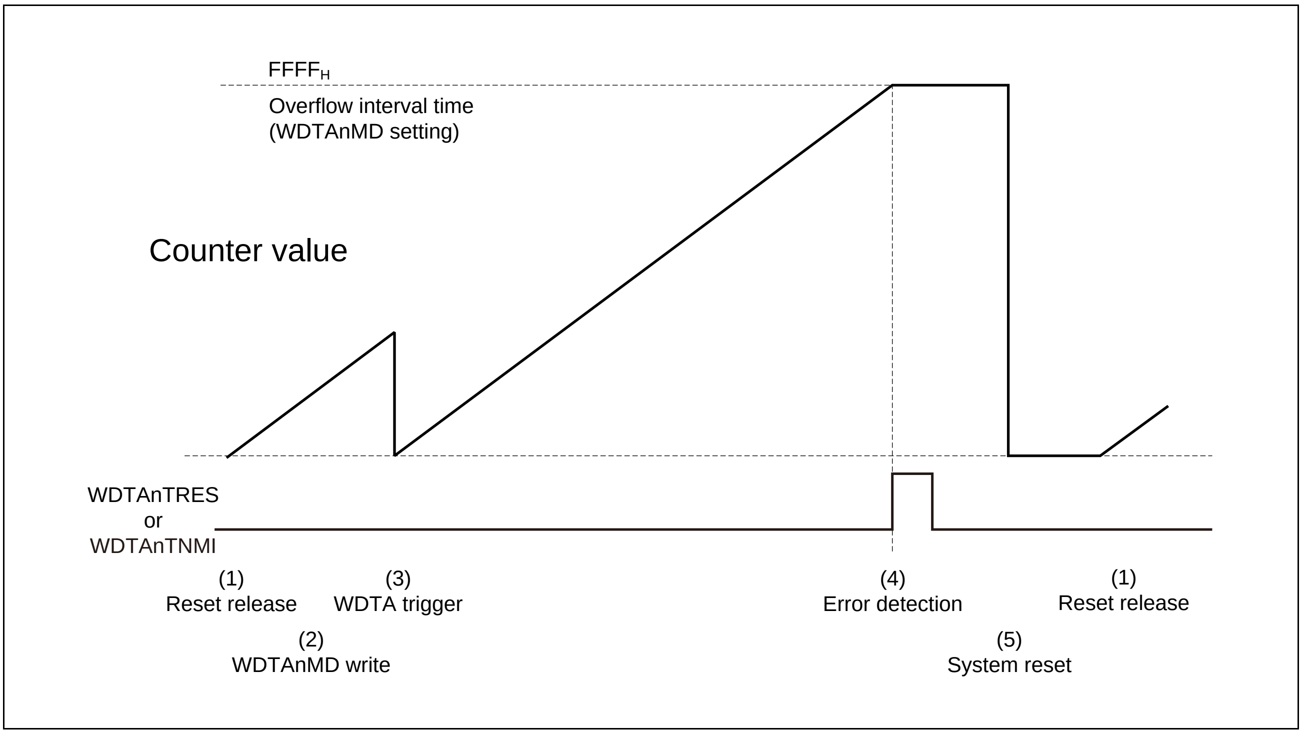


Figure 29.4 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram shown in **Figure 29.4, Timing Diagram of WDTA NMI Request or Reset Generation** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated.
In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.
The counter value remains until a system reset is performed.

(5) When the system is reset, the counter is cleared and stops until reset release.

29.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

This function can be enabled or disabled by the WDTAnMD.WDTAnWIE register.

Figure 29.5, Timing Diagram of WDTA 75% Interrupt Request Signals shows the 75% interrupt request generated under following conditions:

- Default start mode is selected.
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time is 2^{16}/WDTATCKI

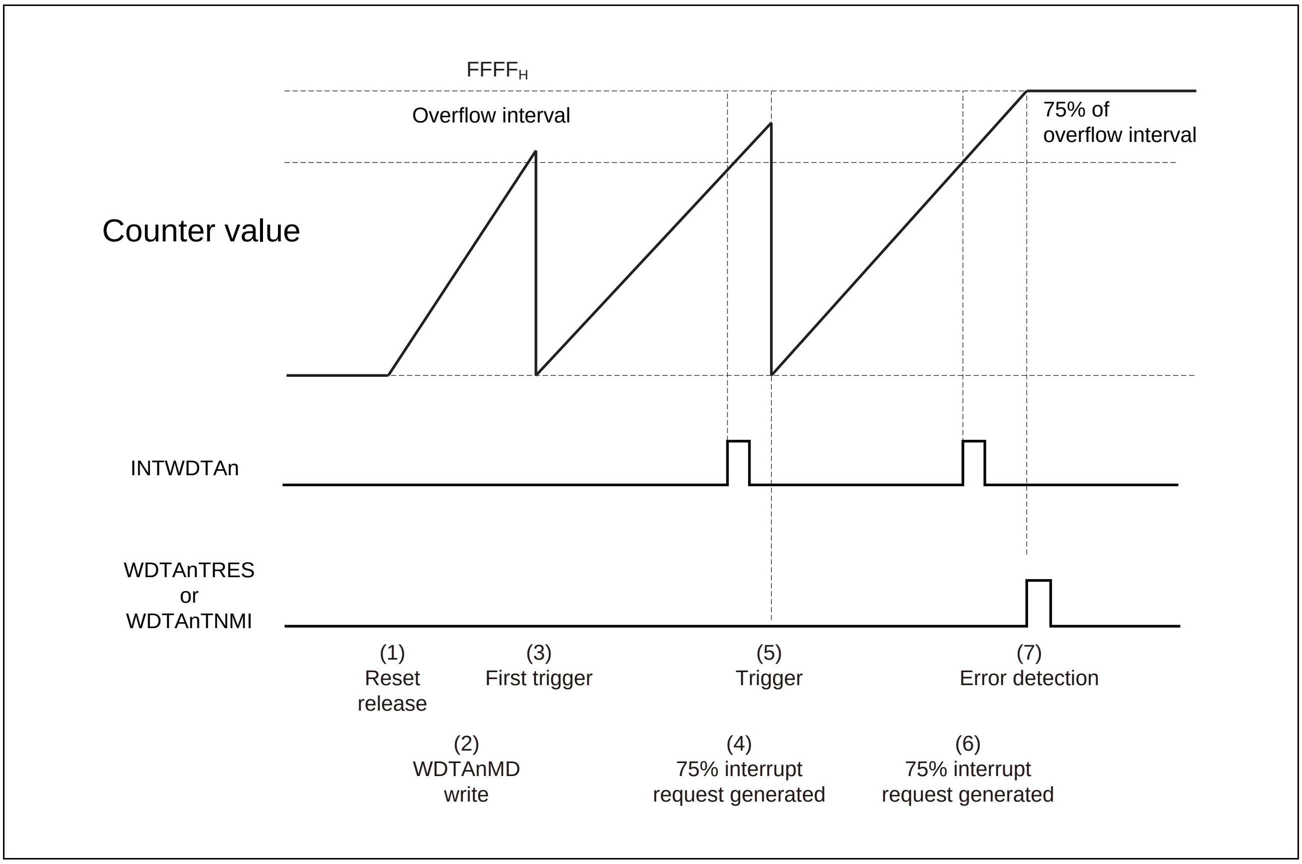


Figure 29.5 Timing Diagram of WDTA 75% Interrupt Request Signals

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(5) The WDTA trigger restarts counting.

(6) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(7) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.

29.5.5 Window Function

The period when a WDTA trigger is valid (window-open period) can be set. If the window-open period is set to the value less than 100%, the generation of a WDTA trigger not in the window-open period causes an error. The window-open period after reset release is 100%. The period is set to the value by the WDTAnMD.WDTAnWS[1:0] setting after the first WDTA trigger is generated.

Figure 29.6, Timing Diagram of WDTA Window Function shows the behavior of the window function under the following conditions.

- Default start mode is selected.
- 25% window-open period is enabled after the first WDTA trigger is generated (WDTAnWS[1:0] = 00_B)
- WDTA overflow interval time is 2^{16}/WDTATCKI

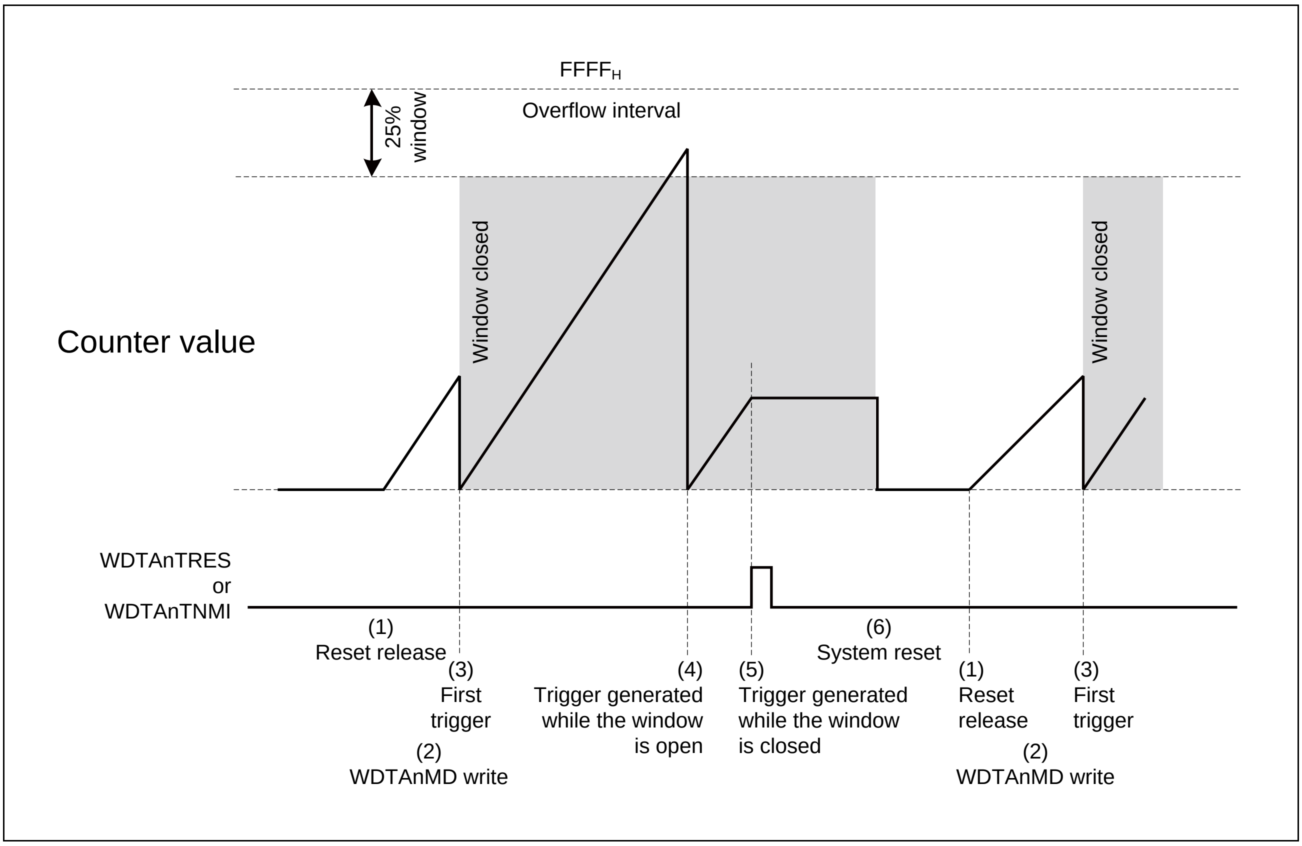


Figure 29.6 Timing Diagram of WDTA Window Function

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) During the window-open period, the WDTA trigger restarts counting.

(5) During the window-closed period, an error is detected by the WDTA trigger. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.

(6) When the system is reset, the counter is cleared and stops until reset release.

Section 30 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of OSTM.

30.1 Features of RH850/F1KH, RH850/F1KM OSTM

30.1.1 Number of Units

This microcontroller has the following number of units of the OSTM.

Each OSTM unit has single channel interface.

Table 30.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	10	10	10
Name	OSTMn (n = 0 to 9)	OSTMn (n = 0 to 9)	OSTMn (n = 0 to 9)

Table 30.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	5	5	5	5	5
Name	OSTMn (n = 0 to 4)	OSTMn (n = 0 to 4)	OSTMn (n = 0 to 4)	OSTMn (n = 0 to 4)	OSTMn (n = 0 to 4)

Table 30.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	OSTMn (n = 0)	OSTMn (n = 0)	OSTMn (n = 0)	OSTMn (n = 0)

Table 30.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual OSTM units are identified by the index “n”; for example, OSTMnCNT is the OSTM counter register.

30.1.2 Register Base Addresses

OSTM base addresses are listed in the following table.

OSTM register addresses are given as offsets from the base addresses.

Table 30.5 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<OSTM0_base>	FFD7 0000$_H$
<OSTM1_base>	FFD7 0100$_H$
<OSTM2_base>	FFD7 0200$_H$
<OSTM3_base>	FFD7 0300$_H$
<OSTM4_base>	FFD7 0400$_H$
<OSTM5_base>	FFD7 1000$_H$
<OSTM6_base>	FFD7 1100$_H$
<OSTM7_base>	FFD7 1200$_H$
<OSTM8_base>	FFD7 1300$_H$
<OSTM9_base>	FFD7 1400$_H$

Table 30.6 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<OSTM0_base>	FFD7 0000$_H$
<OSTM1_base>	FFD7 0100$_H$
<OSTM2_base>	FFD7 0200$_H$
<OSTM3_base>	FFD7 0300$_H$
<OSTM4_base>	FFD7 0400$_H$

Table 30.7 Register Base Address (RH850/F1KM-S1)

Base Address Name	Base Address
<OSTM0_base>	FFD7 0000$_H$

30.1.3 Clock Supply

The OSTM clock supply is shown in the following table.

Table 30.8 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
OSTMn	PCLK	CPUCLK_L	Timer count clock
	Register access clock	CPUCLK_L	Bus clock

30.1.4 Interrupt Requests

OSTM interrupt requests are listed in the following table.

Table 30.9 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	84 (INTOSTM0)	—
OSTM5			
OSTMTINT	OSTM5 interrupt	314 (INTOSTM5)	—

Table 30.10 Interrupt Request (RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	84 (INTOSTM0)	—

Table 30.11 Interrupt Requests (FE Level Maskable Interrupt Request) (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	INTOSTM0_FE	—
OSTM1			
OSTMTINT	OSTM1 interrupt	INTOSTM1_FE	—
OSTM2			
OSTMTINT	OSTM2 interrupt	INTOSTM2_FE	—
OSTM3			
OSTMTINT	OSTM3 interrupt	INTOSTM3_FE	—
OSTM4			
OSTMTINT	OSTM4 interrupt	INTOSTM4_FE	—
OSTM5			
OSTMTINT	OSTM5 interrupt	INTOSTM5_FE	—
OSTM6			
OSTMTINT	OSTM6 interrupt	INTOSTM6_FE	—
OSTM7			
OSTMTINT	OSTM7 interrupt	INTOSTM7_FE	—
OSTM8			
OSTMTINT	OSTM8 interrupt	INTOSTM8_FE	—
OSTM9			
OSTMTINT	OSTM9 interrupt	INTOSTM9_FE	—

Table 30.12 Interrupt Requests (FE Level Maskable Interrupt Request) (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	INTOSTM0_FE	—
OSTM1			
OSTMTINT	OSTM1 interrupt	INTOSTM1_FE	—
OSTM2			
OSTMTINT	OSTM2 interrupt	INTOSTM2_FE	—
OSTM3			
OSTMTINT	OSTM3 interrupt	INTOSTM3_FE	—
OSTM4			
OSTMTINT	OSTM4 interrupt	INTOSTM4_FE	—

Table 30.13 Interrupt Request (FE Level Maskable Interrupt Request) (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	INTOSTM0_FE	—

30.1.5 Reset Sources

OSTM reset sources are listed in the following table. OSTM is initialized by these reset sources.

Table 30.14 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
OSTMn	All reset sources (ISORES)

30.2 Overview

OSTM is a 32-bit timer/counter.

It can be used in interval timer mode or in free-run compare mode. The settings for operating mode specify the direction of counting (up or down) to control the generation of interrupt requests.

30.2.1 Functional Overview

OSTM has the following features.

- Two operating modes
  - Interval timer mode
  - Free-run compare mode
- OSTMTINT interrupt

30.2.2 Block Diagram

The following block diagram shows the main components of OSTM.

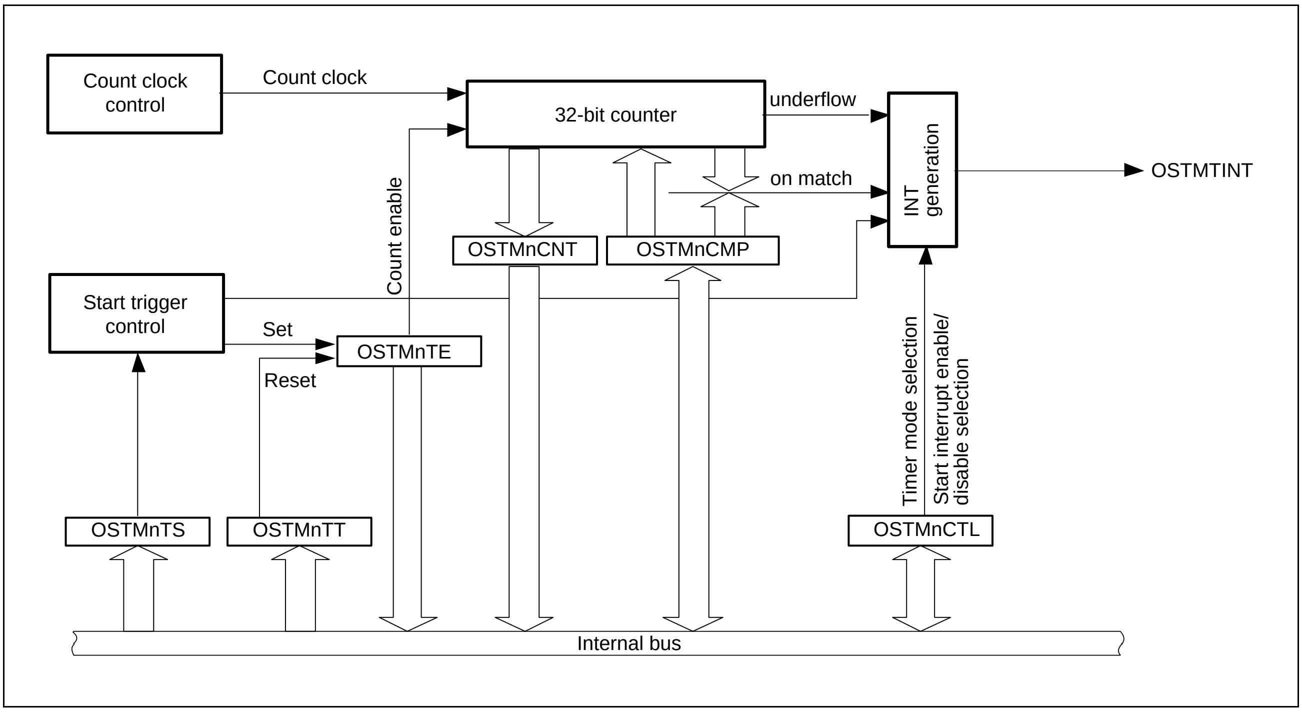


Figure 30.1 Block Diagram of OSTM

30.2.3 Count Clock

The count clock used by OSTM is PCLK.

30.2.4 Interrupt Sources (OSTMTINT)

An OSTMTINT interrupt request is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-run compare mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

This is illustrated in the following figure.

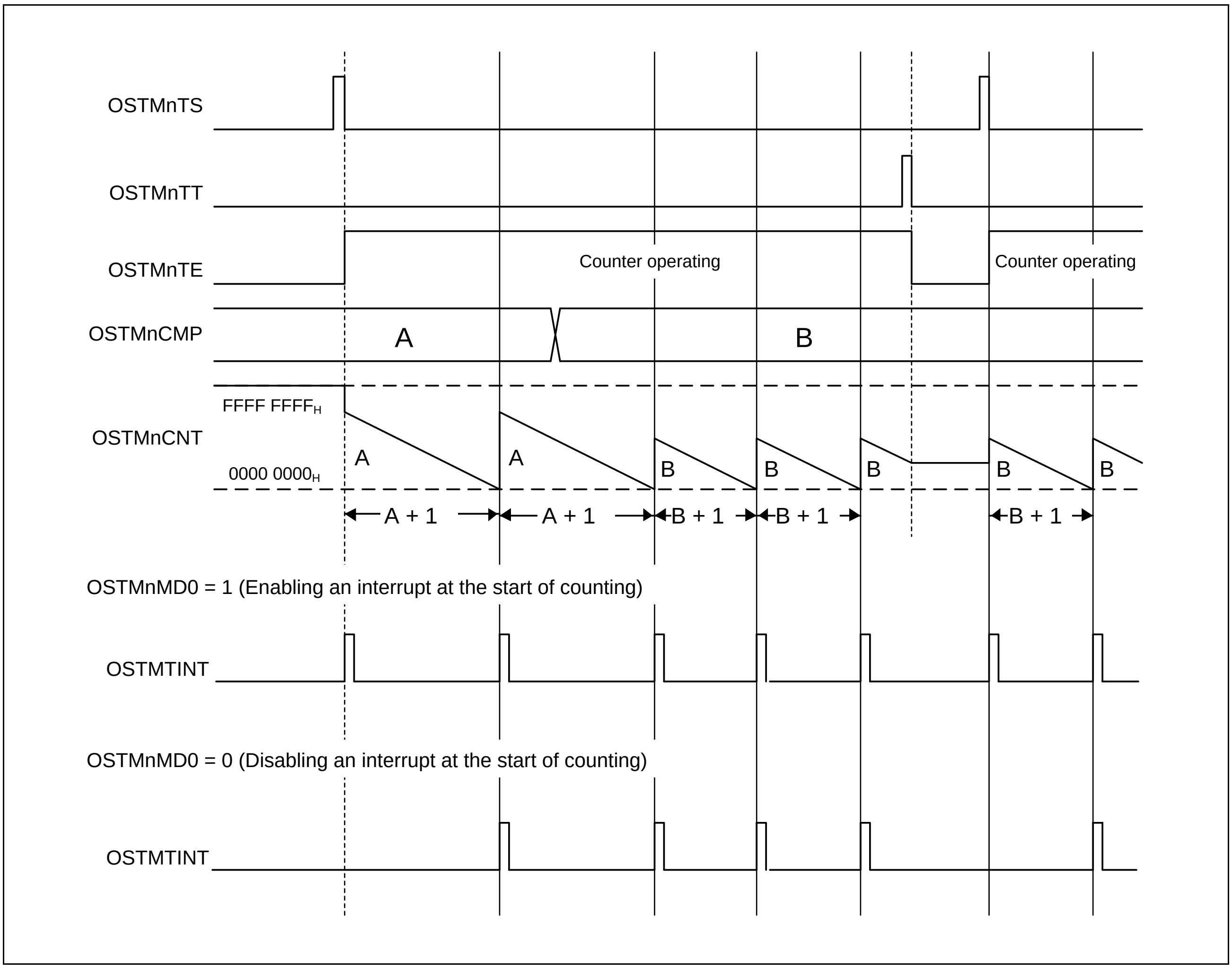


Figure 30.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

30.3 Registers

30.3.1 List of Registers

OSTM registers are listed in the following table.

For details about <OSTMn_base>, see **Section 30.1.2, Register Base Addresses**.

Table 30.15 List of Registers

Module Name	Register Name	Symbol	Address
OSTMn	OSTMn compare register	OSTMnCMP	<OSTMn_base> + 00_H
	OSTMn counter register	OSTMnCNT	<OSTMn_base> + 04_H
	OSTMn count enable status register	OSTMnTE	<OSTMn_base> + 10_H
	OSTMn count start trigger register	OSTMnTS	<OSTMn_base> + 14_H
	OSTMn count stop trigger register	OSTMnTT	<OSTMn_base> + 18_H
	OSTMn control register	OSTMnCTL	<OSTMn_base> + 20_H
	OSTMn emulation register	OSTMnEMU	<OSTMn_base> + 24_H

30.3.2 OSTMnCMP — OSTMn Compare Register

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the operation mode.

Access: This register can be read or written in 32-bit units.

Address: <OSTMn_base> + 00_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.16 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP[31:0]	• In interval timer mode: start value of the down-counter • In free-run compare mode: compare value

30.3.3 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <OSTMn_base> + 04_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.17 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT[31:0]	Timer counter value

Table 30.18, Correspondence among Operating Mode, Counting Direction, and Start Value lists the correspondence among the OSTM operating mode, counting direction, and start value. The start value indicates the value to be read after the operating mode is changed.

Table 30.18 Correspondence among Operating Mode, Counting Direction, and Start Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Start Value
Interval timer mode	0*1	Down	FFFF $FFFF_H$
Free-run compare mode	1	Up	0000 0000_H

Note 1. Value after reset.

30.3.4 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <OSTMn_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.19 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OSTMnTE	Indicates whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1. Setting OSTMnTT.OSTMnTT to 1 resets this bit to 0.

NOTE

If the counter is disabled, the counter value retains its value.

If the counter is restarted, it

- restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- restarts counting up from the counter value $0000\ 0000_H$ if it is in free-run compare mode.

30.3.5 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access:	This register is a write-only register that can be written in 8-bit units.
Address:	<OSTMn_base> + 14_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 30.20 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTS	Starts the counter: 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. ● In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. ● In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

30.3.6 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <OSTMn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 30.21 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTT	Stops the counter: 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

30.3.7 OSTMnCTL — OSTMn Control Register

This register specifies the operating mode for the counter and controls the generation of OSTMTINT interrupt requests when counting starts.

Although this register is readable and writable, this register can only be written when OSTMnTS.OSTMnTE = 0.

Access: This register can be read or written in 8-bit units.

Address: <OSTMn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OSTMnMD1	OSTMnMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.22 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OSTMnMD1	Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls the generation of OSTMTINT interrupt requests at the start of counting: 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled.

30.3.8 OSTMnEMU — OSTMn Emulation Register

This register controls operation in combination with SVSTOP.

Access:	This register can be read or written in 8-bit units. Only proceed with writing while the counter is stopped (OSTMnTE.OSTMnTE = 0 and EPC.SVSTOP = 0).
Address:	<OSTMn_base> + 24_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	OSTMnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 30.23 OSTMnEMU Register Contents

Bit Position	Bit Name	Function
7	OSTMnSVSDIS	When EPC.SVSTOP = 0 Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on) regardless of the value of this bit (0/1). When EPC.SVSTOP = 1 0: The count clock is stopped when the debugger acquires control of the microcontroller (at breakpoints and so on). 1: Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

30.4 Operation

30.4.1 Starting and Stopping OSTM

OSTM is started and stopped as follows:

Starting the Timer

OSTM is started by the following setting.

- Setting the OSTMnTS.OSTMnTS bit to 1

The OSTMnTE.OSTMnTE status bit is set to 1.

The counter starts to count up or down in accordance with the settings for operating mode. For details, see **Section 30.4.2, Interval Timer Mode** and **Section 30.4.3**, **Free-Run Compare Mode**.

Stopping the Timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops OSTM.

This also clears the OSTMnTE.OSTMnTE status bit.

30.4.2 Interval Timer Mode

In interval timer mode, OSTM can be used as a reference timer generating interrupt requests at fixed intervals.

30.4.2.1 Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMTINT interrupt request is generated when the counter underflows (reaches $0000\ 0000_H$).

To select interval timer mode, set OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next $0000\ 0000_H$ is reached. Then the counter continues with the new value.

OSTMTINT Period

The periods of OSTMTINT is:

- OSTMTINT generation period = count clock period × (OSTMnCMP + 1)

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode.

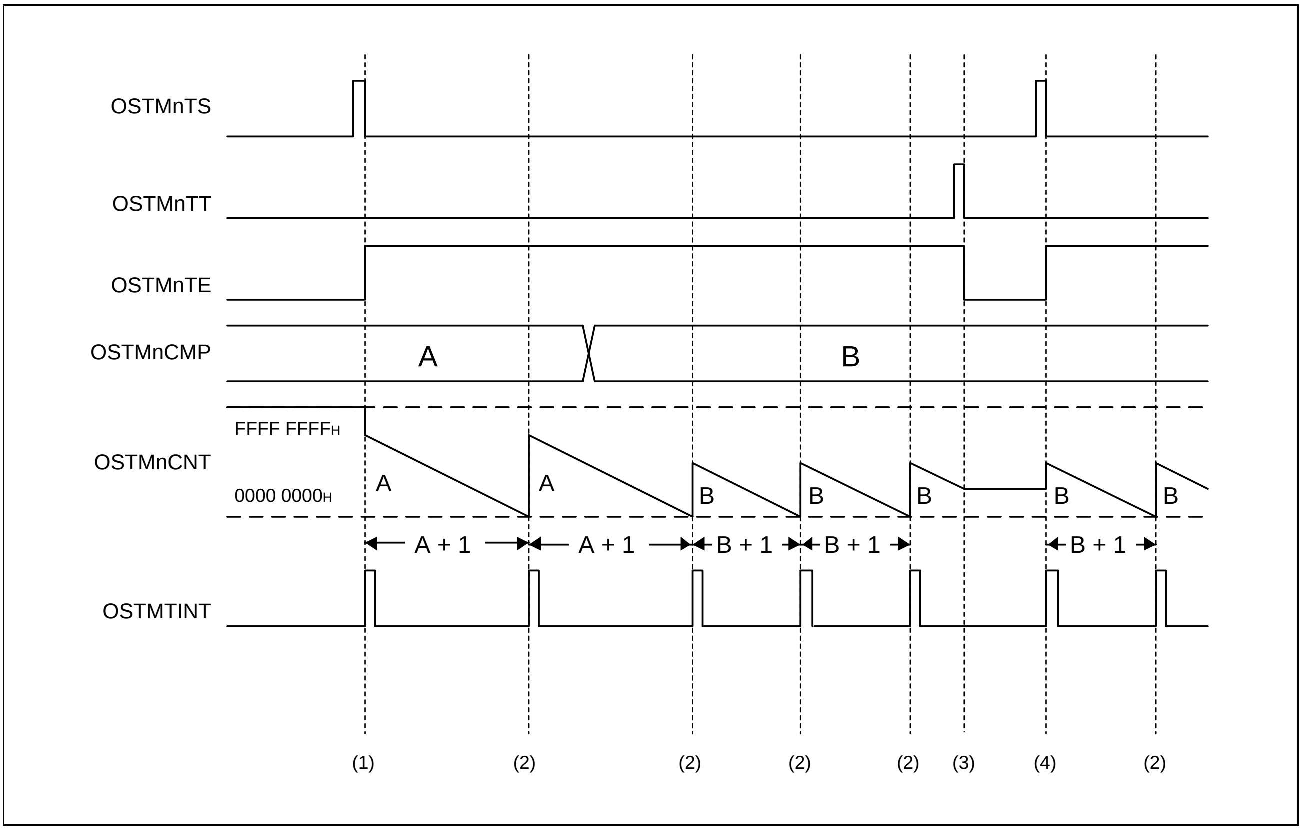


Figure 30.3 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 is 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register indicates the counter value.

(2) When the counter reaches 0000 0000_H, an OSTMTINT interrupt request is generated. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

Forced Restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the timing in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1).

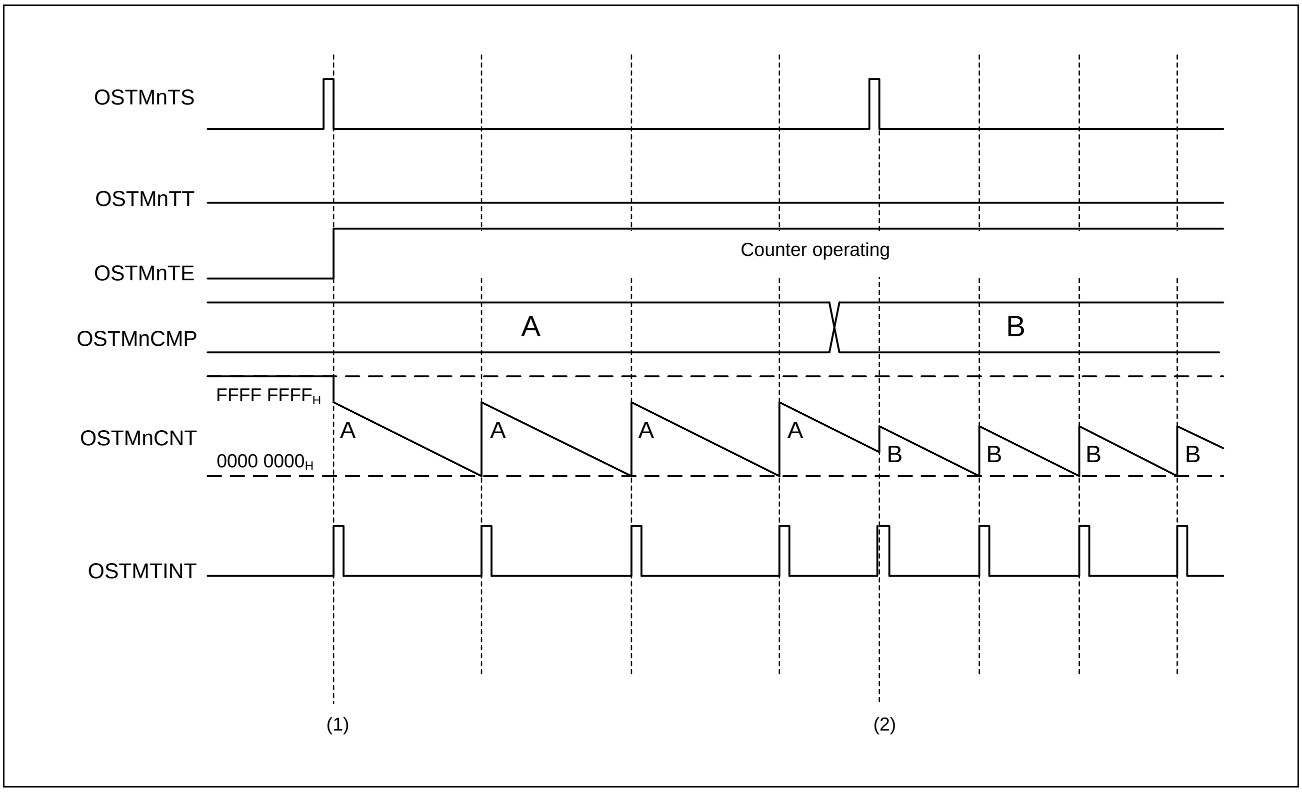


Figure 30.4 Timing Diagram of Forced Restart in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is started and stopped as described under **Figure 30.3, Timing Diagram of OSTM in Interval Timer Mode**.

(2) Setting OSTMnTS.OSTMnTS = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated when counting starts.

30.4.2.2 Operation when OSTMnCMP = 0000 0000_H

When OSTMnCMP = 0000 0000_H, OSTM behaves as follows.

- When the counter is enabled, the OSTMTINT interrupt request is always set to 1.

The following figure shows operations of OSTM when OSTMnCMP = 0000 0000_H, and counter-start interrupts are enabled.

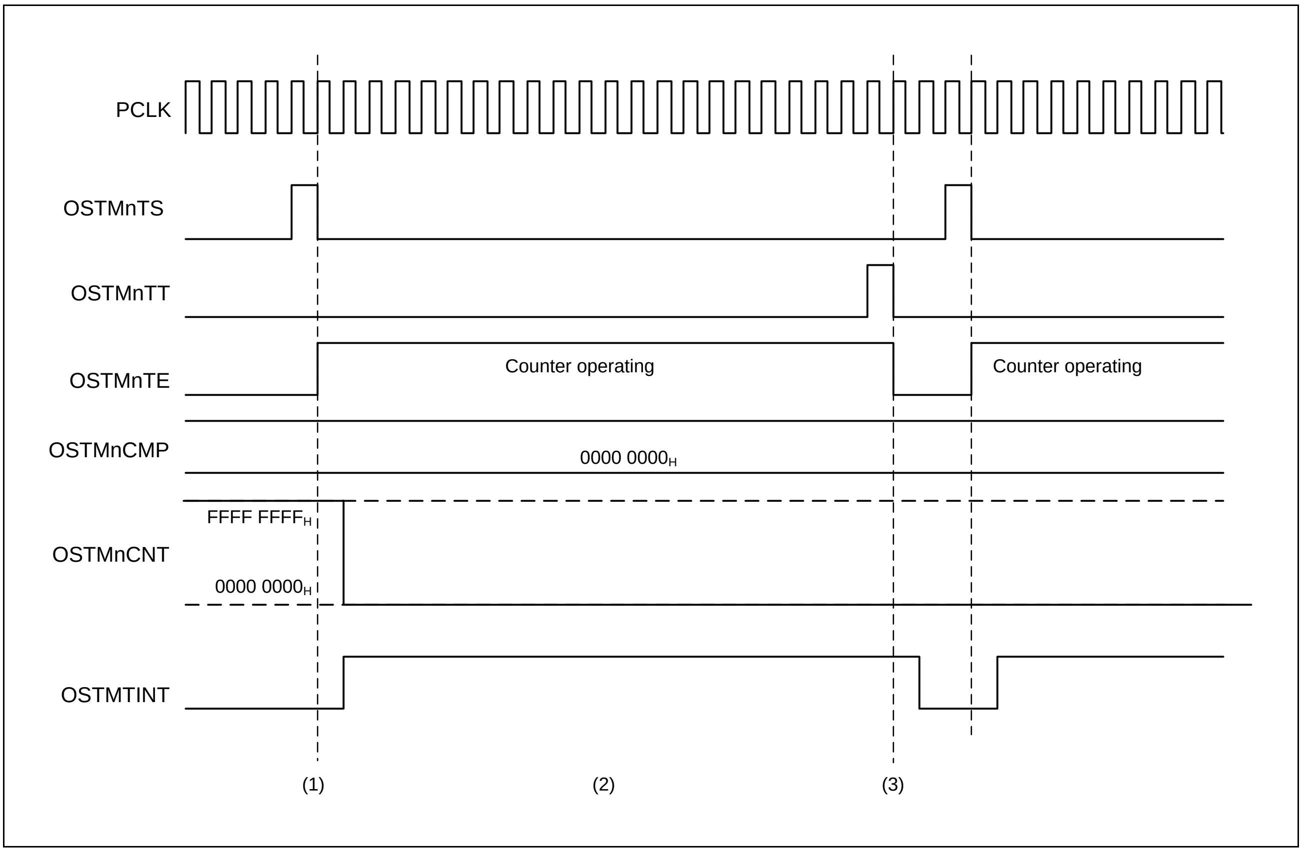


Figure 30.5 Timing Diagram when OSTMnCMP = 0000 0000_H in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000_H is retained in OSTMnCMP.

(2) The OSTMTINT interrupt request is continuously asserted.

(3) After the counter stops, the OSTMTINT interrupt request signal is deasserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

30.4.2.3 Setting Procedure for Interval Timer Mode

The setting procedure in interval timer mode after reset release is described below:

Setting Procedure

(1) Set the start value of the counter in the OSTMnCMP register.

(2) Select interval timer mode by setting the OSTMnCTL.OSTMnMD1 bit to 0.

(3) Enable or disable interrupts when counting starts (OSTMnCTL.OSTMnMD0).

30.4.3 Free-Run Compare Mode

30.4.3.1 Basic Operation in Free-Run Compare Mode

In free-run compare mode, the counter counts up from 0000 0000_H to FFFF $FFFF_H$. When the value of the OSTMnCMP register matches the current counter value, an OSTMTINT interrupt request is output.

In free-run compare mode, set OSTMnCTL.OSTMnMD1 = 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1).

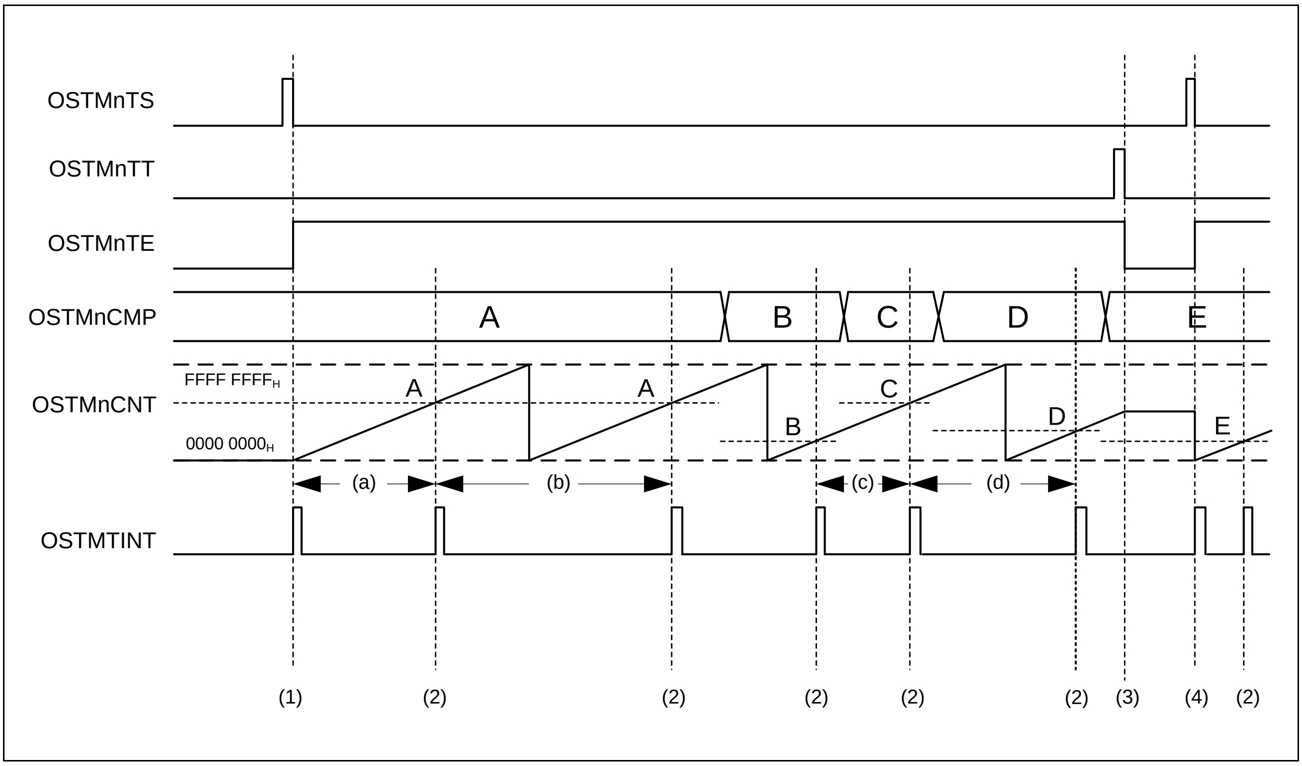


Figure 30.6 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1.
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000_H to FFFF $FFFF_H$. The OSTMnCNT register indicates the counter value.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated at the start of counting.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMTINT interrupt request is generated.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from 0000 0000_H when OSTMnTS.OSTMnTS = 1.

OSTMTINT Period

The OSTMTINT generation period is different depending on the starting time. If OSTMnCMP is rewritten during operation, the period is changed according to the size of the new and old compare values.

Table 30.24 OSTMTINT Generation Timing

Old Value for Comparison	New Value for Comparison	Counter Value at Time of Rewriting	Period of OSTMTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × count clock period	(a)
A	A	No rewriting	(FFFF FFFF_H + 1) × count clock period	(b)
B	C > B	B < counter value < C	(C – B) × count clock period	(c)
C	D < C	Counter value > D,C	(FFFF FFFF_H – C + D + 1) × count clock period	(d)

Forced Restart

Forced restarting of the counter does not proceed if the OSTMnTS.OSTMnTS bit is set during counting. The counter ignores the attempted setting and continues counting.

30.4.3.2 Operation when OSTMnCMP = 0000 0000_H

The following figure shows the operation of OSTM when OSTMnCMP = 0000 0000_H, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1).

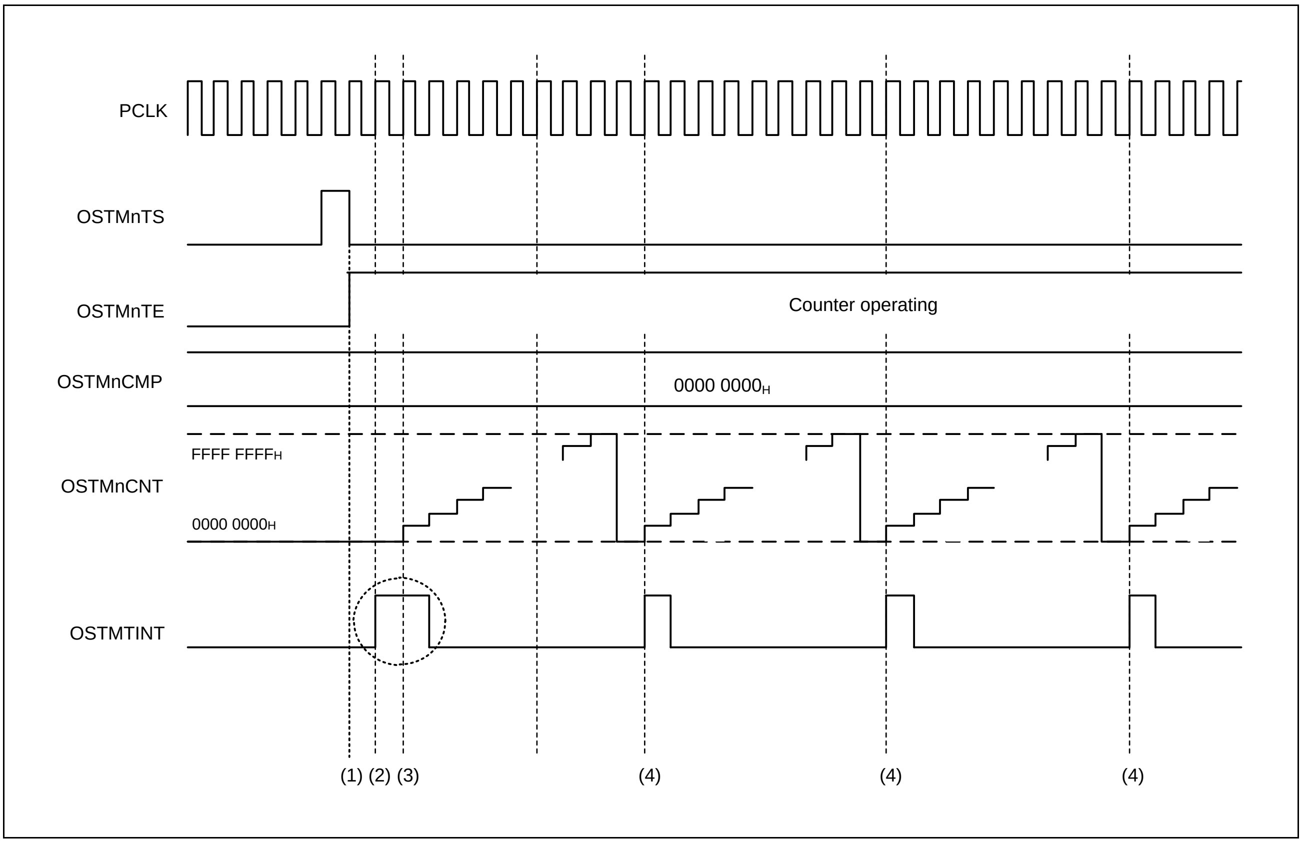


Figure 30.7 Timing Diagram when OSTMnCMP = 0000 0000_H in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from 0000 0000_H to FFFF $FFFF_H$.

(2) An OSTMTINT interrupt request is generated when counting starts.

(3) If the current counter value matches OSTMnCMP, an OSTMTINT interrupt request is generated.
If OSTMnCMP = 0000 0000_H as shown above, OSTMTINT is generated over two clock cycles.

(4) An OSTMTINT interrupt request is generated for each clock cycle (FFFF $FFFF_H$ + 1).

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

30.4.3.3 Setting Procedure for Free-Run Compare Mode

The setting procedure in free-run compare mode after reset release is described below:

Setting Procedure

(1) Set the compare value in the OSTMnCMP register.

(2) Select free-run compare mode by setting the OSTMnCTL.OSTMnMD1 bit to 1.

(3) Enable or disable interrupts when counting starts by the OSTMnCTL.OSTMnMD0 bit.

Section 31 Timer Array Unit B (TAUB)

This section contains a generic description of the timer array unit B (TAUB).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of TAUB.

31.1 Features of RH850/F1KH, RH850/F1KM TAUB

31.1.1 Number of Units and Channels

This microcontroller has the following number of TAUB units and channels.

Table 31.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	2	2	2
Name	TAUBn (n = 0, 1)	TAUBn (n = 0, 1)	TAUBn (n = 0, 1)

Table 31.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	2	2	2
Name	TAUBn (n = 0)	TAUBn (n = 0)	TAUBn (n = 0, 1)	TAUBn (n = 0, 1)	TAUBn (n = 0, 1)

Table 31.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	1	1
Name	—	—	TAUBn (n = 0)	TAUBn (n = 0)

Table 31.4 TAUBn Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name TAUBn	Number of Channels per Unit	RH850/F1KH-D8 176 Pins (32 ch)	RH850/F1KH-D8 233 Pins (32 ch)	RH850/F1KH-D8 324 Pins (32 ch)
TAUB0	16	✓	✓	✓
TAUB1	16	✓	✓	✓

Table 31.5 TAUBn Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name TAUBn	Number of Channels per Unit	RH850/F1KM-S4 100 Pins (16 ch)	RH850/F1KM-S4 144 Pins (16 ch)	RH850/F1KM-S4 176 Pins (32 ch)	RH850/F1KM-S4 233 Pins (32 ch)	RH850/F1KM-S4 272 Pins (32 ch)
TAUB0	16	✓	✓	✓	✓	✓
TAUB1	16	—	—	✓	✓	✓

Table 31.6 TAUBn Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name TAUBn	Number of Channels per Unit	RH850/F1KM-S1 48 Pins (—)	RH850/F1KM-S1 64 Pins (—)	RH850/F1KM-S1 80 Pins (16 ch)	RH850/F1KM-S1 100 Pins (16 ch)
TAUB0	16	—	—	✓	✓

Table 31.7 Indices (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual TAUB units are identified by the index “n”; for example, TAUBnTOM is the TAUBn channel output mode register.
m	The TAUB has 16 channels. Throughout this section, the individual channels are identified by the index “m”, thus a certain channel is denoted as CHm (m = 0 to 15). The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

31.1.2 Register Base Addresses

TAUBn base addresses are listed in the following table.

TAUBn register addresses are given as offsets from the base addresses.

Table 31.8 Register Base Addresses (RH850/F1KH-D8)

Name	Base Address
<TAUB0_base>	FFE3 0000_H
<TAUB1_base>	FFE3 1000_H

Table 31.9 Register Base Addresses (RH850/F1KM-S4)

Name	Base Address
<TAUB0_base>	FFE3 0000_H
<TAUB1_base>	FFE3 1000_H

Table 31.10 Register Base Address (RH850/F1KM-S1)

Name	Base Address
<TAUB0_base>	FFE3 0000_H

31.1.3 Clock Supply

The TAUBn clock supply is shown in the following table.

Table 31.11 TAUBn Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
TAUBn	PCLK	CKSCLK_IPERI2	Timer count clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI2	Bus clock

31.1.4 Interrupt Requests

TAUBn interrupt requests are listed in the following table.

Table 31.12 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
TAUB0			
INTTAUB0I0	Channel 0 interrupt	142	33 to 38
INTTAUB0I1	Channel 1 interrupt	143	33 to 38
INTTAUB0I2	Channel 2 interrupt	144	33 to 38
INTTAUB0I3	Channel 3 interrupt	145	33 to 38
INTTAUB0I4	Channel 4 interrupt	146	33 to 38
INTTAUB0I5	Channel 5 interrupt	147	33 to 38
INTTAUB0I6	Channel 6 interrupt	148	33 to 38
INTTAUB0I7	Channel 7 interrupt	149	33 to 38
INTTAUB0I8	Channel 8 interrupt	150	33 to 38
INTTAUB0I9	Channel 9 interrupt	151	33 to 38
INTTAUB0I10	Channel 10 interrupt	152	33 to 38
INTTAUB0I11	Channel 11 interrupt	153	33 to 38
INTTAUB0I12	Channel 12 interrupt	154	33 to 38
INTTAUB0I13	Channel 13 interrupt	155	33 to 38
INTTAUB0I14	Channel 14 interrupt	156	33 to 38
INTTAUB0I15	Channel 15 interrupt	157	33 to 38
TAUB1			
INTTAUB1I0	Channel 0 interrupt	256	52 to 57
INTTAUB1I1	Channel 1 interrupt	257	52 to 57
INTTAUB1I2	Channel 2 interrupt	258	52 to 57
INTTAUB1I3	Channel 3 interrupt	259	52 to 57
INTTAUB1I4	Channel 4 interrupt	260	52 to 57
INTTAUB1I5	Channel 5 interrupt	261	52 to 57
INTTAUB1I6	Channel 6 interrupt	262	52 to 57
INTTAUB1I7	Channel 7 interrupt	263	52 to 57
INTTAUB1I8	Channel 8 interrupt	264	52 to 57
INTTAUB1I9	Channel 9 interrupt	265	52 to 57
INTTAUB1I10	Channel 10 interrupt	266	52 to 57
INTTAUB1I11	Channel 11 interrupt	267	52 to 57
INTTAUB1I12	Channel 12 interrupt	268	52 to 57
INTTAUB1I13	Channel 13 interrupt	269	52 to 57
INTTAUB1I14	Channel 14 interrupt	270	52 to 57
INTTAUB1I15	Channel 15 interrupt	271	52 to 57

Table 31.13 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
TAUB0			
INTTAUB0I0	Channel 0 interrupt	142	33 to 38
INTTAUB0I1	Channel 1 interrupt	143	33 to 38
INTTAUB0I2	Channel 2 interrupt	144	33 to 38
INTTAUB0I3	Channel 3 interrupt	145	33 to 38
INTTAUB0I4	Channel 4 interrupt	146	33 to 38
INTTAUB0I5	Channel 5 interrupt	147	33 to 38
INTTAUB0I6	Channel 6 interrupt	148	33 to 38
INTTAUB0I7	Channel 7 interrupt	149	33 to 38
INTTAUB0I8	Channel 8 interrupt	150	33 to 38
INTTAUB0I9	Channel 9 interrupt	151	33 to 38
INTTAUB0I10	Channel 10 interrupt	152	33 to 38
INTTAUB0I11	Channel 11 interrupt	153	33 to 38
INTTAUB0I12	Channel 12 interrupt	154	33 to 38
INTTAUB0I13	Channel 13 interrupt	155	33 to 38
INTTAUB0I14	Channel 14 interrupt	156	33 to 38
INTTAUB0I15	Channel 15 interrupt	157	33 to 38
TAUB1			
INTTAUB1I0	Channel 0 interrupt	256	52 to 57
INTTAUB1I1	Channel 1 interrupt	257	52 to 57
INTTAUB1I2	Channel 2 interrupt	258	52 to 57
INTTAUB1I3	Channel 3 interrupt	259	52 to 57
INTTAUB1I4	Channel 4 interrupt	260	52 to 57
INTTAUB1I5	Channel 5 interrupt	261	52 to 57
INTTAUB1I6	Channel 6 interrupt	262	52 to 57
INTTAUB1I7	Channel 7 interrupt	263	52 to 57
INTTAUB1I8	Channel 8 interrupt	264	52 to 57
INTTAUB1I9	Channel 9 interrupt	265	52 to 57
INTTAUB1I10	Channel 10 interrupt	266	52 to 57
INTTAUB1I11	Channel 11 interrupt	267	52 to 57
INTTAUB1I12	Channel 12 interrupt	268	52 to 57
INTTAUB1I13	Channel 13 interrupt	269	52 to 57
INTTAUB1I14	Channel 14 interrupt	270	52 to 57
INTTAUB1I15	Channel 15 interrupt	271	52 to 57

Table 31.14 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
TAUB0			
INTTAUB0I0	Channel 0 interrupt	142	33 to 38
INTTAUB0I1	Channel 1 interrupt	143	33 to 38
INTTAUB0I2	Channel 2 interrupt	144	33 to 38
INTTAUB0I3	Channel 3 interrupt	145	33 to 38
INTTAUB0I4	Channel 4 interrupt	146	33 to 38
INTTAUB0I5	Channel 5 interrupt	147	33 to 38
INTTAUB0I6	Channel 6 interrupt	148	33 to 38
INTTAUB0I7	Channel 7 interrupt	149	33 to 38
INTTAUB0I8	Channel 8 interrupt	150	33 to 38
INTTAUB0I9	Channel 9 interrupt	151	33 to 38
INTTAUB0I10	Channel 10 interrupt	152	33 to 38
INTTAUB0I11	Channel 11 interrupt	153	33 to 38
INTTAUB0I12	Channel 12 interrupt	154	33 to 38
INTTAUB0I13	Channel 13 interrupt	155	33 to 38
INTTAUB0I14	Channel 14 interrupt	156	33 to 38
INTTAUB0I15	Channel 15 interrupt	157	33 to 38

31.1.5 Reset Sources

TAUBn reset sources are listed in the following table. TAUBn is initialized by these reset sources.

Table 31.15 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
TAUBn	All reset sources (ISORES)

31.1.6 External input/output Signals

External input/output signals of TAUBn are listed below.

Table 31.16 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input*[1]	TAUB0I0
TAUBTTIN1	Channel 1 input*[1]	TAUB0I1
TAUBTTIN2	Channel 2 input*[1]	TAUB0I2
TAUBTTIN3	Channel 3 input*[1]	TAUB0I3
TAUBTTIN4	Channel 4 input*[1]	TAUB0I4
TAUBTTIN5	Channel 5 input*[1]	TAUB0I5
TAUBTTIN6	Channel 6 input*[1]	TAUB0I6
TAUBTTIN7	Channel 7 input*[1]	TAUB0I7
TAUBTTIN8	Channel 8 input*[1]	TAUB0I8
TAUBTTIN9	Channel 9 input*[1]	TAUB0I9
TAUBTTIN10	Channel 10 input*[1]	TAUB0I10
TAUBTTIN11	Channel 11 input*[1]	TAUB0I11
TAUBTTIN12	Channel 12 input*[1]	TAUB0I12
TAUBTTIN13	Channel 13 input*[1]	TAUB0I13
TAUBTTIN14	Channel 14 input*[1]	TAUB0I14
TAUBTTIN15	Channel 15 input*[1]	TAUB0I15
TAUBTTOUT0	Channel 0 output	TAUB0O0
TAUBTTOUT1	Channel 1 output	TAUB0O1
TAUBTTOUT2	Channel 2 output	TAUB0O2
TAUBTTOUT3	Channel 3 output	TAUB0O3
TAUBTTOUT4	Channel 4 output	TAUB0O4
TAUBTTOUT5	Channel 5 output	TAUB0O5
TAUBTTOUT6	Channel 6 output	TAUB0O6
TAUBTTOUT7	Channel 7 output	TAUB0O7
TAUBTTOUT8	Channel 8 output	TAUB0O8
TAUBTTOUT9	Channel 9 output	TAUB0O9
TAUBTTOUT10	Channel 10 output	TAUB0O10
TAUBTTOUT11	Channel 11 output	TAUB0O11
TAUBTTOUT12	Channel 12 output	TAUB0O12
TAUBTTOUT13	Channel 13 output	TAUB0O13
TAUBTTOUT14	Channel 14 output	TAUB0O14
TAUBTTOUT15	Channel 15 output	TAUB0O15

Table 31.16 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUB1		
TAUBTTIN0	Channel 0 input*[1]	TAUB1I0
TAUBTTIN1	Channel 1 input*[1]	TAUB1I1
TAUBTTIN2	Channel 2 input*[1]	TAUB1I2
TAUBTTIN3	Channel 3 input*[1]	TAUB1I3
TAUBTTIN4	Channel 4 input*[1]	TAUB1I4
TAUBTTIN5	Channel 5 input*[1]	TAUB1I5
TAUBTTIN6	Channel 6 input*[1]	TAUB1I6
TAUBTTIN7	Channel 7 input*[1]	TAUB1I7
TAUBTTIN8	Channel 8 input*[1]	TAUB1I8
TAUBTTIN9	Channel 9 input*[1]	TAUB1I9
TAUBTTIN10	Channel 10 input*[1]	TAUB1I10
TAUBTTIN11	Channel 11 input*[1]	TAUB1I11
TAUBTTIN12	Channel 12 input*[1]	TAUB1I12
TAUBTTIN13	Channel 13 input*[1]	TAUB1I13
TAUBTTIN14	Channel 14 input*[1]	TAUB1I14
TAUBTTIN15	Channel 15 input*[1]	TAUB1I15
TAUBTTOUT0	Channel 0 output	TAUB1O0
TAUBTTOUT1	Channel 1 output	TAUB1O1
TAUBTTOUT2	Channel 2 output	TAUB1O2
TAUBTTOUT3	Channel 3 output	TAUB1O3
TAUBTTOUT4	Channel 4 output	TAUB1O4
TAUBTTOUT5	Channel 5 output	TAUB1O5
TAUBTTOUT6	Channel 6 output	TAUB1O6
TAUBTTOUT7	Channel 7 output	TAUB1O7
TAUBTTOUT8	Channel 8 output	TAUB1O8
TAUBTTOUT9	Channel 9 output	TAUB1O9
TAUBTTOUT10	Channel 10 output	TAUB1O10
TAUBTTOUT11	Channel 11 output	TAUB1O11
TAUBTTOUT12	Channel 12 output	TAUB1O12
TAUBTTOUT13	Channel 13 output	TAUB1O13
TAUBTTOUT14	Channel 14 output	TAUB1O14
TAUBTTOUT15	Channel 15 output	TAUB1O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**.

Table 31.17 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input*1	TAUB0I0
TAUBTTIN1	Channel 1 input*1	TAUB0I1
TAUBTTIN2	Channel 2 input*1	TAUB0I2
TAUBTTIN3	Channel 3 input*1	TAUB0I3
TAUBTTIN4	Channel 4 input*1	TAUB0I4
TAUBTTIN5	Channel 5 input*1	TAUB0I5
TAUBTTIN6	Channel 6 input*1	TAUB0I6
TAUBTTIN7	Channel 7 input*1	TAUB0I7
TAUBTTIN8	Channel 8 input*1	TAUB0I8
TAUBTTIN9	Channel 9 input*1	TAUB0I9
TAUBTTIN10	Channel 10 input*1	TAUB0I10
TAUBTTIN11	Channel 11 input*1	TAUB0I11
TAUBTTIN12	Channel 12 input*1	TAUB0I12
TAUBTTIN13	Channel 13 input*1	TAUB0I13
TAUBTTIN14	Channel 14 input*1	TAUB0I14
TAUBTTIN15	Channel 15 input*1	TAUB0I15
TAUBTTOUT0	Channel 0 output	TAUB0O0
TAUBTTOUT1	Channel 1 output	TAUB0O1
TAUBTTOUT2	Channel 2 output	TAUB0O2
TAUBTTOUT3	Channel 3 output	TAUB0O3
TAUBTTOUT4	Channel 4 output	TAUB0O4
TAUBTTOUT5	Channel 5 output	TAUB0O5
TAUBTTOUT6	Channel 6 output	TAUB0O6
TAUBTTOUT7	Channel 7 output	TAUB0O7
TAUBTTOUT8	Channel 8 output	TAUB0O8
TAUBTTOUT9	Channel 9 output	TAUB0O9
TAUBTTOUT10	Channel 10 output	TAUB0O10
TAUBTTOUT11	Channel 11 output	TAUB0O11
TAUBTTOUT12	Channel 12 output	TAUB0O12
TAUBTTOUT13	Channel 13 output	TAUB0O13
TAUBTTOUT14	Channel 14 output	TAUB0O14
TAUBTTOUT15	Channel 15 output	TAUB0O15

Table 31.17 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUB1		
TAUBTTIN0	Channel 0 input*[1]	TAUB1I0
TAUBTTIN1	Channel 1 input*[1]	TAUB1I1
TAUBTTIN2	Channel 2 input*[1]	TAUB1I2
TAUBTTIN3	Channel 3 input*[1]	TAUB1I3
TAUBTTIN4	Channel 4 input*[1]	TAUB1I4
TAUBTTIN5	Channel 5 input*[1]	TAUB1I5
TAUBTTIN6	Channel 6 input*[1]	TAUB1I6
TAUBTTIN7	Channel 7 input*[1]	TAUB1I7
TAUBTTIN8	Channel 8 input*[1]	TAUB1I8
TAUBTTIN9	Channel 9 input*[1]	TAUB1I9
TAUBTTIN10	Channel 10 input*[1]	TAUB1I10
TAUBTTIN11	Channel 11 input*[1]	TAUB1I11
TAUBTTIN12	Channel 12 input*[1]	TAUB1I12
TAUBTTIN13	Channel 13 input*[1]	TAUB1I13
TAUBTTIN14	Channel 14 input*[1]	TAUB1I14
TAUBTTIN15	Channel 15 input*[1]	TAUB1I15
TAUBTTOUT0	Channel 0 output	TAUB1O0
TAUBTTOUT1	Channel 1 output	TAUB1O1
TAUBTTOUT2	Channel 2 output	TAUB1O2
TAUBTTOUT3	Channel 3 output	TAUB1O3
TAUBTTOUT4	Channel 4 output	TAUB1O4
TAUBTTOUT5	Channel 5 output	TAUB1O5
TAUBTTOUT6	Channel 6 output	TAUB1O6
TAUBTTOUT7	Channel 7 output	TAUB1O7
TAUBTTOUT8	Channel 8 output	TAUB1O8
TAUBTTOUT9	Channel 9 output	TAUB1O9
TAUBTTOUT10	Channel 10 output	TAUB1O10
TAUBTTOUT11	Channel 11 output	TAUB1O11
TAUBTTOUT12	Channel 12 output	TAUB1O12
TAUBTTOUT13	Channel 13 output	TAUB1O13
TAUBTTOUT14	Channel 14 output	TAUB1O14
TAUBTTOUT15	Channel 15 output	TAUB1O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2B.12, Noise Filter & Edge/Level Detector**.

Table 31.18 External Input/Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input*[1]	TAUB0I0
TAUBTTIN1	Channel 1 input*[1]	TAUB0I1
TAUBTTIN2	Channel 2 input*[1]	TAUB0I2
TAUBTTIN3	Channel 3 input*[1]	TAUB0I3
TAUBTTIN4	Channel 4 input*[1]	TAUB0I4
TAUBTTIN5	Channel 5 input*[1]	TAUB0I5
TAUBTTIN6	Channel 6 input*[1]	TAUB0I6
TAUBTTIN7	Channel 7 input*[1]	TAUB0I7
TAUBTTIN8	Channel 8 input*[1]	TAUB0I8
TAUBTTIN9	Channel 9 input*[1]	TAUB0I9
TAUBTTIN10	Channel 10 input*[1]	TAUB0I10
TAUBTTIN11	Channel 11 input*[1]	TAUB0I11
TAUBTTIN12	Channel 12 input*[1]	TAUB0I12
TAUBTTIN13	Channel 13 input*[1]	TAUB0I13
TAUBTTIN14	Channel 14 input*[1]	TAUB0I14
TAUBTTIN15	Channel 15 input*[1]	TAUB0I15
TAUBTTOUT0	Channel 0 output	TAUB0O0
TAUBTTOUT1	Channel 1 output	TAUB0O1
TAUBTTOUT2	Channel 2 output	TAUB0O2
TAUBTTOUT3	Channel 3 output	TAUB0O3
TAUBTTOUT4	Channel 4 output	TAUB0O4
TAUBTTOUT5	Channel 5 output	TAUB0O5
TAUBTTOUT6	Channel 6 output	TAUB0O6
TAUBTTOUT7	Channel 7 output	TAUB0O7
TAUBTTOUT8	Channel 8 output	TAUB0O8
TAUBTTOUT9	Channel 9 output	TAUB0O9
TAUBTTOUT10	Channel 10 output	TAUB0O10
TAUBTTOUT11	Channel 11 output	TAUB0O11
TAUBTTOUT12	Channel 12 output	TAUB0O12
TAUBTTOUT13	Channel 13 output	TAUB0O13
TAUBTTOUT14	Channel 14 output	TAUB0O14
TAUBTTOUT15	Channel 15 output	TAUB0O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2C.12, Noise Filter & Edge/Level Detector**.

31.1.7 TAUBn Input Selection

The output from port TAUBnIm (n = 0, 1, m = 0 to 15) can be input to TAUBTTINm (m = 0 to 15) as shown in the following figure.

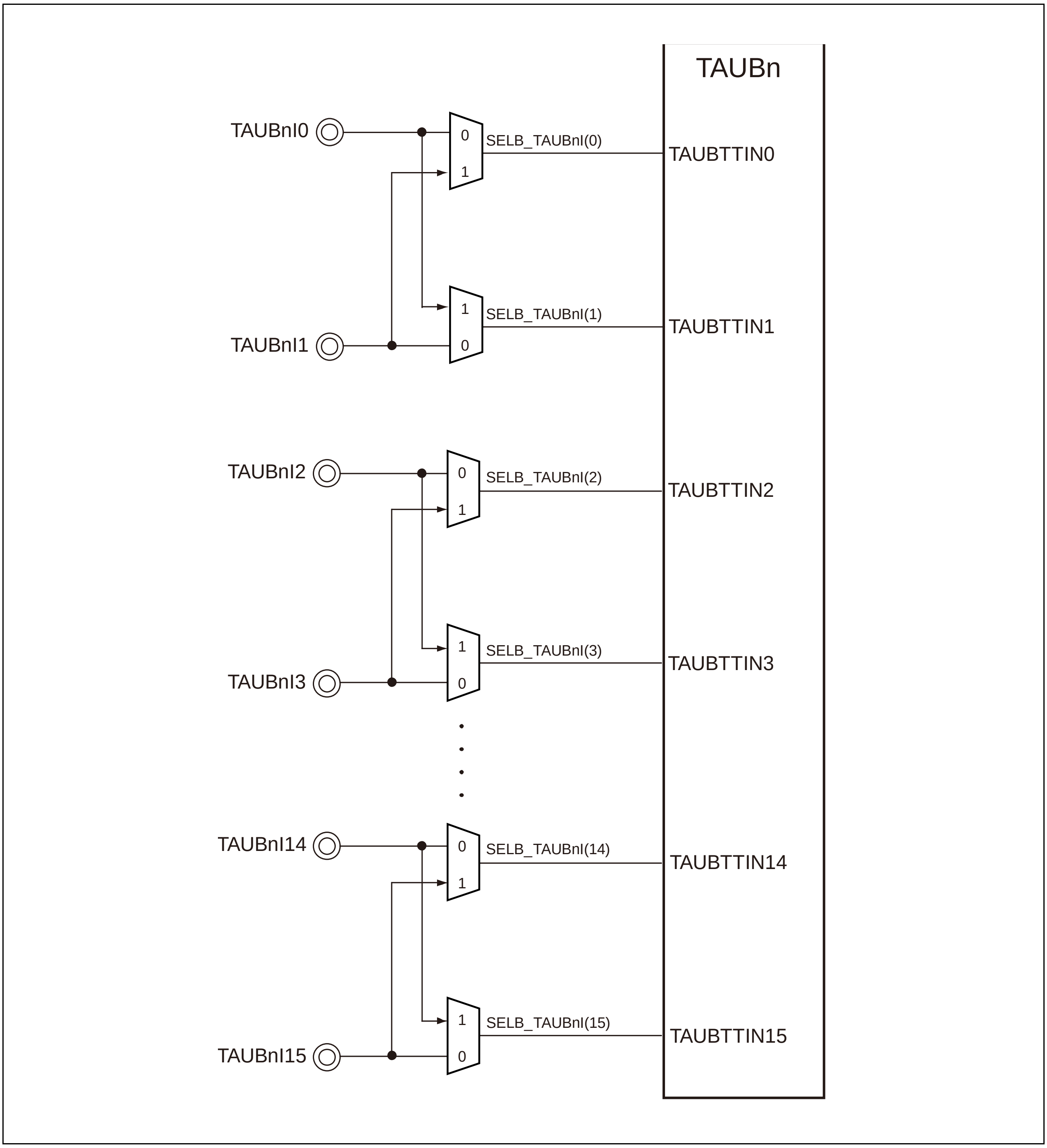


Figure 31.1 TAUBn Input/Output and Interrupt Request Signals

The following table shows the input signals to several TAUBn inputs.

Table 31.19 TAUBn Input Selection (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Input Signal	Function	Settings
TAUBTTIN[m]	Port TAUBnI[m]	SELB_TAUBnI[m] = 0
	Port TAUBnI[m + 1]	SELB_TAUBnI[m] = 1
TAUBTTIN[m + 1]	Port TAUBnI[m + 1]	SELB_TAUBnI[m + 1] = 0
	Port TAUBnI[m]	SELB_TAUBnI[m + 1] = 1

Note: m = 0, 2, 4, 6, 8, 10, 12, 14

31.1.7.1 List of Registers

Input signal selection register is listed in the following table.

Table 31.20 List of Registers (RH850/F1KH-D8)

Module Name	Register Name	Symbol	Address
TAUBn input selection registers			
SL_TAUBn	TAUBTTINm input signal selection register	SELB_TAUB0I	FFE3 2000_H
	TAUBTTINm input signal selection register	SELB_TAUB1I	FFE3 3000_H

Table 31.21 List of Registers (RH850/F1KM-S4)

Module Name	Register Name	Symbol	Address
TAUBn input selection registers			
SL_TAUBn	TAUBTTINm input signal selection register	SELB_TAUB0I	FFE3 2000_H
	TAUBTTINm input signal selection register	SELB_TAUB1I	FFE3 3000_H

Table 31.22 List of Register (RH850/F1KM-S1)

Module Name	Register Name	Symbol	Address
TAUBn input selection register			
SL_TAUBn	TAUBTTINm input signal selection register	SELB_TAUB0I	FFE3 2000_H

31.1.7.2 SELB_TAUB0I — TAUBTTINm Input Signal Selection Register

This register selects the input signals to several TAUBTTINm inputs.

Access: This register can be read or written in 16-bit units.

Address: FFE3 2000$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SELB_ TAUB0I 15	SELB_ TAUB0I 14	SELB_ TAUB0I 13	SELB_ TAUB0I 12	SELB_ TAUB0I 11	SELB_ TAUB0I 10	SELB_ TAUB0I 9	SELB_ TAUB0I 8	SELB_ TAUB0I 7	SELB_ TAUB0I 6	SELB_ TAUB0I 5	SELB_ TAUB0I 4	SELB_ TAUB0I 3	SELB_ TAUB0I 2	SELB_ TAUB0I 1	SELB_ TAUB0I 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.23 SELB_TAUB0I Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>SELB_TAUB0Im
(m = 0 to 15)</td><td>Selection of TAUBTTINm input signal
<table>
<tr><th>TAUB Input</th><th>Bit[m+1]</th><th>Bit[m]</th><th>Input signal</th></tr>
<tr><td>TAUBTTIN[m]</td><td>x</td><td>0</td><td>Selection of port TAUB0I[m]</td></tr>
<tr><td></td><td>x</td><td>1</td><td>Selection of port TAUB0I[m + 1]</td></tr>
<tr><td>TAUBTTIN[m+1]</td><td>0</td><td>x</td><td>Selection of port TAUB0I[m + 1]</td></tr>
<tr><td></td><td>1</td><td>x</td><td>Selection of port TAUB0I[m]</td></tr>
</table>
NOTE: m = 0, 2, 4, 6, 8, 10, 12, 14</td></tr>
</table>

CAUTION

Do not change the input signal of the each channel during the timer counting.

31.1.7.3 SELB_TAUB1I — TAUBTTINm Input Signal Selection Register

This register selects the input signals to several TAUBTTINm inputs.

Access: This register can be read or written in 16-bit units.

Address: FFE3 3000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SELB_ TAUB1I 15	SELB_ TAUB1I 14	SELB_ TAUB1I 13	SELB_ TAUB1I 12	SELB_ TAUB1I 11	SELB_ TAUB1I 10	SELB_ TAUB1I 9	SELB_ TAUB1I 8	SELB_ TAUB1I 7	SELB_ TAUB1I 6	SELB_ TAUB1I 5	SELB_ TAUB1I 4	SELB_ TAUB1I 3	SELB_ TAUB1I 2	SELB_ TAUB1I 1	SELB_ TAUB1I 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.24 SELB_TAUB1I Register Contents

Bit Position	Bit Name	Function
15 to 0	SELB_TAUB1Im (m = 0 to 15)	Selection of TAUBTTINm input signal

TAUB Input	Bit[m+1]	Bit[m]	Input signal
TAUBTTIN[m]	x	0	Selection of port TAUB1I[m]
	x	1	Selection of port TAUB1I[m + 1]
TAUBTTIN[m+1]	0	x	Selection of port TAUB1I[m + 1]
	1	x	Selection of port TAUB1I[m]

NOTE: m = 0, 2, 4, 6, 8, 10, 12, 14

CAUTION

Do not change the input signal of the each channel during the timer counting.

31.2 Overview

31.2.1 Functional Overview

The TAUB has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUB is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUBnCNTm and a 16-bit data register TAUBnCDRm to hold the count start value and compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels. The synchronous operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

31.2.2 Terms

In this section, the following terms are used:

Independent channel operation function/synchronous channel operation function

TAUB has 16 channels, and provides an independent channel operation function whereby individual channels operate independently and a synchronous channel operation function whereby multiple channels operate in combination.

- The independent channel operation function can be used by any channel independently of all other channels.
- The synchronous channel operation function is implemented using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.
A channel group has one master channel and one or more slave channels.

Upper/lower channel

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:
For channel 5, channel 3 is an upper channel and channel 9 is a lower channel. Channel 0 is the highest channel and channel 15 is the lowest channel.

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operating mode (selected with the TAUBnCMORm.TAUBnMD[4:0] bits)
- Counter start enable (TAUBnTS.TAUBnTSm) and counter stop (TAUBnTT.TAUBnTTm)
  When counter start is enabled, status flag TAUBnTE.TAUBnTEm is set.
- Count direction (up/down) (can be controlled by the master channel)

Trigger selector

The counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

31.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 31.25 Functional List of TAUB Operations

Operation Function		Setting Example
Independent Channel Operation Functions		**Section 31.12**
	Interval Timer Function	**Section 31.12.1**
	TAUBTTINm Input Interval Timer Function	**Section 31.12.2**
	Clock Divide Function	**Section 31.12.3**
	External Event Count Function	**Section 31.12.4**
	One-Pulse Output Function	**Section 31.12.5**
	TAUBTTINm Input Pulse Interval Measurement Function	**Section 31.12.6**
	TAUBTTINm Input Signal Width Measurement Function	**Section 31.12.7**
	TAUBTTINm Input Position Detection Function	**Section 31.12.8**
	TAUBTTINm Input Period Count Detection Function	**Section 31.12.9**
	TAUBTTINm Input Pulse Interval Judgment Function	**Section 31.12.10**
	TAUBTTINm Input Signal Width Judgment Function	**Section 31.12.11**
	Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)	**Section 31.12.12**
	Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	**Section 31.12.13**
Independent Channel Simultaneous Rewrite Functions		**Section 31.13**
	Simultaneous Rewrite Trigger Generation Function Type 1	**Section 31.13.1**
Synchronous Channel Operation Functions		**Section 31.14**
	PWM Output Function	**Section 31.14.1**
	One-Shot Pulse Output Function	**Section 31.14.2**
	Delay Pulse Output Function	**Section 31.14.3**
	AD Conversion Trigger Output Function Type 1	**Section 31.14.4**
	Triangle PWM Output Function	**Section 31.14.5**
	Triangle PWM Output Function with Dead Time	**Section 31.14.6**
	A/D Conversion Trigger Output Function Type 2	**Section 31.14.7**

31.2.4 TAUB I/O and Interrupt Request Signals

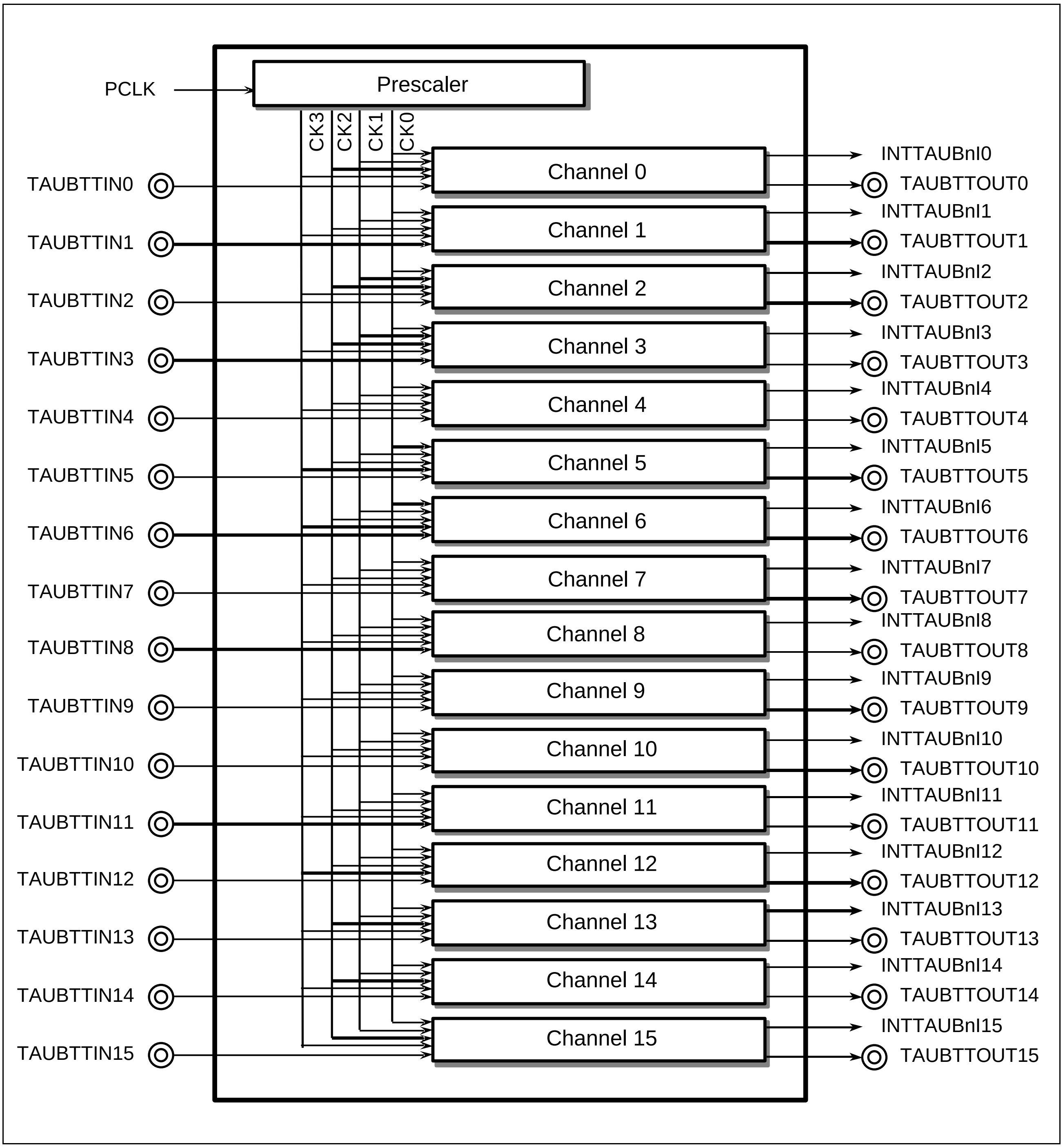


Figure 31.2 TAUB Input/Output and Interrupt Request Signals

31.2.5 Block Diagram

The following figure shows the main components of the TAUB.

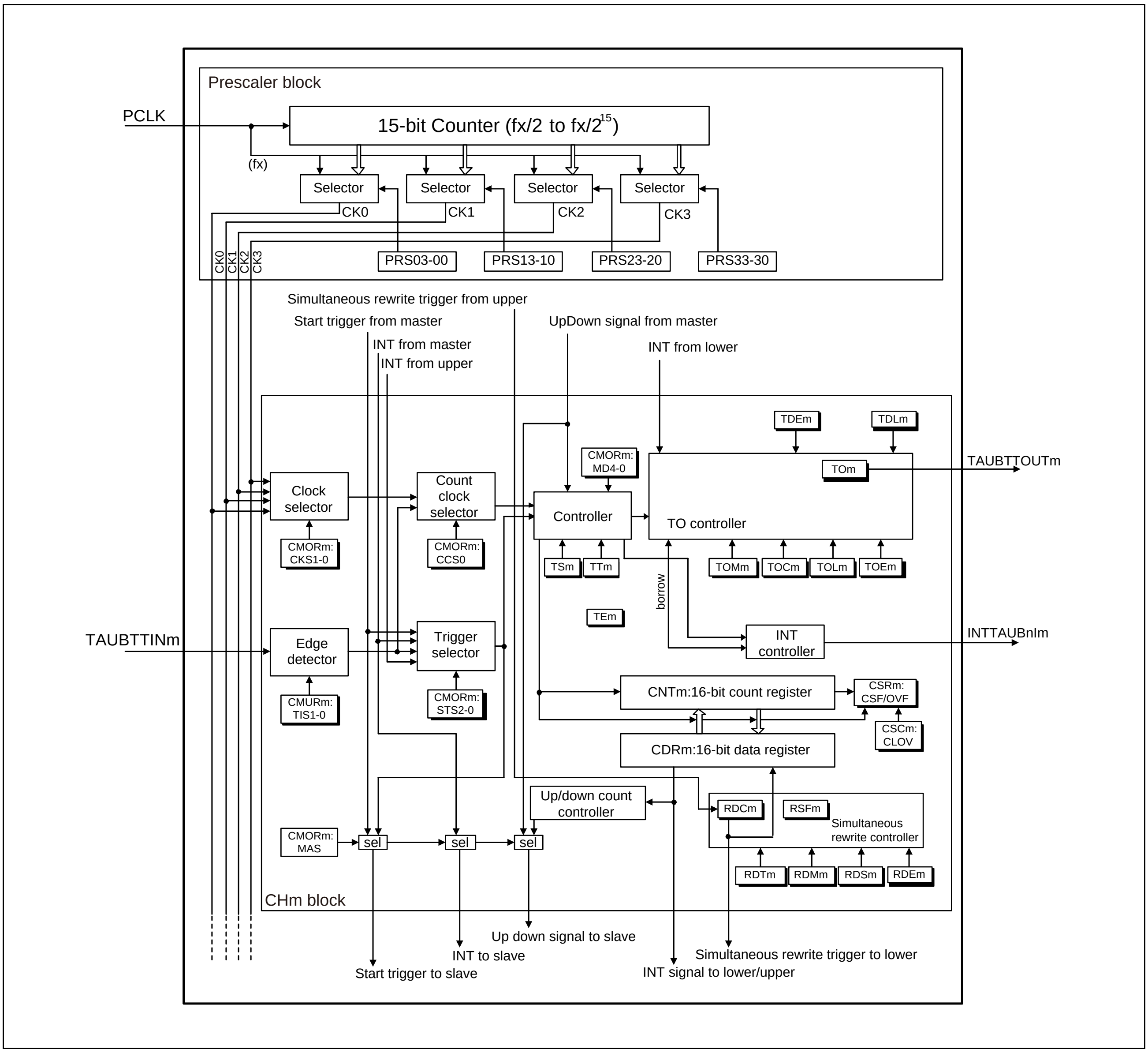


Figure 31.3 Block Diagram of the TAUB

The prefix “TAUBn” has been omitted from the register names for the sake of clarity in the above figure.

31.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUBnCMORm.TAUBnMD[4:0])
- Counter start enable (TAUBnTS.TAUBnTSm) and counter stop (TAUBnTT.TAUBnTTm)
  When counter start is enabled, status flag TAUBnTE.TAUBnTEm is set.
- Count direction (up/down) (can be controlled by the master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Input of TAUBTTINm valid edge
- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used if multiple channels like synchronous operation functions are used. The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUBnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

31.3 Registers

31.3.1 List of Registers

TAUB registers are listed in the following table.

For details about <TAUBn_base>, see **Section 31.1.2, Register Base Addresses**.

Table 31.26 List of Registers

Module Name	Register Name	Symbol	Address
TAUBn prescaler registers			
TAUBn	TAUBn prescaler clock select register	TAUBnTPS	<TAUBn_base> + 240_H
TAUBn control registers			
TAUBn	TAUBn channel data register	TAUBnCDRm	<TAUBn_base> + 0_H + m × 4_H
	TAUBn channel counter register	TAUBnCNTm	<TAUBn_base> + 80_H + m × 4_H
	TAUBn channel mode OS register	TAUBnCMORm	<TAUBn_base> + 200_H + m × 4_H
	TAUBn channel mode user register	TAUBnCMURm	<TAUBn_base> + $C0_H$ + m × 4_H
	TAUBn channel status register	TAUBnCSRm	<TAUBn_base> + 140_H + m × 4_H
	TAUBn channel status clear trigger register	TAUBnCSCm	<TAUBn_base> + 180_H + m × 4_H
	TAUBn channel start trigger register	TAUBnTS	<TAUBn_base> + $1C4_H$
	TAUBn channel enable status register	TAUBnTE	<TAUBn_base> + $1C0_H$
	TAUBn channel stop trigger register	TAUBnTT	<TAUBn_base> + $1C8_H$
TAUBn output registers			
TAUBn	TAUBn channel output enable register	TAUBnTOE	<TAUBn_base> + $5C_H$
	TAUBn channel output register	TAUBnTO	<TAUBn_base> + 58_H
	TAUBn channel output mode register	TAUBnTOM	<TAUBn_base> + 248_H
	TAUBn channel output configuration register	TAUBnTOC	<TAUBn_base> + $24C_H$
	TAUBn channel output active level register	TAUBnTOL	<TAUBn_base> + 040_H
	TAUBn channel dead time output enable register	TAUBnTDE	<TAUBn_base> + 250_H
	TAUBn channel dead time output level register	TAUBnTDL	<TAUBn_base> + 54_H
TAUBn reload data registers			
TAUBn	TAUBn channel reload data enable register	TAUBnRDE	<TAUBn_base> + 260_H
	TAUBn channel reload data mode register	TAUBnRDM	<TAUBn_base> + 264_H
	TAUBn channel reload data control channel select register	TAUBnRDS	<TAUBn_base> + 268_H
	TAUBn channel reload data control register	TAUBnRDC	<TAUBn_base> + $26C_H$
	TAUBn channel reload data trigger register	TAUBnRDT	<TAUBn_base> + 44_H
	TAUBn channel reload status register	TAUBnRSF	<TAUBn_base> + 48_H
TAUBn emulation register			
TAUBn	TAUBn emulation register	TAUBnEMU	<TAUBn_base> + 290_H

31.3.2 Details of TAUBn Prescaler Registers

31.3.2.1 TAUBnTPS — TAUBn Prescaler Clock Select Register

This register specifies the clocks CK0, CK1, CK2, and CK3 for all channels of the PCLK prescaler.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 240_H

Value after reset: $FFFF_H$

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="4">TAUBnPRS3[3:0]</td><td colspan="4">TAUBnPRS2[3:0]</td><td colspan="4">TAUBnPRS1[3:0]</td><td colspan="4">TAUBnPRS0[3:0]</td></tr>
<tr><td>Value after reset</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>

Table 31.27 TAUBnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 12</td><td>TAUBnPRS3[3:0]</td><td>Specifies the CK3 clock.
<table>
<tr><th>TAUBnPRS3[3:0]</th><th>CK3 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK3 are stopped (TAUBnTE.TAUBnTEm = 0).</td></tr>
</table>

Table 31.27 TAUBnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUBnPRS2[3:0]</td><td>Specifies the CK2 clock.
<table>
<tr><th>TAUBnPRS2[3:0]</th><th>CK2 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK2 are stopped (TAUBnTE.TAUBnTEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUBnPRS1[3:0]</td><td>Specifies the CK1 clock.
<table>
<tr><th>TAUBnPRS1[3:0]</th><th>CK1 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK1 are stopped (TAUBnTE.TAUBnTEm = 0).</td></tr>
</table>

Table 31.27 TAUBnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUBnPRS0[3:0]</td><td>Specifies the CK0 clock.
<table>
<tr><th>TAUBnPRS0[3:0]</th><th>CK0 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK0 are stopped (TAUBnTE.TAUBnTEm = 0).</td></tr>
</table>

NOTE

The TAUBn clock input PCLK is specified in the first part of this section, **Section 31.1.3, Clock Supply**.

31.3.3 Details of TAUBn Control Registers

31.3.3.1 TAUBnCDRm — TAUBn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUBnCMORm.TAUBnMD[4:1].

Access:	This register can be read or written in 16-bit units. • When this register functions as a capture register, only reading is possible. Write operation is ignored. • When this register functions as a compare register, reading and writing is possible.
Address:	<TAUBn_base> + m × 4_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.28 TAUBnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCDR[15:0]	Data register for the capture/compare value.

31.3.3.2 TAUBnCNTm — TAUBn Channel Counter Register

This register is the channel m counter register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + 80$_H$ + m × 4$_H$

Value after reset: FFFF$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.29 TAUBnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCNT[15:0]	16-bit counter value.

The read value depends on the counter, the operation mode change, and the values of the TAUBnTS.TAUBnTSm and TAUBnTT.TAUBnTTm bits.

The initial counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUBnTT.TAUBnTTm = 1)

The following table lists the initial counter read values after the counter has stopped (TAUBnTE.TAUBnTEm = 0) and re-enabled (TAUBnTS.TAUBnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUBnTS.TAUBnTSm = 1) for modes where the counter waits for a start trigger.

Table 31.30 TAUBnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUBnCNTm		
		Start Value^{*1}	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF$_H$	Stop value	—
Judge mode	Count down	FFFF$_H$	Stop value	—
Capture mode	Count up	0000$_H$	Stop value	—
Event count mode	Count down	FFFF$_H$	Stop value	—
One-count mode	Count down	FFFF$_H$	Stop value	FFFF$_H$
Capture and one-count mode	Count up	0000$_H$	Stop value	Capture value + 1 (TAUBnCDRm)
Judge and one-count mode	Count down	FFFF$_H$	Stop value	TAUBnCNTm value – 1
Count-up/-down mode	Count up/down	FFFF$_H$	Stop value	—
Pulse one-count mode	Count down	FFFF$_H$	Stop value	0000$_H$
Count capture mode	Count up	0000$_H$	Stop value	—
Gate count mode	Count down	FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000$_H$	Stop value	Stop value

Note 1. The value set for TAUBnCNTm when operation mode is changed after reset release

31.3.3.3 TAUBnCMORm — TAUBn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.31 TAUBnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>TAUBnCKS[1:0]</td><td>Selects the operation clock.
The operation clock is used for the TAUBTTINm input edge detection circuit.
TAUBnCNTm can also be used as the count clock depending on the setting of the TAUBnCMORm.TAUBnCCS0 bit
<table>
<tr><th>TAUBn CKS1</th><th>TAUBn CKS0</th><th>Selection of Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>CK0</td></tr>
<tr><td>0</td><td>1</td><td>CK1</td></tr>
<tr><td>1</td><td>0</td><td>CK2</td></tr>
<tr><td>1</td><td>1</td><td>CK3</td></tr>
</table></td></tr>
<tr><td>13</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>12</td><td>TAUBnCCS0</td><td>Selects the count clock for the TAUBnCNTm counter
<table>
<tr><th>TAUBnCCS0</th><th>Selected Count Clock</th></tr>
<tr><td>0</td><td>Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0].</td></tr>
<tr><td>1</td><td>Valid edge of TAUBTTINm input signal</td></tr>
</table></td></tr>
<tr><td>11</td><td>TAUBnMAS</td><td>Specifies the channel as master or slave channel during synchronous channel operation:
0: Slave
1: Master
This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.</td></tr>
</table>

Table 31.31 TAUBnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUBnSTS[2:0]</td><td>Selects the external start trigger:
<table>
<tr><th>TAUBn STS2</th><th>TAUBn STS1</th><th>TAUBn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUBTTINm input signal.
TAUBnCMURm.TAUBnTIS[1:0] specifies the valid edge.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of the TAUBTTINm input signal is used as the start trigger and the reverse edge is used as the stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUBnIm is the start trigger of the master channel</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Dead-time output signal of the TAUBTTOUTm generation unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of the master channel.</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUBnCOS[1:0]</td><td>Specifies when the capture register TAUBnCDRm and the overflow flag TAUBnCSRm.TAUBnOVF of channel m are updated.
These bits are only valid if channel m is in capture function.
The bits must be fixed to 01_B in Capture mode and Capture and gate count mode.
<table>
<tr><th>TAUBn COS1</th><th>TAUBn COS0</th><th>TAUBnCDRm</th><th>TAUBnCSRm.TAUBnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge.</td><td>Updated (cleared or set) upon detection of a TAUBTTINm input valid edge:
● If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.
● If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:
● TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm
● Overflow: $FFFF_H$ is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</td><td>Not set.</td></tr>
<tr><td>1</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 31.31 TAUBnCMORm Register Contents

Bit Position	Bit Name	Function
4 to 0	TAUBnMD[4:0]	Specifies the operation mode. For details, refer to the settings for individual functions.

TAUBn MD4	TAUBn MD3	TAUBn MD2	TAUBn MD1	TAUBn MD0	Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Judge mode
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Event count mode
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Judge and one-count mode
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Count-up/-down mode
1	0	1	0	1/0	Pulse one-count mode
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUBnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUBnIm is output at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUBnIm is not output. 1: INTTAUBnIm is output.
Event count mode Count-up/-down mode	This bit should be set to 0 (INTTAUBnIm signal is not output at the beginning of count operation).
One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION:** ● In one-count mode, INTTAUBnIm signal is not output at the beginning of count operation. ● In pulse one-count mode, INTTAUBnIm signal is output at the beginning of count operation.
Gate count mode	This bit should be set to 0 (start trigger detection during counting is disabled).
Capture and one-count mode Capture and gate count mode	This bit should be set to 0. **CAUTION:** INTTAUBnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.
Judge mode Judge and one-count mode	Specifies INTTAUBnIm output timing. 0: When TAUBnCNTm ≤ TAUBnCDRm 1: When TAUBnCNTm > TAUBnCDRm

31.3.3.4 TAUBnCMURm — TAUBn Channel Mode User Register

This register specifies the type of valid edge detection used for the TAUBTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUBn_base> + $C0_H$ + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.32 TAUBnCMURm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 2</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>1, 0</td><td>TAUBnTIS[1:0]</td><td>Specifies the valid edge of the TAUBTTINm input signal:
<table>
<tr><th>TAUBn TIS1</th><th>TAUBn TIS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Rising and falling edges (low-width measurement selection).
Start trigger: falling edge
Stop trigger (capture): rising edge</td></tr>
<tr><td>1</td><td>1</td><td>Rising and falling edges (high-width measurement selection).
Start trigger: rising edge
Stop trigger (capture): falling edge</td></tr>
</table>
● Edge detection for TAUBTTINm input signals is performed based on the operation clock selected by TAUBnCMORm.TAUBnCKS[1:0].</td></tr>
</table>

31.3.3.5 TAUBnCSRm — TAUBn Channel Status Register

This register indicates the count direction and the overflow status of the counter for channel m.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUBn_base> + 140_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnCSF	TAUBnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.33 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUBnCSF	Indicates the count direction: 0: Counts up 1: Counts down The read value of this bit is only valid in the following mode: ● Up/Down Count mode
0	TAUBnOVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is used only in the following modes: ● Capture mode ● Capture and one-count mode The function of this bit depends on the setting of control bits TAUBnCMORm.TAUBnCOS[1:0].

31.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUBnCSRm.TAUBnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H

Address: <TAUBn_base> + 180_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 31.34 TAUBnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUBnCLOV	0: No function 1: Clears the overflow flag TAUBnCSRm.TAUBnOVF

31.3.3.7 TAUBnTS — TAUBn Channel Start Trigger Register

This register enables the counter for each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000$_H$

Address: <TAUBn_base> + 1C4$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TS15	TAUBn TS14	TAUBn TS13	TAUBn TS12	TAUBn TS11	TAUBn TS10	TAUBn TS09	TAUBn TS08	TAUBn TS07	TAUBn TS06	TAUBn TS05	TAUBn TS04	TAUBn TS03	TAUBn TS02	TAUBn TS01	TAUBn TS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 31.35 TAUBnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTSm	Enables the counter for channel m: 0: No function 1: Enables the counter and sets TAUBnTE.TAUBnTEm = 1. TAUBnTE.TAUBnTEm = 1 only enables the counter. Whether the counter starts depends on the selected operation mode.

31.3.3.8 TAUBnTE — TAUBn Channel Enable Status Register

This register indicates whether counter is enabled or disabled.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + 1C0$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TE15	TAUBn TE14	TAUBn TE13	TAUBn TE12	TAUBn TE11	TAUBn TE10	TAUBn TE09	TAUBn TE08	TAUBn TE07	TAUBn TE06	TAUBn TE05	TAUBn TE04	TAUBn TE03	TAUBn TE02	TAUBn TE01	TAUBn TE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.36 TAUBnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTEm	Indicates whether counter for channel m is enabled or disabled: 0: Counter disabled 1: Counter enabled Setting TAUBnTS.TAUBnTSm to 1 sets this bit to 1. Setting TAUBnTT.TAUBnTTm to 1 resets this bit to 0.

31.3.3.9 TAUBnTT — TAUBn Channel Stop Trigger Register

This register stops the counter for each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000$_H$.

Address: <TAUBn_base> + 1C8$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TT15	TAUBn TT14	TAUBn TT13	TAUBn TT12	TAUBn TT11	TAUBn TT10	TAUBn TT09	TAUBn TT08	TAUBn TT07	TAUBn TT06	TAUBn TT05	TAUBn TT04	TAUBn TT03	TAUBn TT02	TAUBn TT01	TAUBn TT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 31.37 TAUBnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTTm	Stops the counter of channel m: 0: No function 1: Stops the counter and resets TAUBnTE.TAUBnTEm. TAUBnCNTm, TAUBnTO.TAUBnTOm, and TAUBTTOUTm all retain the values they had before the counter was stopped.

31.3.4 Details of TAUBn Simultaneous Rewrite Registers

31.3.4.1 TAUBnRDE — TAUBn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUBnCDRm/TAUBnTOLm.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 260$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDE15	TAUBn RDE14	TAUBn RDE13	TAUBn RDE12	TAUBn RDE11	TAUBn RDE10	TAUBn RDE09	TAUBn RDE08	TAUBn RDE07	TAUBn RDE06	TAUBn RDE05	TAUBn RDE04	TAUBn RDE03	TAUBn RDE02	TAUBn RDE01	TAUBn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.38 TAUBnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

31.3.4.2 TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register

This register selects the control channel for simultaneous rewrite.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 268$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDS15	TAUBn RDS14	TAUBn RDS13	TAUBn RDS12	TAUBn RDS11	TAUBn RDS10	TAUBn RDS09	TAUBn RDS08	TAUBn RDS07	TAUBn RDS06	TAUBn RDS05	TAUBn RDS04	TAUBn RDS03	TAUBn RDS02	TAUBn RDS01	TAUBn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.39 TAUBnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDSm	Specifies which channel is controlled for the simultaneous rewrite trigger: 0: Master channel 1: Another upper channel

31.3.4.3 TAUBnRDM — TAUBn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 264_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDM15	TAUBn RDM14	TAUBn RDM13	TAUBn RDM12	TAUBn RDM11	TAUBn RDM10	TAUBn RDM09	TAUBn RDM08	TAUBn RDM07	TAUBn RDM06	TAUBn RDM05	TAUBn RDM04	TAUBn RDM03	TAUBn RDM02	TAUBn RDM01	TAUBn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.40 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDMm	Selects when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: At the top of a triangle wave cycle These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 0.

31.3.4.4 TAUBnRDC — TAUBn Channel Reload Data Control Register

This register specifies the channel that generates the INTTAUBnIm signal that triggers simultaneous rewrite.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0

Address: <TAUBn_base> + $26C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDC15	TAUBn RDC14	TAUBn RDC13	TAUBn RDC12	TAUBn RDC11	TAUBn RDC10	TAUBn RDC09	TAUBn RDC08	TAUBn RDC07	TAUBn RDC06	TAUBn RDC05	TAUBn RDC04	TAUBn RDC03	TAUBn RDC02	TAUBn RDC01	TAUBn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.41 TAUBnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 1.

31.3.4.5 TAUBnRDT — TAUBn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUBn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDT15	TAUBn RDT14	TAUBn RDT13	TAUBn RDT12	TAUBn RDT11	TAUBn RDT10	TAUBn RDT09	TAUBn RDT08	TAUBn RDT07	TAUBn RDT06	TAUBn RDT05	TAUBn RDT04	TAUBn RDT03	TAUBn RDT02	TAUBn RDT01	TAUBn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 31.42 TAUBnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDTm	Triggers the simultaneous rewrite enabling state: 0: No function. Writing 0 is ignored (the operation is not affected). 1: The simultaneous rewrite enabling flag (TAUBnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: ● TAUBnRDE.TAUBnRDEm = 1

31.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RSF15	TAUBn RSF14	TAUBn RSF13	TAUBn RSF12	TAUBn RSF11	TAUBn RSF10	TAUBn RSF09	TAUBn RSF08	TAUBn RSF07	TAUBn RSF06	TAUBn RSF05	TAUBn RSF04	TAUBn RSF03	TAUBn RSF02	TAUBn RSF01	TAUBn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.43 TAUBnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRSFm	Indicates the simultaneous rewrite status: 0: Indicates simultaneous rewrite is completed due to the generation of the simultaneous rewrite trigger. 1: Indicates the simultaneous rewrite trigger waiting state when simultaneous rewrite is enabled (TAUBnRDTm = 1).

31.3.5 Details of TAUBn Output Registers

31.3.5.1 TAUBnTOE — TAUBn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + $5C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOE15	TAUBn TOE14	TAUBn TOE13	TAUBn TOE12	TAUBn TOE11	TAUBn TOE10	TAUBn TOE09	TAUBn TOE08	TAUBn TOE07	TAUBn TOE06	TAUBn TOE05	TAUBn TOE04	TAUBn TOE03	TAUBn TOE02	TAUBn TOE01	TAUBn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.44 TAUBnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOEm	Enables/disables independent channel output mode: 0: Disables independent channel output mode (controlled by software) 1: Enables independent channel output mode

31.3.5.2 TAUBnTO — TAUBn Channel Output Register

This register specifies and reads the level of TAUBTTOUTm.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TO15	TAUBn TO14	TAUBn TO13	TAUBn TO12	TAUBn TO11	TAUBn TO10	TAUBn TO09	TAUBn TO08	TAUBn TO07	TAUBn TO06	TAUBn TO05	TAUBn TO04	TAUBn TO03	TAUBn TO02	TAUBn TO01	TAUBn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.45 TAUBnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOm	Specifies/reads the level of TAUBTTOUTm: 0: Low 1: High Only TAUBnTOm bits for which Independent Channel Output function is disabled (TAUBnTOEm = 0) can be written.

31.3.5.3 TAUBnTOM — TAUBn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 248_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOM15	TAUBn TOM14	TAUBn TOM13	TAUBn TOM12	TAUBn TOM11	TAUBn TOM10	TAUBn TOM09	TAUBn TOM08	TAUBn TOM07	TAUBn TOM06	TAUBn TOM05	TAUBn TOM04	TAUBn TOM03	TAUBn TOM02	TAUBn TOM01	TAUBn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.46 TAUBnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode

31.3.5.4 TAUBnTOC — TAUBn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUBnTOMm.

Access:	This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).
Address:	<TAUBn_base> + $24C_H$
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOC15	TAUBn TOC14	TAUBn TOC13	TAUBn TOC12	TAUBn TOC11	TAUBn TOC10	TAUBn TOC09	TAUBn TOC08	TAUBn TOC07	TAUBn TOC06	TAUBn TOC05	TAUBn TOC04	TAUBn TOC03	TAUBn TOC02	TAUBn TOC01	TAUBn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.47 TAUBnTOC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>TAUBnTOCm</td><td>Specifies the output mode:
0: Operation mode 1
1: Operation mode 2
The output mode also depends on TAUBnTOM.TAUBnTOMm, as can be seen in the following table.
<table>
<tr><th>TOMm</th><th>TOCm</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Toggle mode: Toggles when INTTAUBnIm occurs.</td></tr>
<tr><td></td><td>1</td><td>Set/reset mode: Set when INTTAUBnIm occurs upon count start and reset when INTTAUBnIm occurs due to detection of a match between TAUBnCNTm and TAUBnCDRm.</td></tr>
<tr><td>1</td><td>0</td><td>Synchronous channel operation mode 1:
Set when INT occurs on the master channel and reset when INT occurs on the slave channel.</td></tr>
<tr><td></td><td>1</td><td>Synchronous channel operation mode 2:
Set when INTTAUBnIm occurs while the slave channel is counting down and reset when INTTAUBnIm occurs while the slave channel is counting up.</td></tr>
</table>
</td></tr>
</table>

31.3.5.5 TAUBnTOL — TAUBn Channel Output Active Level Register

This register specifies the output logic of the channel output bit (TAUBnTO.TAUBnTOm).

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOL15	TAUBn TOL14	TAUBn TOL13	TAUBn TOL12	TAUBn TOL11	TAUBn TOL10	TAUBn TOL09	TAUBn TOL08	TAUBn TOL07	TAUBn TOL06	TAUBn TOL05	TAUBn TOL04	TAUBn TOL03	TAUBn TOL02	TAUBn TOL01	TAUBn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.48 TAUBnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOLm	Specifies the output logic of the channel m output bit (TAUBnTO.TAUBnTOm): 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1.

31.3.6 Details of TAUBn Dead Time Output Registers

31.3.6.1 TAUBnTDE — TAUBn Channel Dead Time Output Enable Register

This register enables/disables dead time operation for each channel.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDE15	TAUBn TDE14	TAUBn TDE13	TAUBn TDE12	TAUBn TDE11	TAUBn TDE10	TAUBn TDE09	TAUBn TDE08	TAUBn TDE07	TAUBn TDE06	TAUBn TDE05	TAUBn TDE04	TAUBn TDE03	TAUBn TDE02	TAUBn TDE01	TAUBn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.49 TAUBnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDEm	Enables/disables dead time control operation of channel m: 0: Disables dead time operation 1: Enables dead time operation The same settings must be set for the even and the odd slave channel that comprise a set. ● These bits only apply when: TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm = 1

31.3.6.2 TAUBnTDL — TAUBn Channel Dead Time Output Level Register

This register selects the phase period to which dead time is added.

Access:	This register can be read or written in 16-bit units.
Address:	<TAUBn_base> + 54_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDL15	TAUBn TDL14	TAUBn TDL13	TAUBn TDL12	TAUBn TDL11	TAUBn TDL10	TAUBn TDL09	TAUBn TDL08	TAUBn TDL07	TAUBn TDL06	TAUBn TDL05	TAUBn TDL04	TAUBn TDL03	TAUBn TDL02	TAUBn TDL01	TAUBn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.50 TAUBnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDLm	Selects the phase period to which dead time is added: 0: Positive phase period 1: Negative phase period These bits only apply when: ● TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm, TAUBnTDE.TAUBnTDEm = 1

31.3.7 TAUBn Emulation Register

31.3.7.1 TAUBnEMU — TAUBn Emulation Register

This register controls SVSTOP operations.

Access: This register can be read or written in 8-bit units.

Write to this register only when the counter is stopped (TAUBnTE.TAUBnTEm = 0) and when EPC.SVSTOP = 0.

Address: <TAUBn_base> + 290_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUBnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 31.51 TAUBnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUBnSVSDIS	(When EPC.SVSTOP = 0) Regardless of the value of this bit (1/0), the count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). (When EPC.SVSTOP = 1) 0: The count clock stops when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). 1: The count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

31.4 General Operating Procedure

The following describes the general operation procedure for the TAUBn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUBTTOUTm is also initialized and outputs a low level.

1. Set the TAUBnTPS register to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUBn function:
   - Set the operation mode
   - Set the channel output mode
   - Set any other control bits
3. Enable the counter by setting the TAUBnTS.TAUBnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUBnTT.TAUBnTTm bit to 1. The counter can be forcibly restarted by setting the TAUBnTS.TAUBnTSm bit to 1.
5. Stop the function by setting the TAUBnTT.TAUBnTTm bit to 1.

NOTE

- A detailed description of the required control bits and the operation of the individual functions is given below.
  - **Section 31.12, Independent Channel Operation Functions**
  - **Section 31.14, Synchronous Channel Operation Functions**
- The function can be changed while the counter is stopped (TAUBnTE.TAUBnTEm=0).

31.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 31.5.1, Rules of Synchronous Channel Operation Function**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 31.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 31.6, Simultaneous Rewrite**

31.5.1 Rules of Synchronous Channel Operation Function

Number of master and slave channels

- Only even channels (CH0, CH2, CH4, …) can be set as master channels.
  Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, a range of slave channels that includes another master channel cannot be set for a given master channel.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the master channel and the synchronized slave channel. This is achieved by setting the same value to the TAUBnCMORm.TAUBnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave channel usage and operation clocks are illustrated in **Figure 31.4, Grouping of the Channels and Assignment of Operation Clocks**.

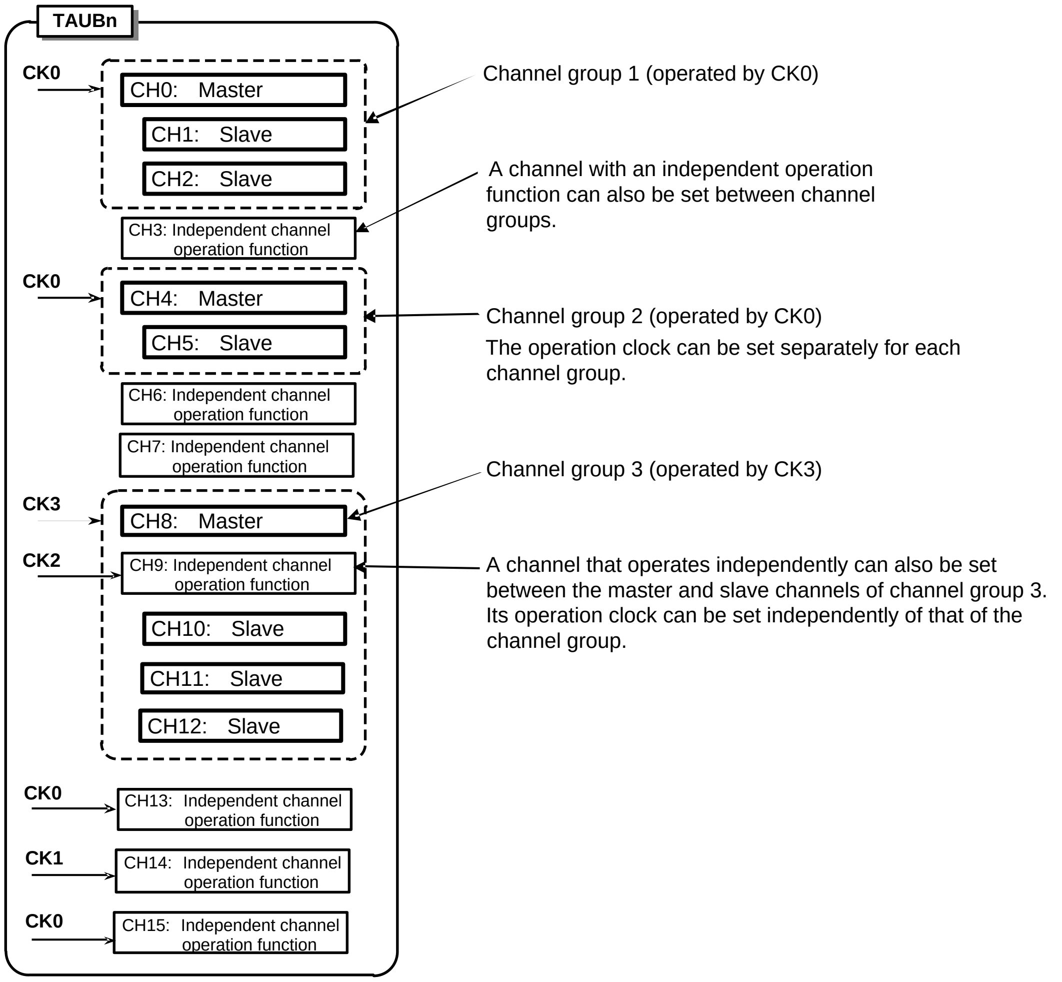


Figure 31.4 Grouping of the Channels and Assignment of Operation Clocks

31.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit.

31.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUBnTS.TAUBnTSm bits of the channels must be set at the same time.
- To simultaneously stop synchronized channels, the TAUBnTT.TAUBnTTm bits of the channels must be set at the same time.

Setting the TAUBnTS.TAUBnTSm bits to 1 sets the corresponding TAUBnTE.TAUBnTEm bits to 1, enabling counting. The exact time that it starts depends on the operation mode.

31.6 Simultaneous Rewrite

31.6.1 Introduction

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUBnCDRm and TAUBnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUBnIm being issued on the upper channel specified by TAUBnRDC.TAUBnRDCm

There are three methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 31.52 Simultaneous Rewrite Methods and when They are Triggered

Method	Trigger	TAUBnRDE. TAUBnRDEm	TAUBnRDS. TAUBnRDSm	TAUBnRDM. TAUBnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular wave of the corresponding slave channel.	1	0	1
C1	INTTAUBnIm is generated on an upper channel specified by TAUBnRDC.TAUBnRDCm	1	1	0/1

The following table lists which of these three methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 31.13, Independent Channel Simultaneous Rewrite Functions** and **Section 31.14, Synchronous Channel Operation Functions**.

Table 31.53 Channel Functions and the Methods They Use for Simultaneous Rewrite

Descriptions	A	B	C1	TAUBnTOL. TAUBnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			✓	
PWM Output Function	✓		✓	✓
One-Shot Pulse Output Function	✓			
Delay Pulse Output Function	✓			
Triangle PWM Output Function		✓	✓	✓
Triangle PWM Output Function with Dead Time		✓	✓	
A/D Conversion Trigger Output Function Type 1	✓		✓	
A/D Conversion Trigger Output Function Type 2		✓	✓	

Note: ✓: Available, (Blank): Unavailable

31.6.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite.

The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

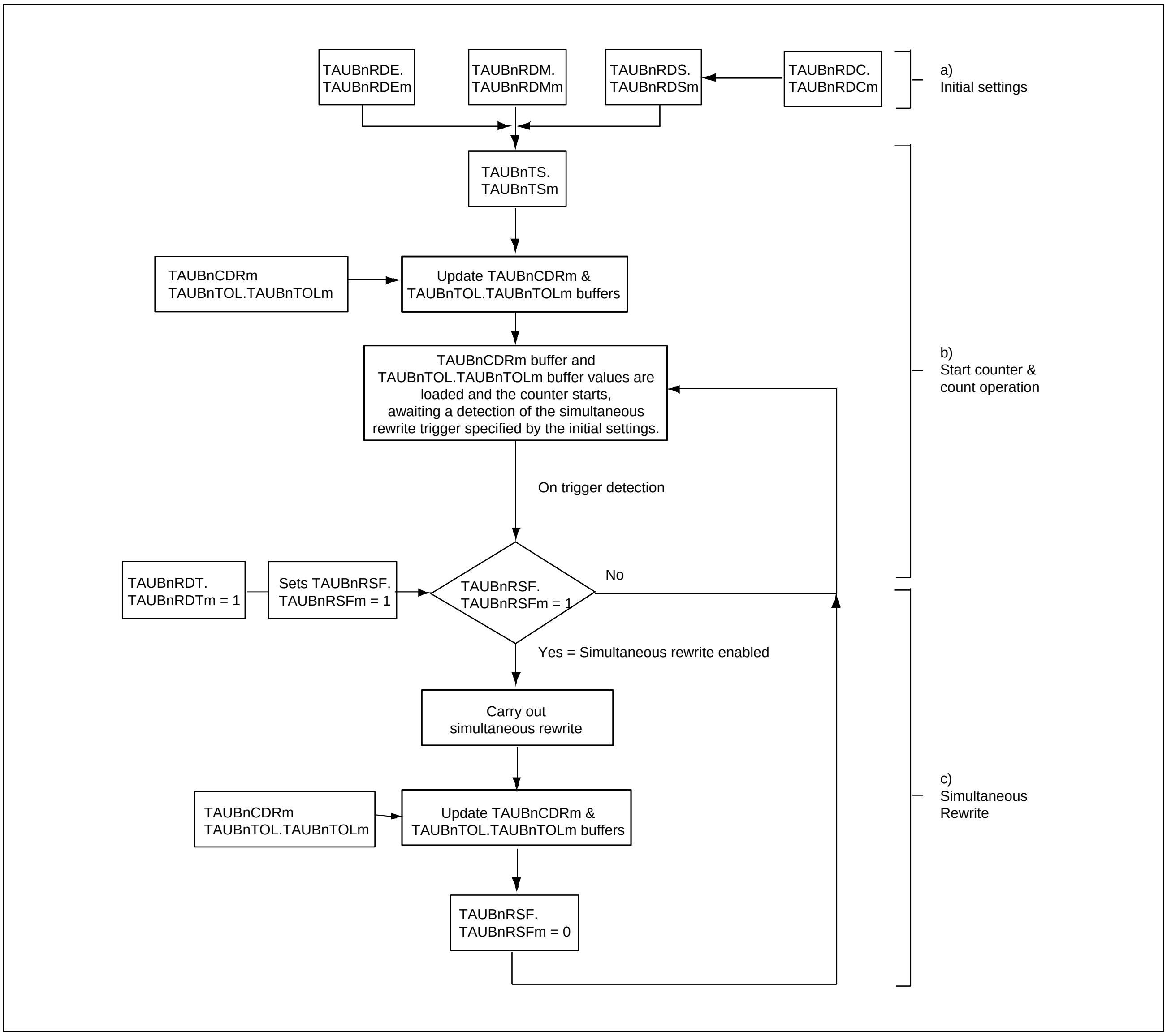


Figure 31.5 General Procedure for Simultaneous Rewrite

31.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUBnRDE.TAUBnRDEm = 1.
- To select the type of simultaneous rewrite, set TAUBnRDM.TAUBnRDMm and TAUBnRDS.TAUBnRDSm according to the values in **Table 31.52, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using TAUBnRDC.TAUBnRDCm. (Prerequisite: TAUBnRDS.TAUBnRDSm has been set to the upper channel.)

31.6.2.2 Start Counter and Count Operation

- To start all the TAUBnCNTm counters in the channel group, set the corresponding TAUBnTS.TAUBnTSm bits to 1. TAUBnTOL.TAUBnTOLm and the values in the data registers (TAUBnCDRm) are written to the corresponding TAUBnTOL.TAUBnTOLm buffer (TAUBnTOL.TAUBnTOLm buf) and data buffer registers (TAUBnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1 sets the reload flag (TAUBnRSF.TAUBnRSFm) to 1, enabling simultaneous rewrite. TAUBnRSF.TAUBnRSFm remains at 1 until simultaneous rewrite has taken place.
- When the specified trigger for simultaneous rewrite is detected, the TAUBnRSF.TAUBnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise, simultaneous rewrite is not carried out, and the system awaits the next simultaneous rewrite trigger detection.

31.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then written to the corresponding counters and the values are applied the next time the counter starts or restarts.
- When simultaneous rewrite is finished, the TAUBnRSF.TAUBnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

31.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUBnRDE.TAUBnRDEm, TAUBnRDS.TAUBnRDSm, TAUBnRDM.TAUBnRDMm, and TAUBnRDC.TAUBnRDCm cannot be changed while the counter is in operation (TAUBnTE.TAUBnTEm = 1).
- TAUBnTOL.TAUBnTOLm can only be rewritten during operation in PWM output function or triangle PWM output function. For all other output functions, TAUBnTOL.TAUBnTOLm must be written before the counter starts. If it is rewritten in another function mode, TAUBTTOUTm outputs an invalid wave.
- When a simultaneous rewrite trigger is issued on an upper channel (TAUBnRDS.TAUBnRDSm = 1), the TAUBnRDC.TAUBnRDCm bit controls all the lower channels. This means that if the TAUBnRDC.TAUBnRDCm bits of CH2 and CH7 are set to 1 and the TAUBnRDC.TAUBnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger generation channel (TAUBnRDE.TAUBnRDEm and TAUBnRDS.TAUBnRDSm = 1) but no upper channel is set (TAUBnRDC.TAUBnRDC[15:0] = 0), simultaneous rewrite cannot take place.

31.6.4 Types of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

31.6.4.1 Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A)

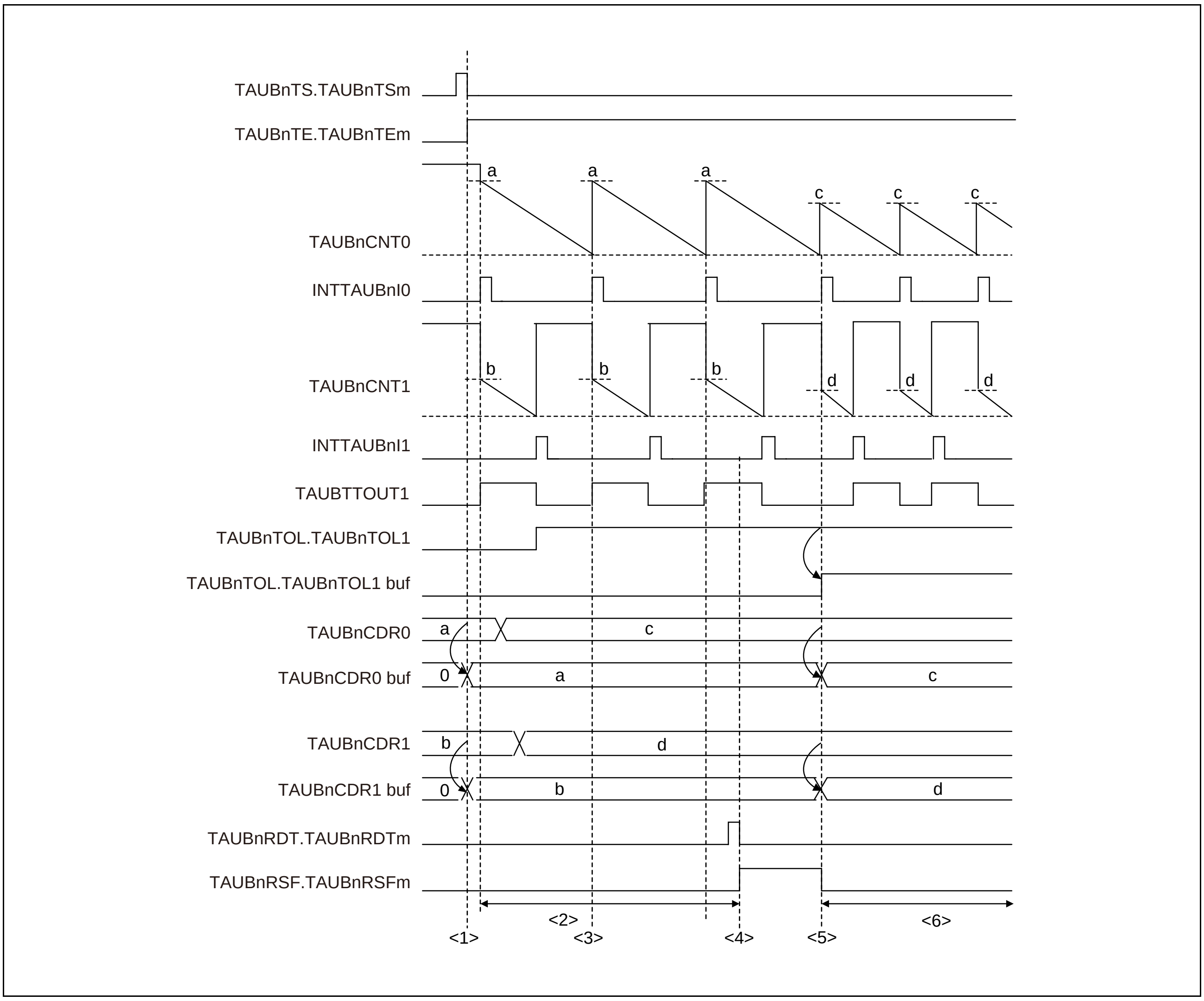


Figure 31.6 Simultaneous Rewrite when the Master Channel (Re)Starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer and the value of TAUBnTOL.TAUBnTOLm is copied to the TAUBnTOL.TAUBnTOLm buffer.

(2) The TAUBnCDRm and TAUBnTOL.TAUBnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0)

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

31.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B)

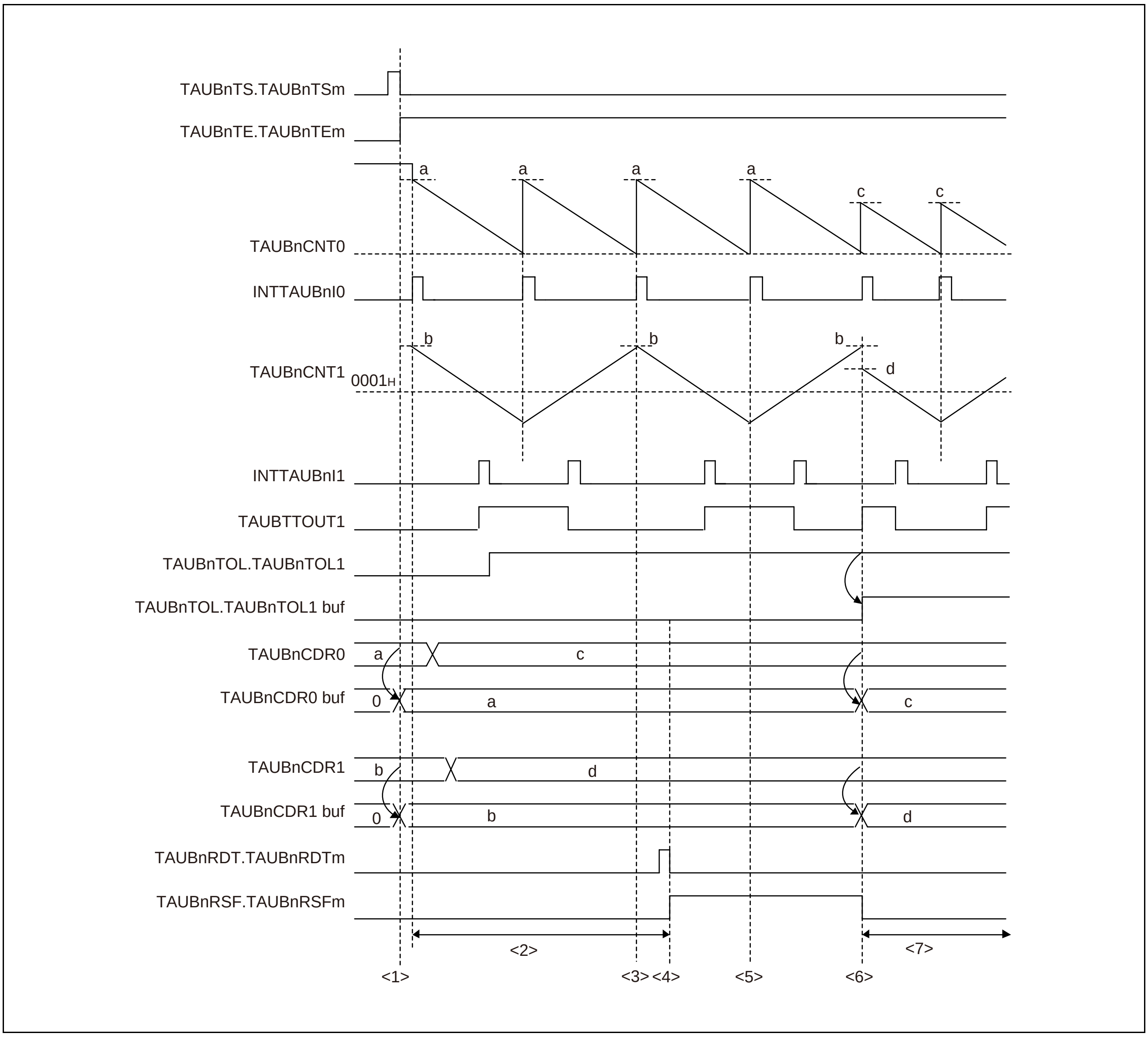


Figure 31.7 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm and TAUBnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0).

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the start timing of the top of the triangular cycle. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

31.6.4.3 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1)

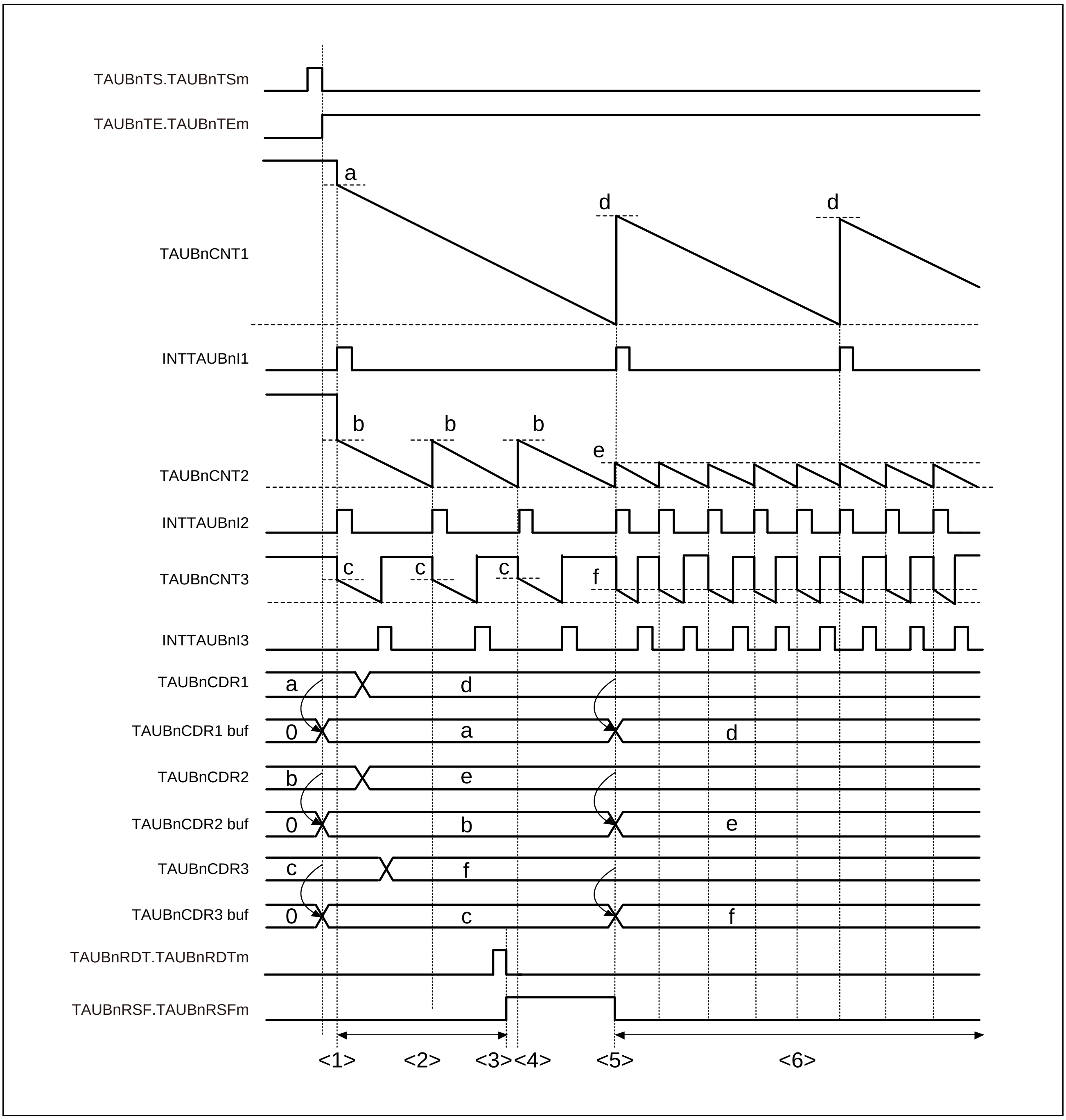


Figure 31.8 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm

Setting:

CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUBnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

(1) When TAUBnTS.TAUBnTSm is set to 1, the TAUBnCDRm value is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1, the status flag is set (TAUBnRSF.TAUBnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUBnCDRm values are loaded into the corresponding TAUBnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUBnCDRm register can be rechanged.

31.7 Channel Output Modes

The output of the TAUBTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUBnTOE.TAUBnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUBnTO.TAUBnTOm) is sent to the output pin (TAUBTTOUTm).
- By TAUB signals (TAUBnTOE.TAUBnTOEm = 1)
  When controlled by TAUB signals, the output level of TAUBTTOUTm is set or reset or toggled by internal signals. The value of TAUBnTO.TAUBnTOm is updated accordingly to reflect the value of TAUBTTOUTm.
  - Independently (TAUBnTOM.TAUBnTOMm = 0)
    In case of independent operation, the output of the TAUBTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUBnTOM.TAUBnTOMm = 0).
  - Synchronously (TAUBnTOM.TAUBnTOMm = 1)
    In case of synchronous operation, the output of the TAUBTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUBnTOM.TAUBnTOMm = 1).

The TAUBnTO.TAUBnTOm bit can always be read to determine the current value of TAUBTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 31.54, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 31.7.2, Channel Output Modes Controlled Independently by TAUBn Signals**
- **Section 31.7.3, Channel Output Modes Controlled Synchronously by TAUBn Signals**

Batch operation of TAUBnTOm bit

Whether a set value is reflected to the TAUBnTOm bit or not is controlled by the TAUBnTOE.TAUBnTOEm bit.

The TAUBnTOm setting is written only to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 0 when a write to the TAUBnTO register is attempted. No TAUBnTOm setting is reflected to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 1.

NOTE

The TAUBnTO.TAUBnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUBnTOL.TAUBnTOLm.

The value of TAUBnTOL.TAUBnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. If TAUBnTOL.TAUBnTOLm is changed after the counter starts, the output of TAUBTTOUTm is undefined.

See **Section 31.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 31.54, Channel Output Modes**.

Table 31.54 Channel Output Modes

Channel Output Mode	TAUBnTOE.TAUBnTOEm	TAUBnTOM.TAUBnTOMm	TAUBnTOC.TAUBnTOCm	TAUBnTDE.TAUBnTDEm
By software				
Independent channel output mode controlled by software	0	x		
By TAUB signals, independently				
Independent channel output mode 1	1	0	0	0
Independent channel output mode 2			1	
By TAUB signals, synchronously				
Synchronous channel output mode 1	1	1	0	0
Synchronous channel output mode 2			1	0
Synchronous channel output mode 2 with dead time output				1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUBnTE.TAUBnTEm = 1):

- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

31.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUBTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUBnTOE.TAUBnTOEm = 0).

(1) Set TAUBnTO.TAUBnTOm to specify the initial level of the TAUBTTOUTm output.

(2) Set channel output mode according to **Table 31.54, Channel Output Modes**, and the output logic using the TAUBnTOL.TAUBnTOLm bit.

(3) Start the counter (TAUBnTS.TAUBnTSm = 1).

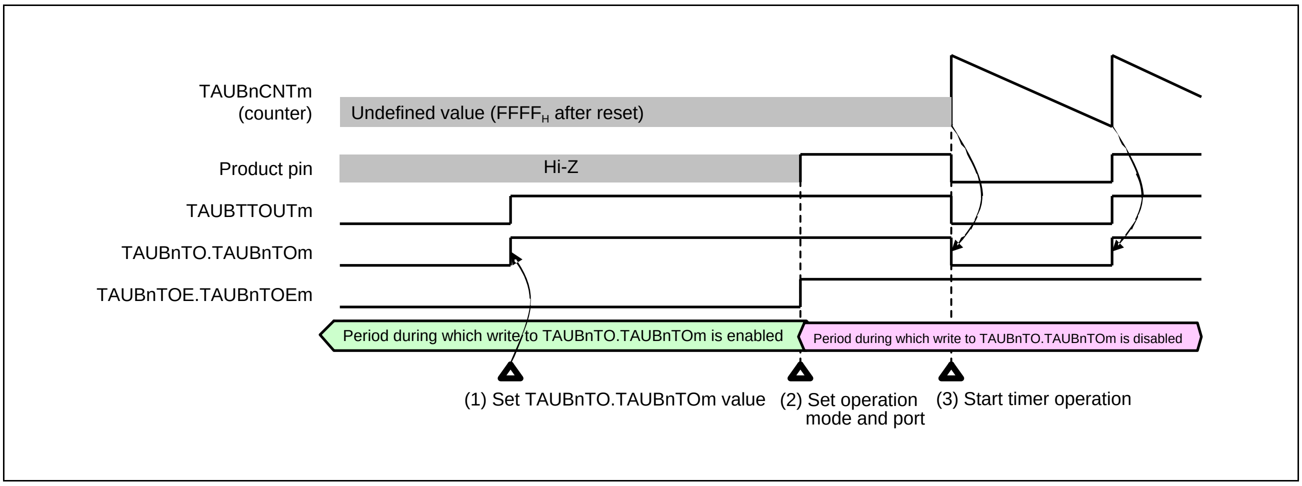


Figure 31.9 General Procedure for Specifying a TAUBTTOUTm Channel Output Mode

31.7.2 Channel Output Modes Controlled Independently by TAUBn Signals

This section lists the channel output modes that are controlled independently by TAUBn signals. The control bits used to specify a mode are listed in **Table 31.54, Channel Output Modes**.

31.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUBTTOUTm toggles when INTTAUBnIm is detected. The value of TAUBnTOL.TAUBnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 31.54, Channel Output Modes**.

31.7.2.2 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUBTTOUTm is set when INTTAUBnIm occurs at the time of count start, and reset when INTTAUBnIm occurs due to a match between TAUBnCNTm and TAUBnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 31.54, Channel Output Modes**.

31.7.3 Channel Output Modes Controlled Synchronously by TAUBn Signals

This section lists the channel output modes that are controlled synchronously by TAUBn signals. The control bits used to specify a mode are listed in **Table 31.54, Channel Output Modes**.

31.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUBnIm of master channel serves as a set signal and INTTAUBnIm of the slave channel as a reset signal. If INTTAUBnIm of the master channel and INTTAUBnIm of the slave channel are generated at the same time, INTTAUBnIm of the slave channel (reset signal) has priority over INTTAUBnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 31.54, Channel Output Modes**.

31.7.3.2 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is triangular wave PWM output at TAUBTTOUTm. For details, see **Section 31.14.5, Triangle PWM Output Function**.

Set/reset conditions

TAUBnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUBTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUBTTOUTm should be set to 0 before the function starts.

31.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUBTTOUTm. The set/reset conditions are shown in **Figure 31.10, Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output**.

Set/reset conditions

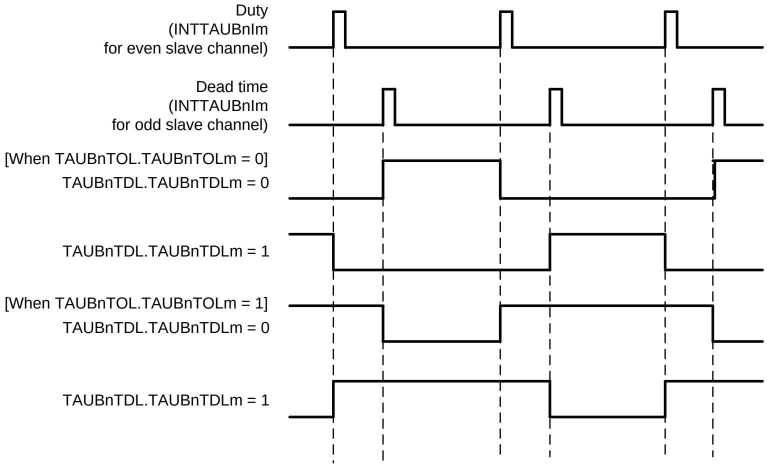


Figure 31.10 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUBnTDL.TAUBnTDLm = 0 for rising edges and TAUBnTDL.TAUBnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUBnTOE.TAUBnTOEm
- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

31.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUBnTS.TAUBnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

31.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating at the next count clock after TAUBnTS.TAUBnTSm is set to 1. The value of data register is also loaded when the counter starts.

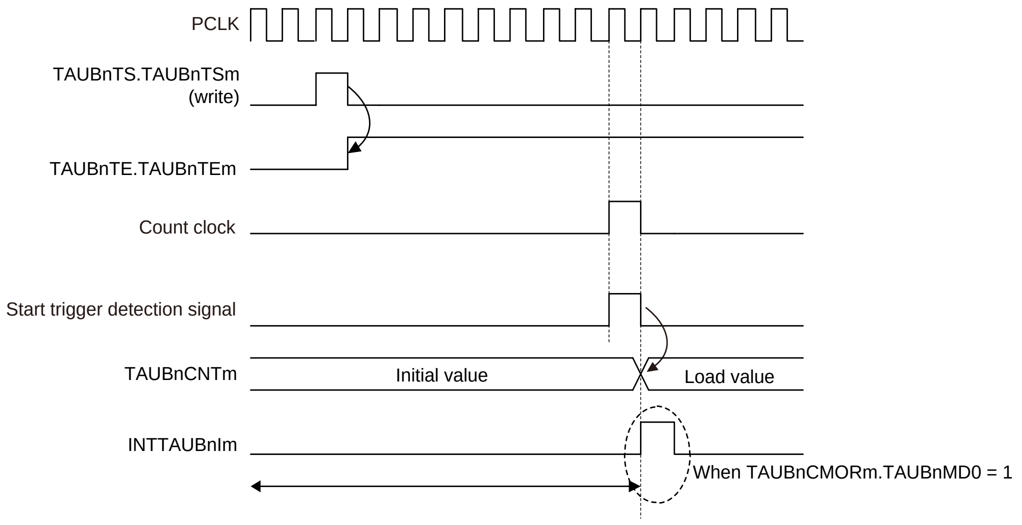


Figure 31.11 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Up/Down Count Mode, and Count Capture Mode

NOTE

Make sure to set TAUBnCMORm.TAUBnMD0 to 0 when using the count-up/-down mode.

31.8.2 Event Count Mode

The value of data register is loaded as soon as TAUBnTS.TAUBnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

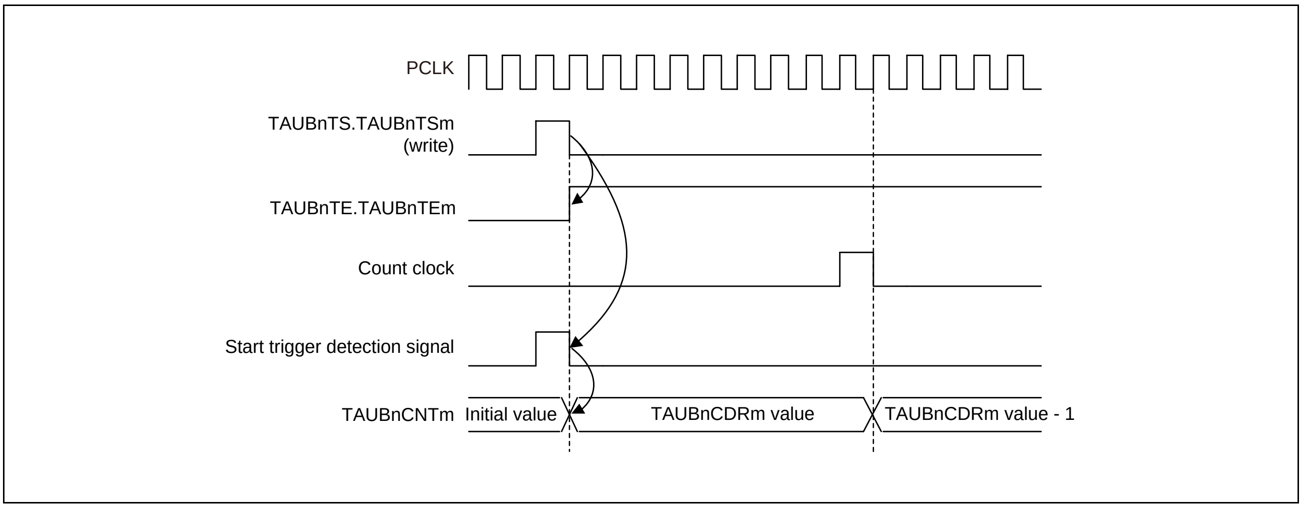


Figure 31.12 Start Timing in Event Count Mode

31.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUBTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

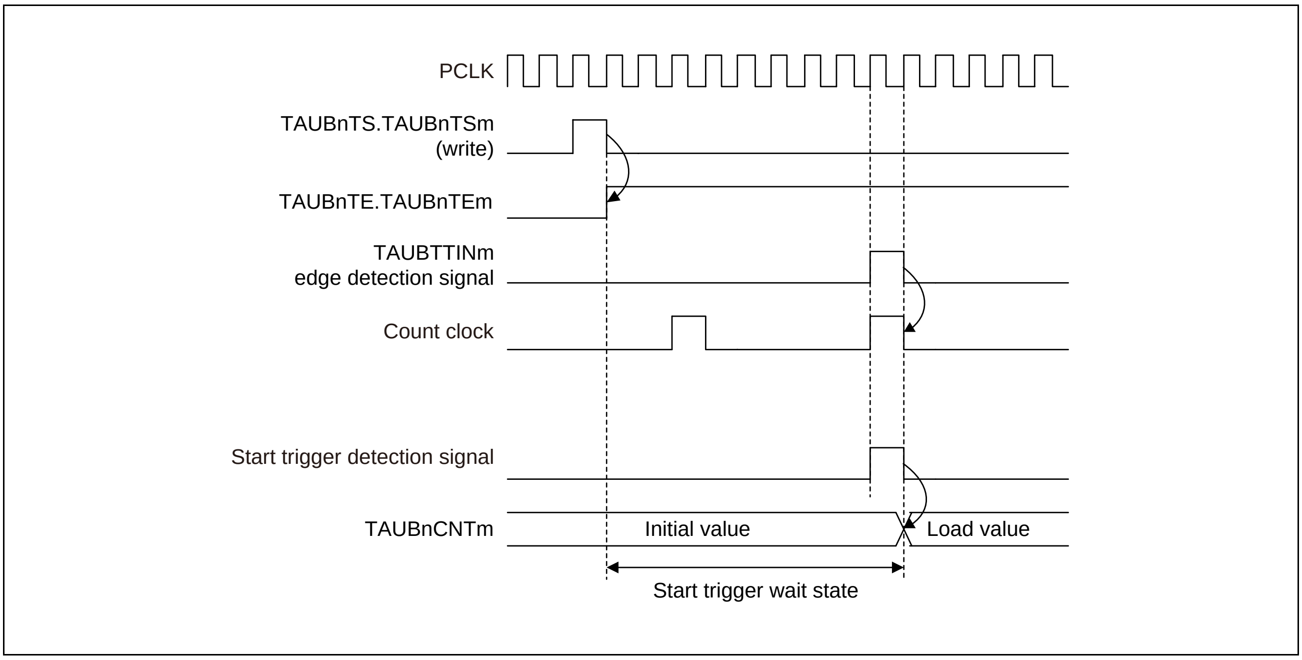
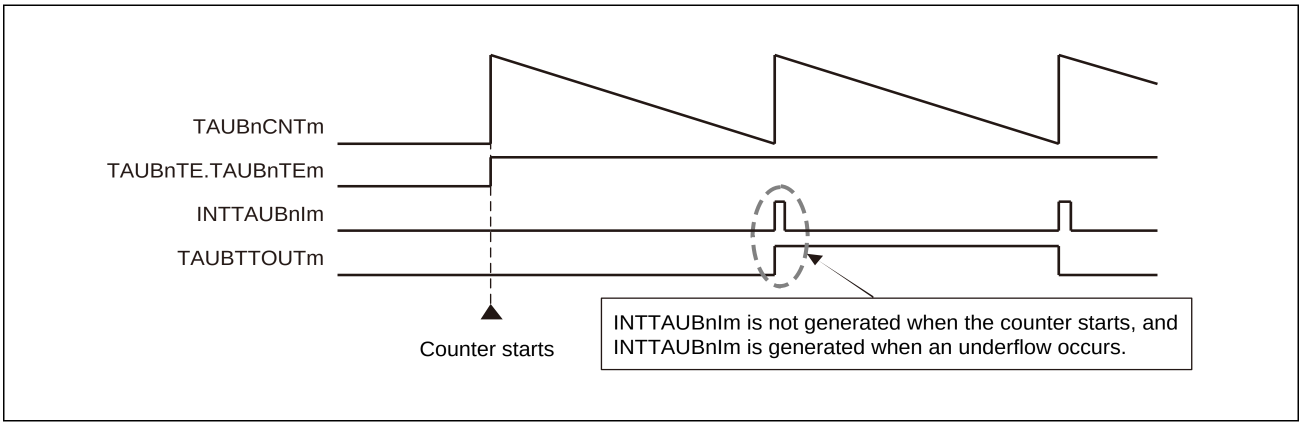


Figure 31.13 Count Start Timing in Other Operating Modes

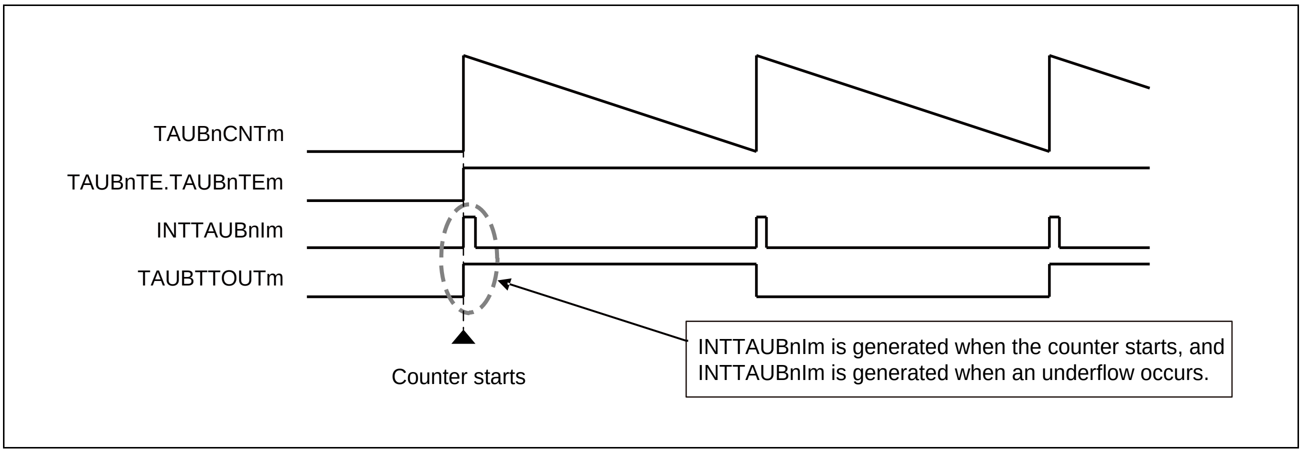
31.9 TAUBTTOUTm Output and INTTAUBnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUBnIm is generated using the TAUBnCMORm.TAUBnMD0 bit. The generation of INTTAUBnIm when the TAUBnCMORm.TAUBnMD0 bit starts counting and the effect to TAUBTTOUTm depend on the selected function. For details, refer to the description of TAUBnCMORm.TAUBnMD0 of each function.



INTTAUBnIm is not generated when the counter starts, and INTTAUBnIm is generated when an underflow occurs.

Figure 31.14 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0 = 0)



INTTAUBnIm is generated when the counter starts, and INTTAUBnIm is generated when an underflow occurs.

Figure 31.15 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0 = 1)

31.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUBnCNTm = $FFFF_H$).
- Set TAUBnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUBTTINm input.
- The trigger detection settings (TAUBnCMORm.TAUBnSTS[2:0] and TAUBnCMURm.TAUBnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUBnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

31.10.1 Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the TAUBTTINm input interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input pulse interval measurement function exceeds $FFFF_H$.

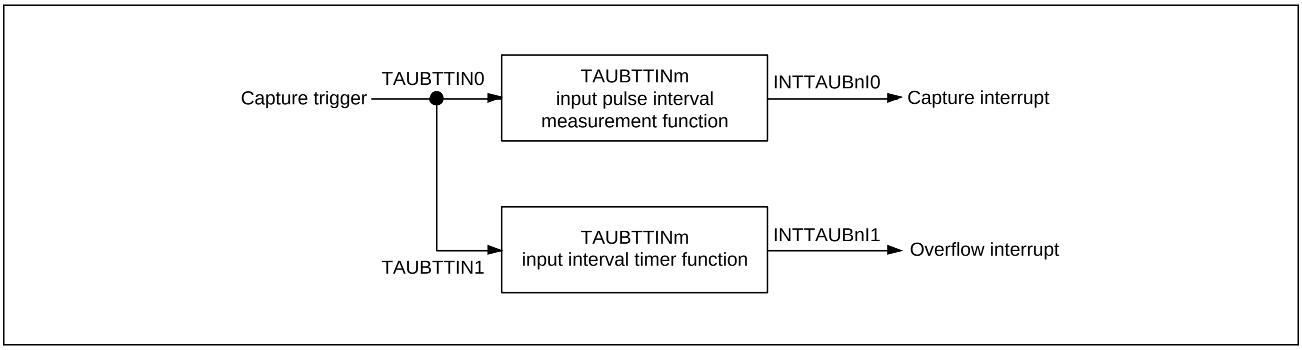


Figure 31.16 Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

Timing diagram

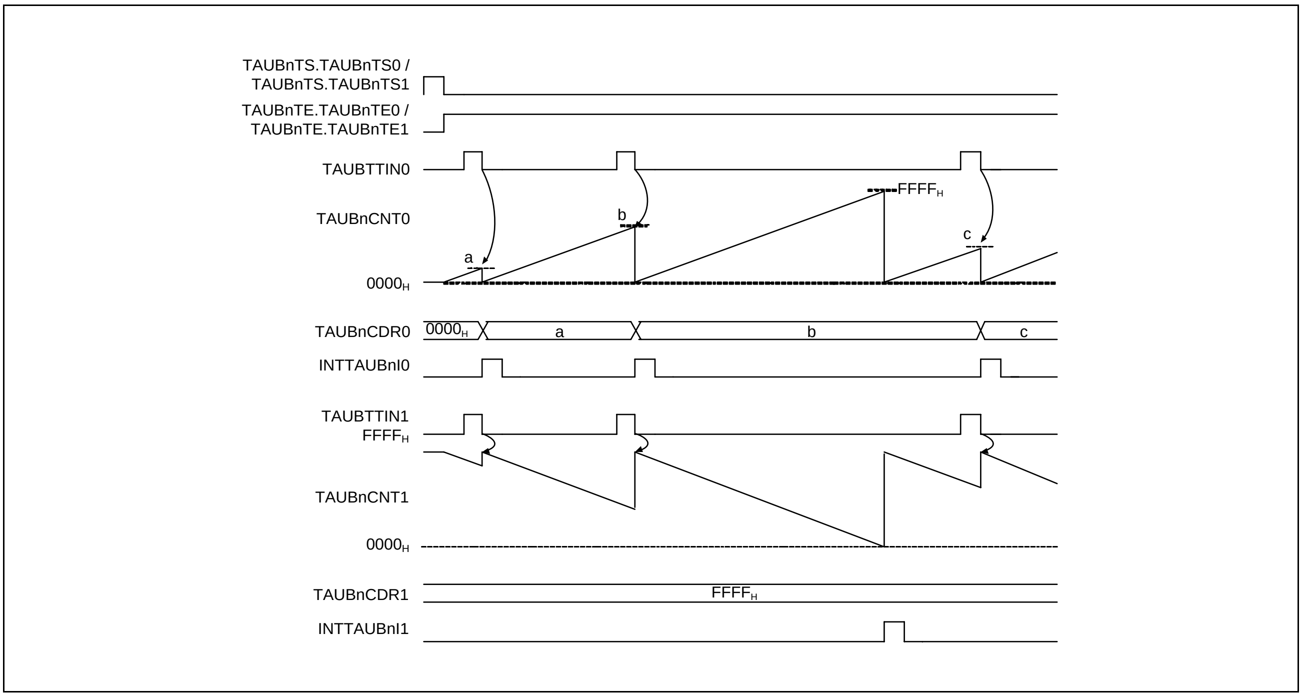


Figure 31.17 Interrupt Generation by Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

31.10.2 Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm width measurement) can detect the overflow when TAUBnCNTm of the TAUBTTINm input signal width measurement function exceeds $FFFF_H$.

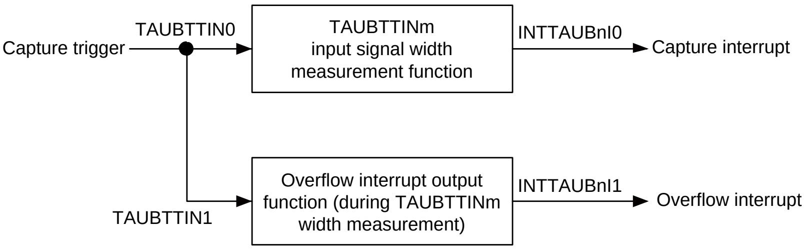


Figure 31.18 Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Timing diagram

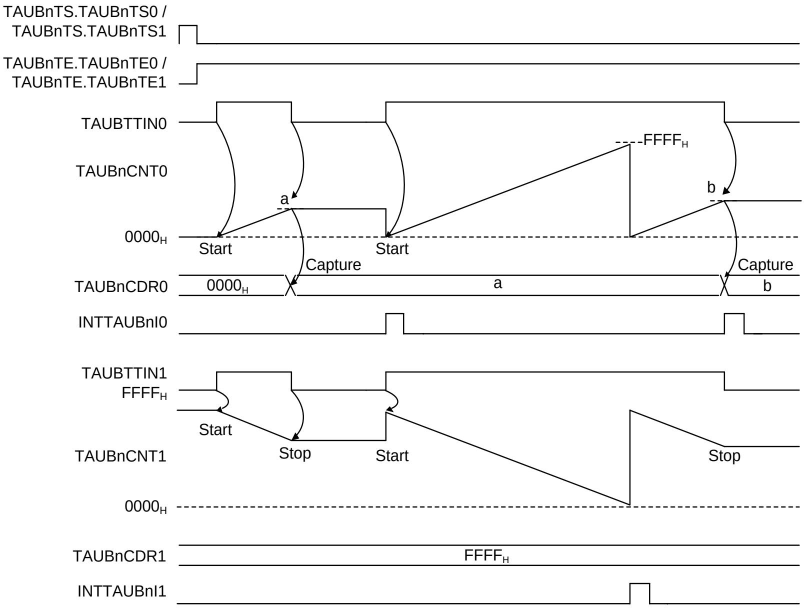


Figure 31.19 Interrupt Generation by Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

31.10.3 Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUBnIm of the interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input position detection function exceeds FFFF$_H$.

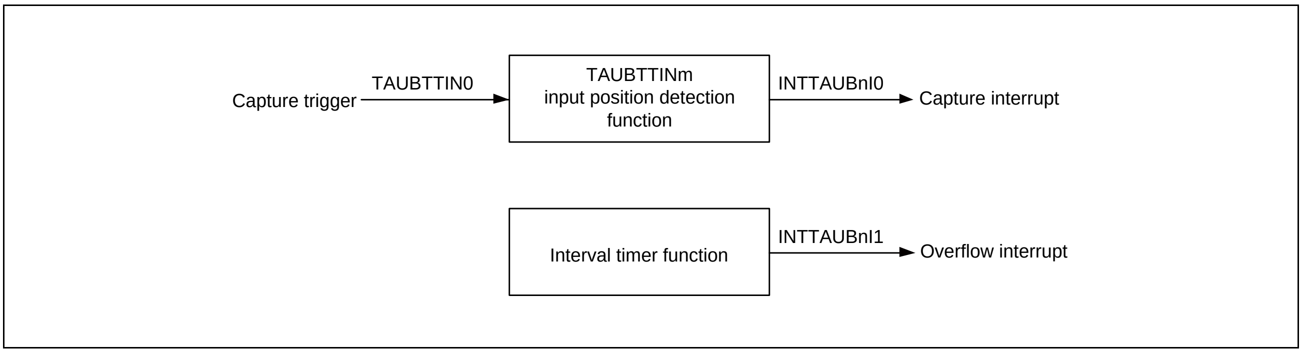


Figure 31.20 Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

Timing diagram

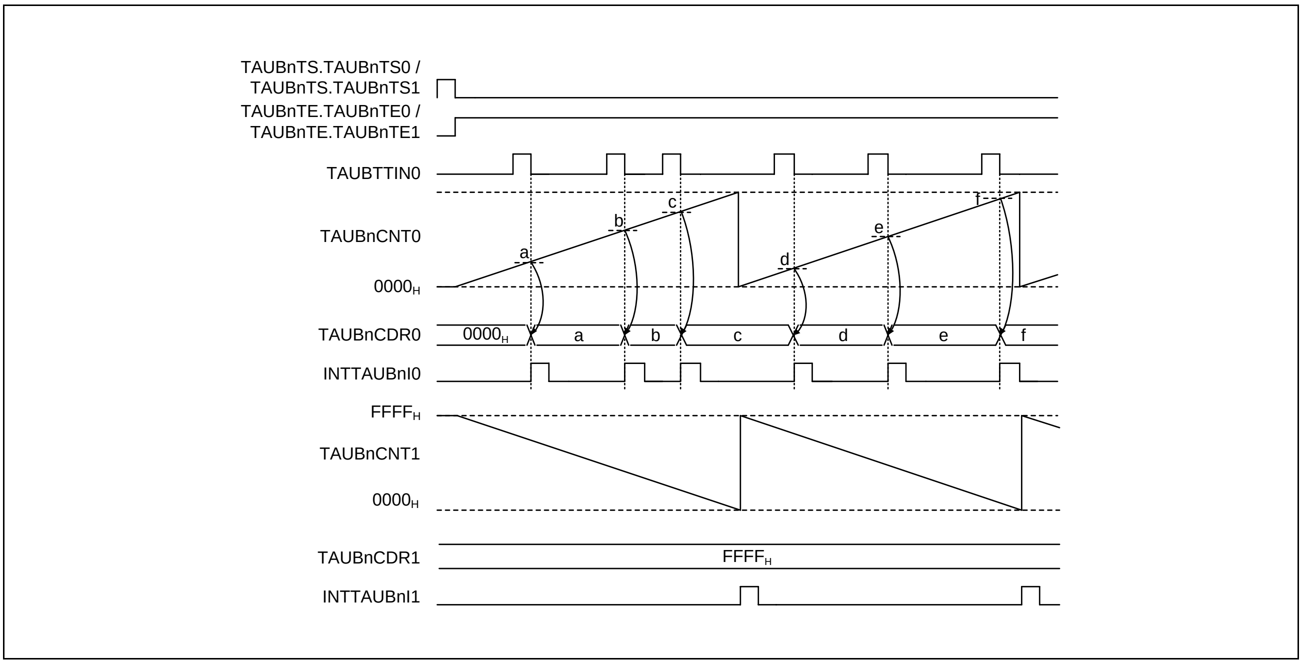


Figure 31.21 Interrupt Generation by Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

31.10.4 Example of Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm input period count detection) can detect the overflow when TAUBnCNTm of the TAUBTTINm input period count detection function exceeds FFFF$_H$.

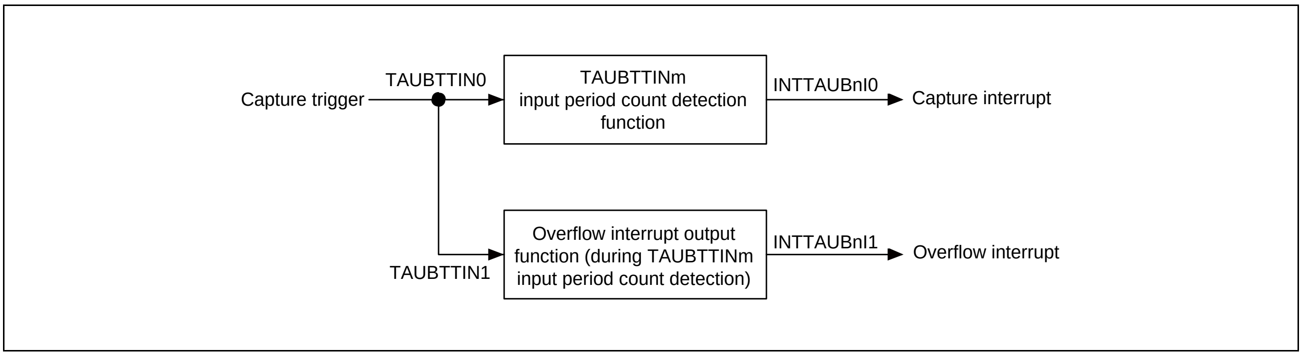


Figure 31.22 Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (TAUBTTINm Input Period Count Detection)

Timing diagram

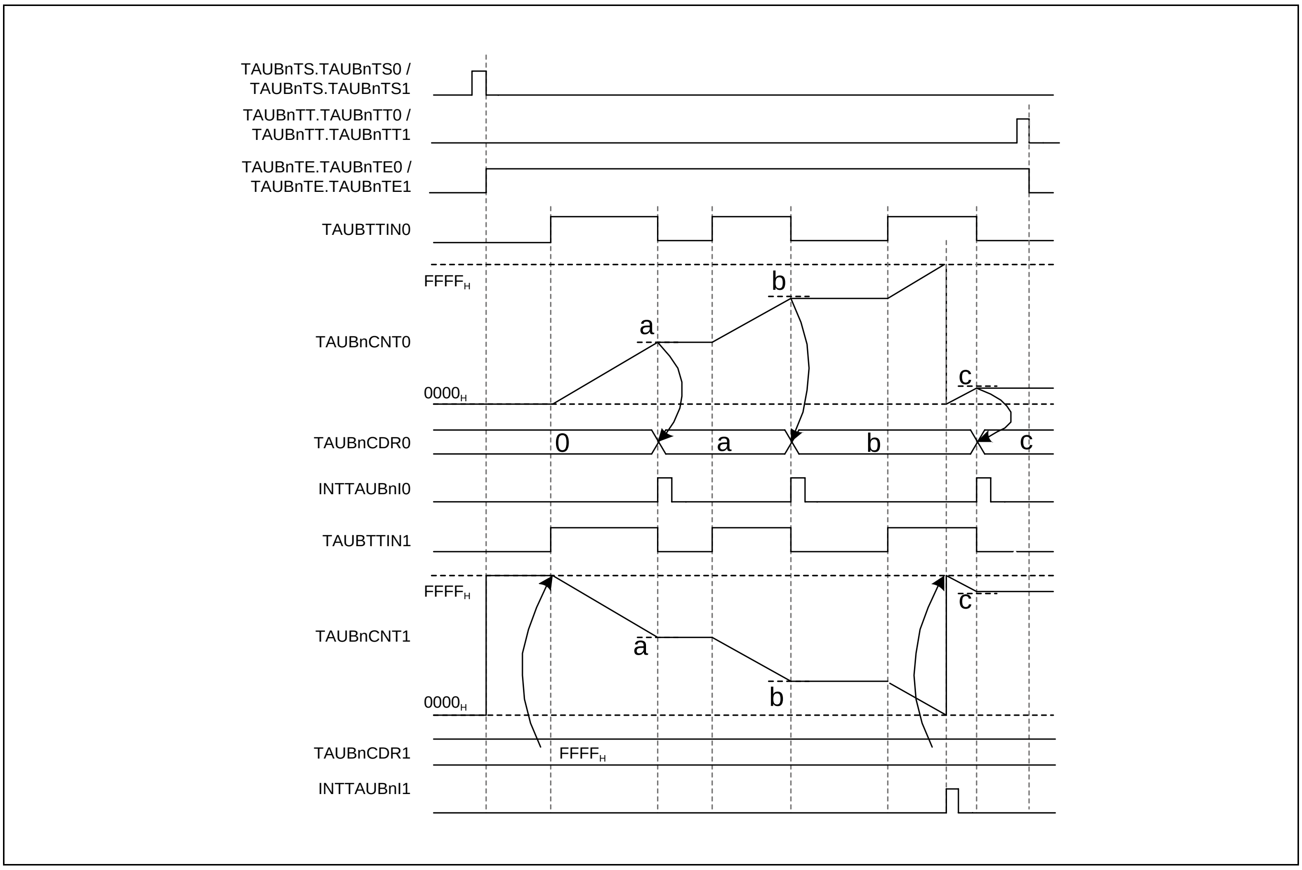


Figure 31.23 Interrupt Generation by Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

31.11 TAUBTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

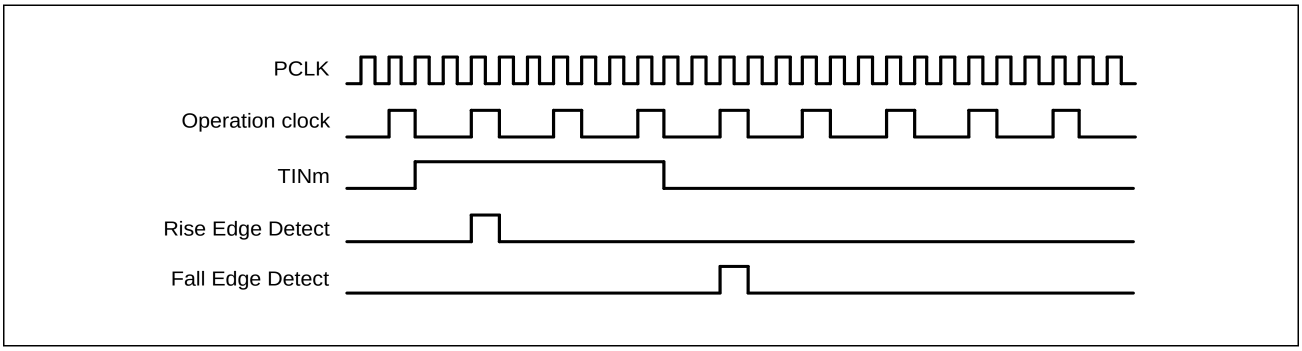


Figure 31.24 Basic Edge Detection Timing

Figure 31.24, Basic Edge Detection Timing is an image of the operation timing. Actually, the delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn will be generated.

31.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUB. For a general overview of independent channel operation functions, see **Section 31.2, Overview**.

31.12.1 Interval Timer Function

31.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal output when TAUBnCMORm.TAUBnMD0 is set to 1.

31.12.1.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

31.12.1.3 Block Diagram and General Timing Diagram

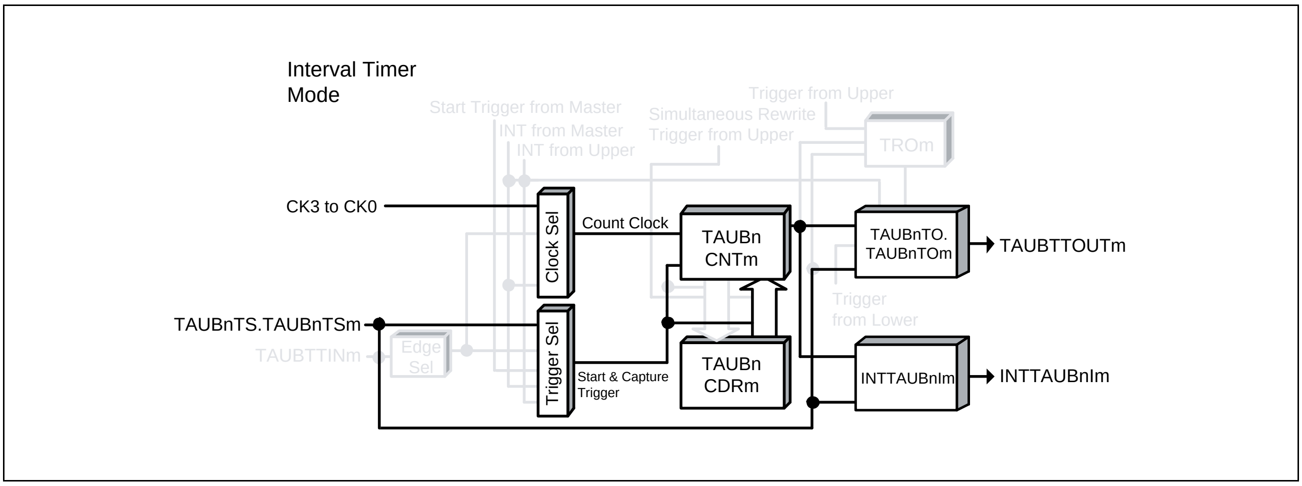


Figure 31.25 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

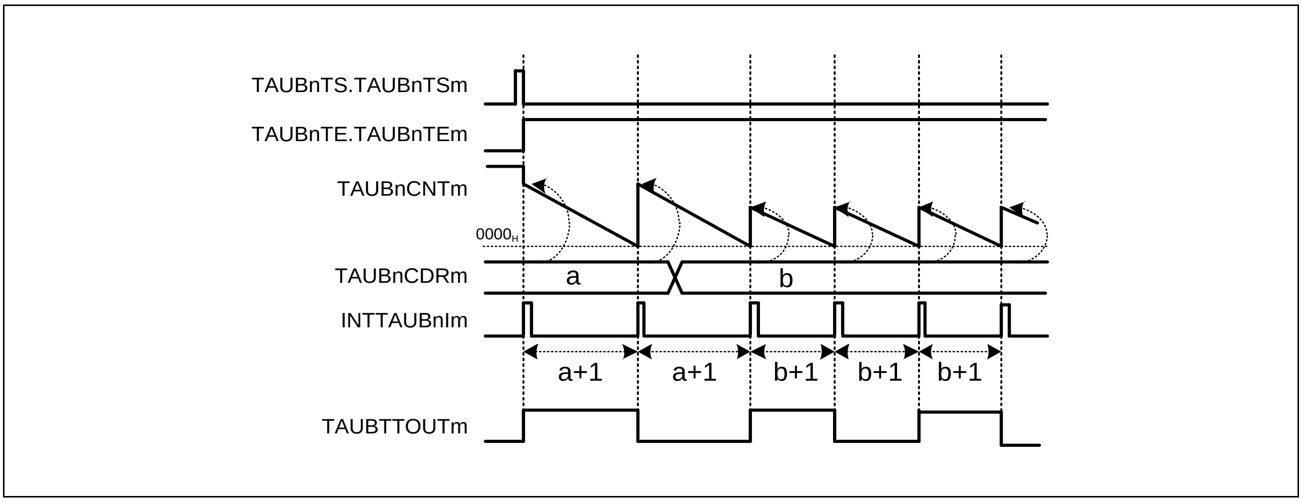


Figure 31.26 General Timing Diagram for Interval Timer Function

31.12.1.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.55 Contents of the TAUBnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm is not generated and TAUBTTOUTm does not toggle at operation start. 1: INTTAUBnIm is generated and TAUBTTOUTm toggles at operation start or restart.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.56 Contents of the TAUBnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 31.57 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 31.58 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.1.5 Operating Procedure for Interval Timer Function

Table 31.59 Operating Procedure for Interval Timer Function

		Operation	Status of TAUBn
	Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.55, Contents of the TAUBnCMORm Register for Interval Timer Function** and **Table 31.56, Contents of the TAUBnCMURm Register for Interval Timer Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 31.57, Control Bit Settings for Independent Channel Output Mode 1**	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated and TAUBTTOUTm toggles.
	During operation	The TAUBnCDRm register value can be changed at any time. The TAUBnCNTm register can be read at all times.	● TAUBnCNTm counts down. When the counter reaches 0000_H: ● TAUBnCNTm reloads the TAUBnCDRm value and continues count operation INTTAUBnIm is generated and TAUBTTOUTm toggles.
	Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

31.12.1.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000$_H$, count clock = PCLK/2

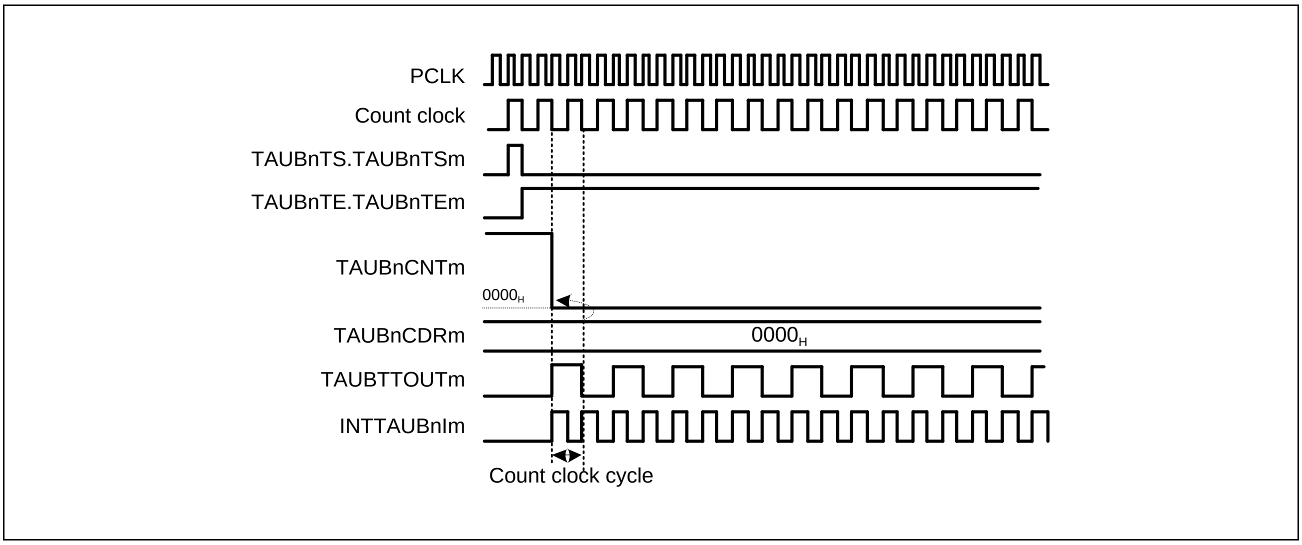


Figure 31.27 TAUBnCDRm = 0000$_H$, Count Clock = PCLK/2

- TAUBnCDRm = 0000$_H$, and the count clock = PCLK/2, the TAUBnCDRm value is written to TAUBnCNTm every count clock, meaning that TAUBnCNTm is always 0000$_H$.
- INTTAUBnIm is generated every count clock, resulting in TAUBTTOUTm toggling every count clock.

(2) TAUBnCDRm = 0000_H, count clock = PCLK

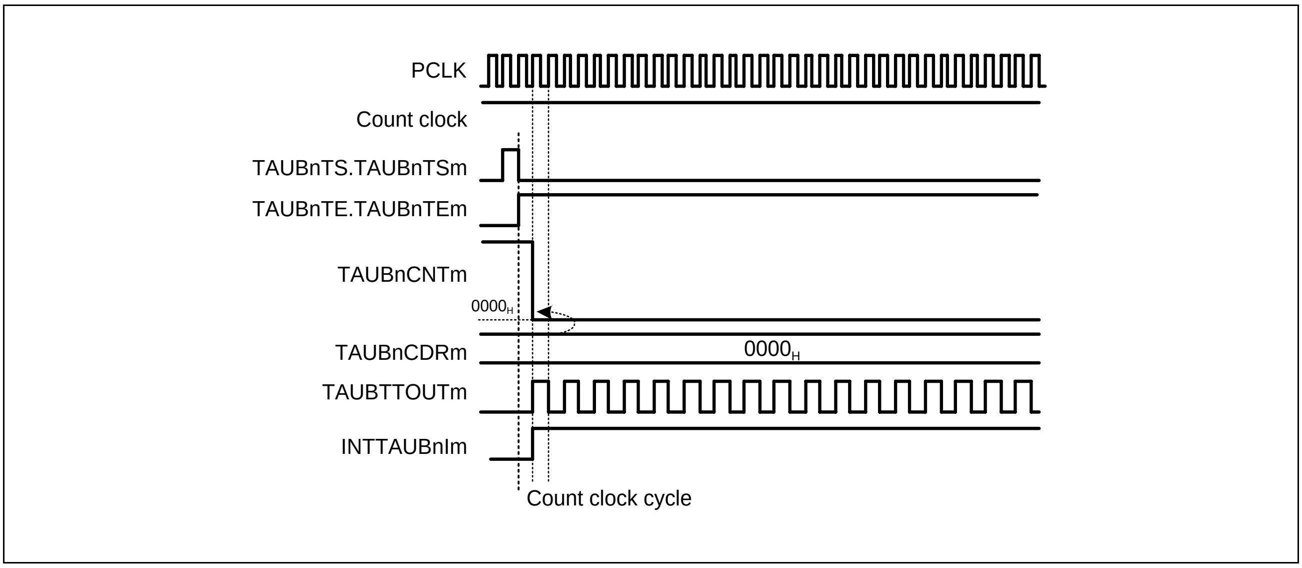


Figure 31.28 TAUBnCDRm = 0000_H, Count Clock = PCLK

- TAUBnCDRm = 0000_H, and the count clock = PCLK, the TAUBnCDRm value is written to TAUBnCNTm every PCLK clock, meaning that TAUBnCNTm is always 0000_H.
- INTTAUBnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUBTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart (TAUBnCMORm TAUBnMD0 = 1)

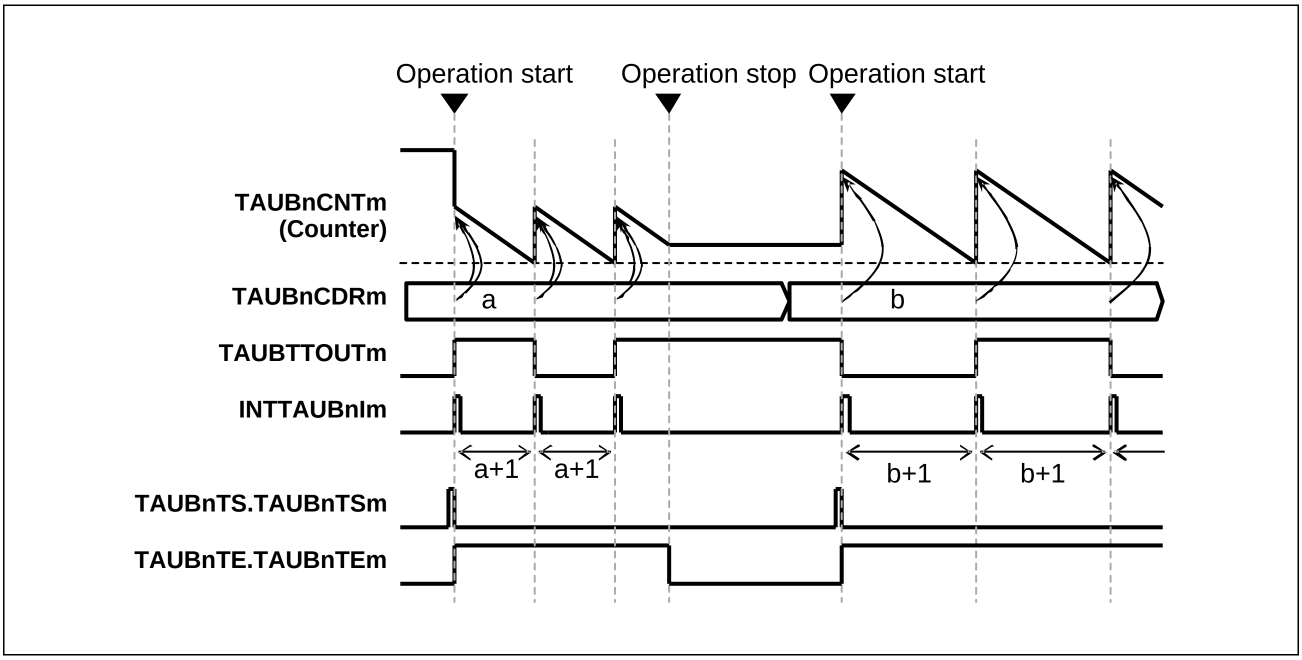


Figure 31.29 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1.

(4) Operation stop and restart (TAUBnCMORm.TAUBnMD0 = 0)

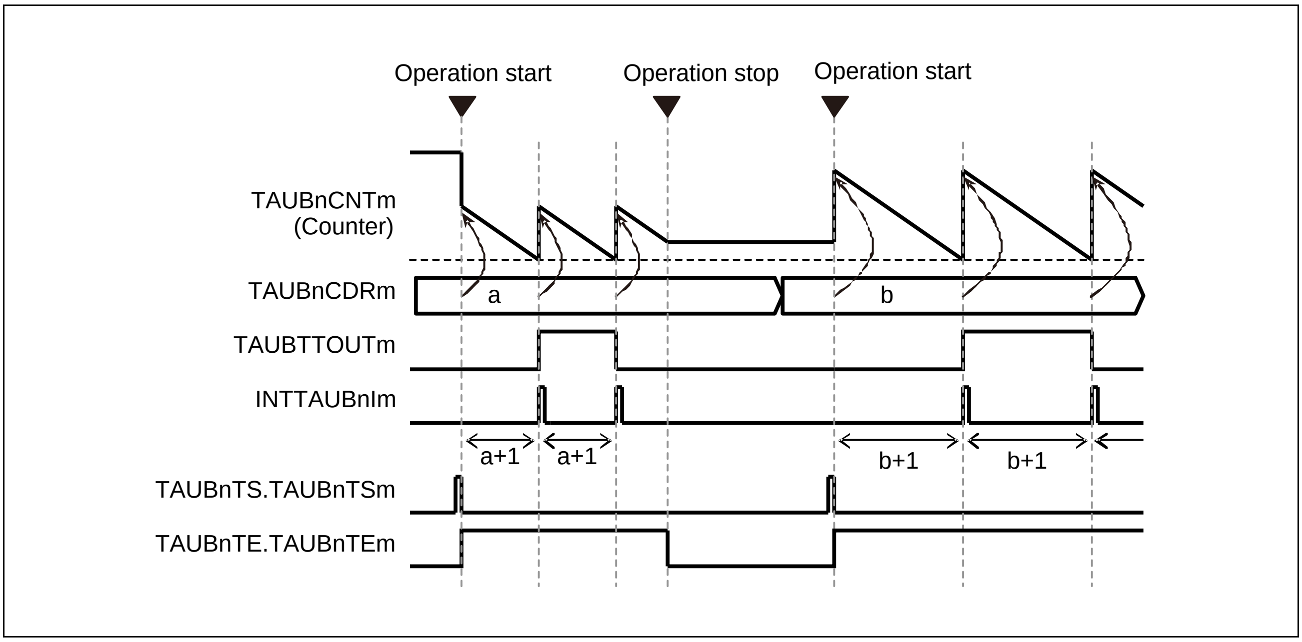


Figure 31.30 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 0

(5) Forced restart (TAUBnCMORm.TAUBnMD0 = 1)

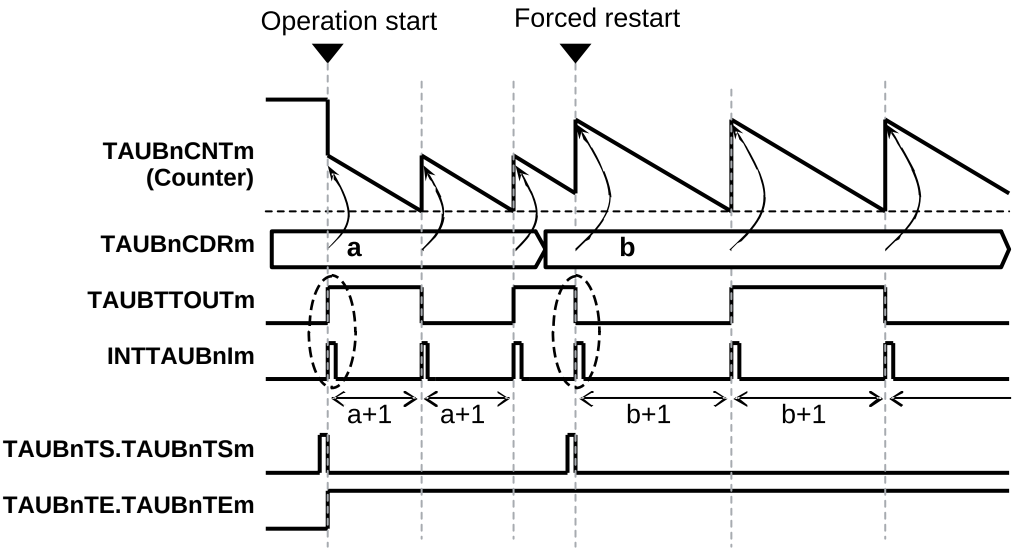


Figure 31.31 Forced Restart Operation, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt at start or restart is generated and the output TAUBTTOUTm toggles.

(6) Forced restart (TAUBnCMORm.TAUBnMD0 = 0)

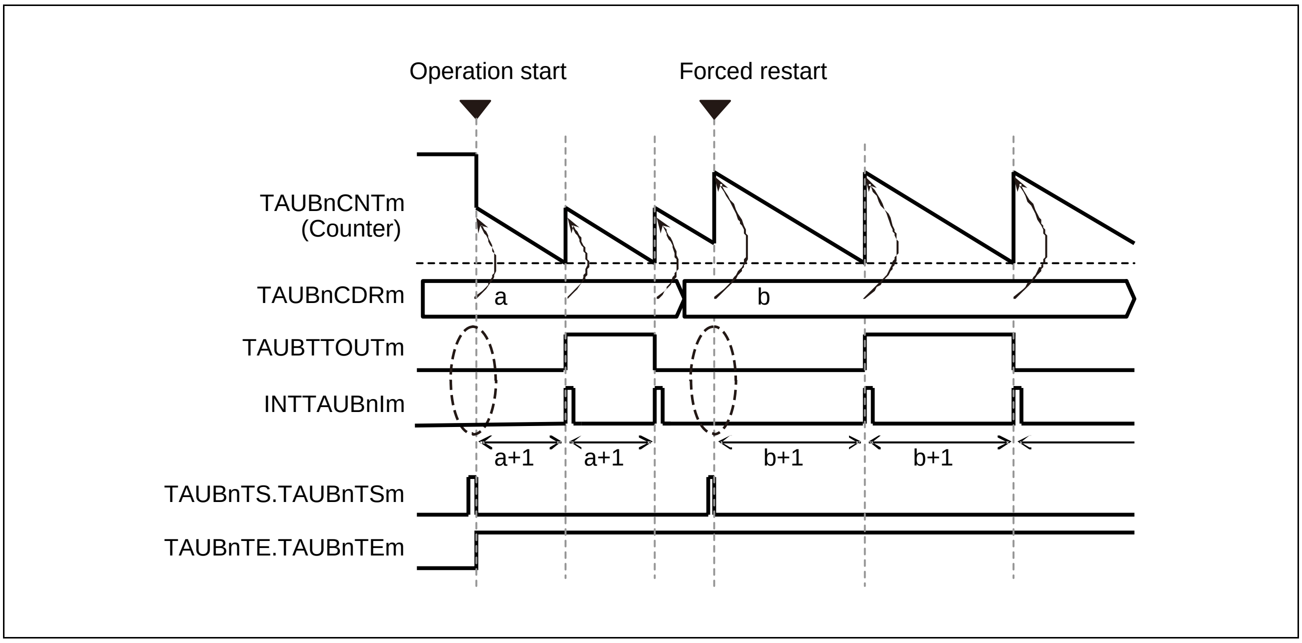


Figure 31.32 Forced Restart Operation (TAUBnCMORm.TAUBnMD0 = 0)

- The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle.

31.12.2 TAUBTTINm Input Interval Timer Function

31.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals or when a valid TAUBTTINm input edge is detected.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

INTTAUBnIm is generated when the counter reaches 0000_H or by an effective TAUBTTINm input edge. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The type of edge used as the trigger is specified using the TAUBnCMURm.TAUBnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

31.12.2.2 Block Diagram and General Timing Diagram

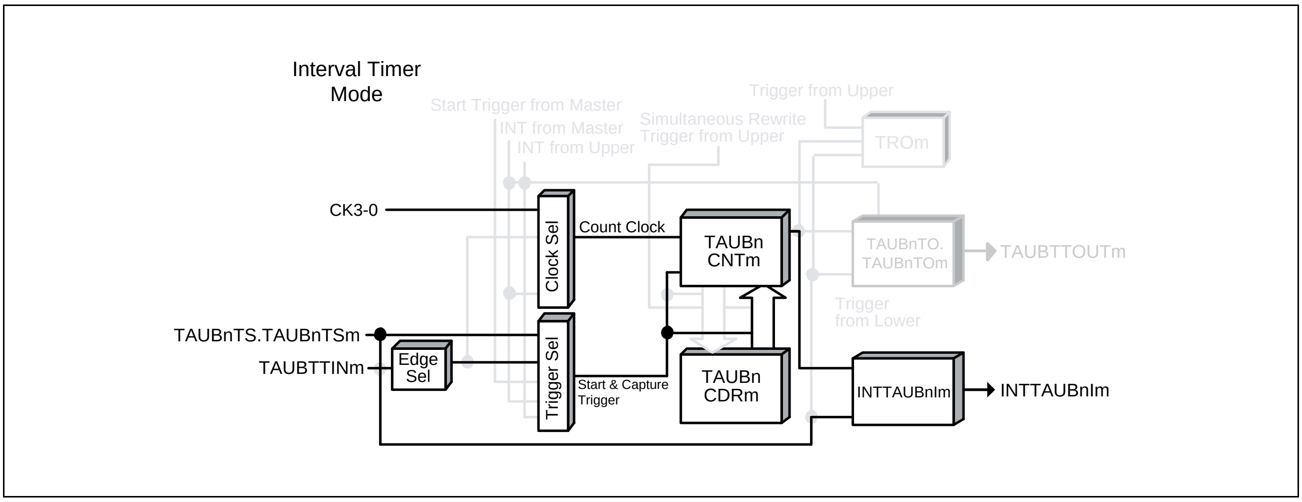


Figure 31.33 Block Diagram for TAUBTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1).
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

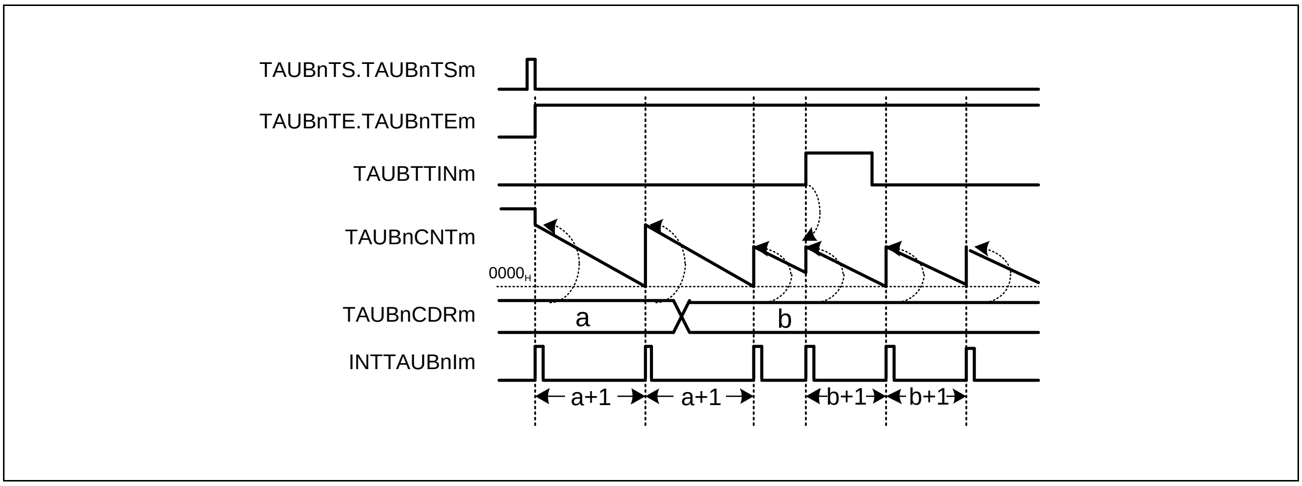


Figure 31.34 General Timing Diagram for TAUBTTINm Input Interval Timer Function

31.12.2.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.60 Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.61 Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 31.62 Simultaneous Rewrite Settings for TAUBTTINm Input Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.2.4 Operating Procedure for TAUBTTINm Input Interval Timer Function

Table 31.63 Operating Procedure for TAUBTTINm Input Interval Timer Function

Restart operation

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.60, Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function** and **Table 31.61, Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function** Set the value of the TAUBnCDRm register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	The values of the TAUBnCMURm.TAUBnTIS[1:0] and the TAUBnCDRm register can be changed at any time. The TAUBnCNTm register can be read at all times. Detection of TAUBTTINm edge	TAUBnCNTm counts down. When the counter reaches 0000_H: ● TAUBnCNTm reloads the TAUBnCDRm value and continues count operation ● INTTAUBnIm is generated When a TAUBTTINm input valid edge is detected during count operation, TAUBnCNTm reloads the TAUBnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1 TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stop and retain their current values.

31.12.2.5 Specific Timing Diagrams

The timing diagrams in **Section 31.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by an effective TAUBTTINm input edge.

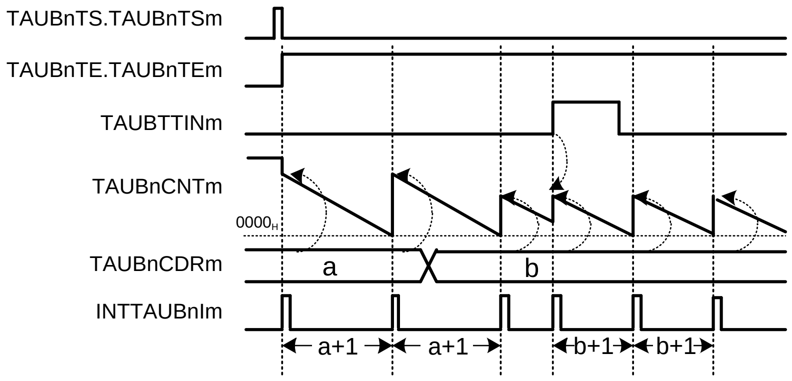


Figure 31.35 Counter Triggered by Rising TAUBTTINm Input Edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B), TAUBnCMORm.TAUBnMD0 = 1

- If an effective TAUBTTINm input edge is detected, an interrupt INTTAUBnIm is generated. In this example, the effective edge is a rising edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B).

31.12.3 Clock Divide Function

31.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUBTTINm is divided by a factor related to TAUBnCDRm, and an interrupt INTTAUBnIm is generated.

Prerequisites

- TAUBTTINm must have a fixed frequency
- The operation mode must be set to interval timer mode, see **Table 31.64, Contents of the TAUBnCMORm Register for Clock Divide Function**

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value, using TAUBTTINm as the count clock.

When the counter value reaches 0000_H, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retain their values. The function can be restarted by setting TAUBnTS.TAUBnTSm = 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits.

31.12.3.2 Block Diagram and General Timing Diagram

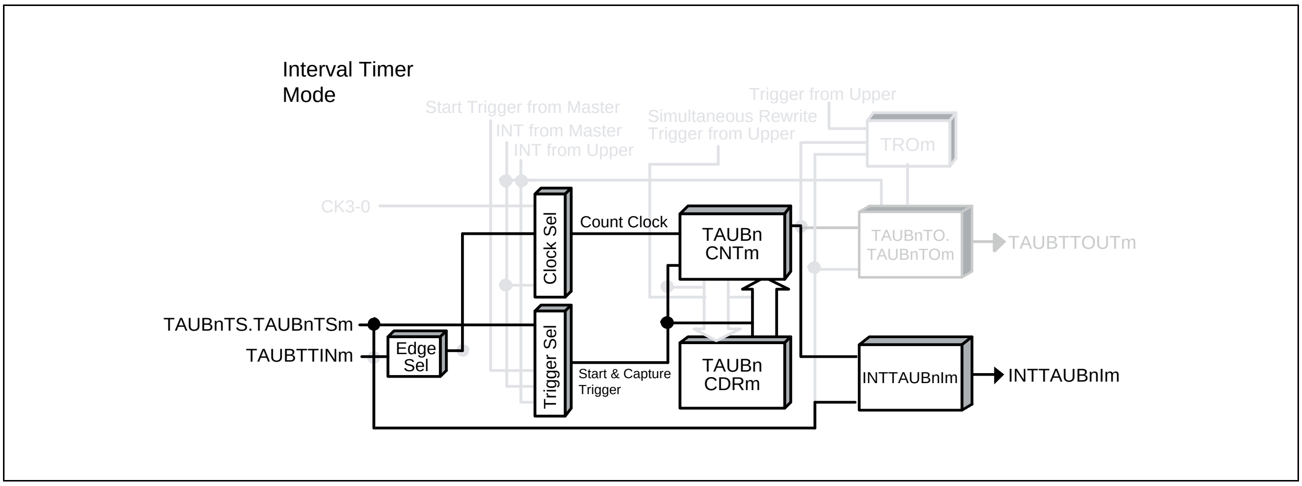


Figure 31.36 Block Diagram for Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

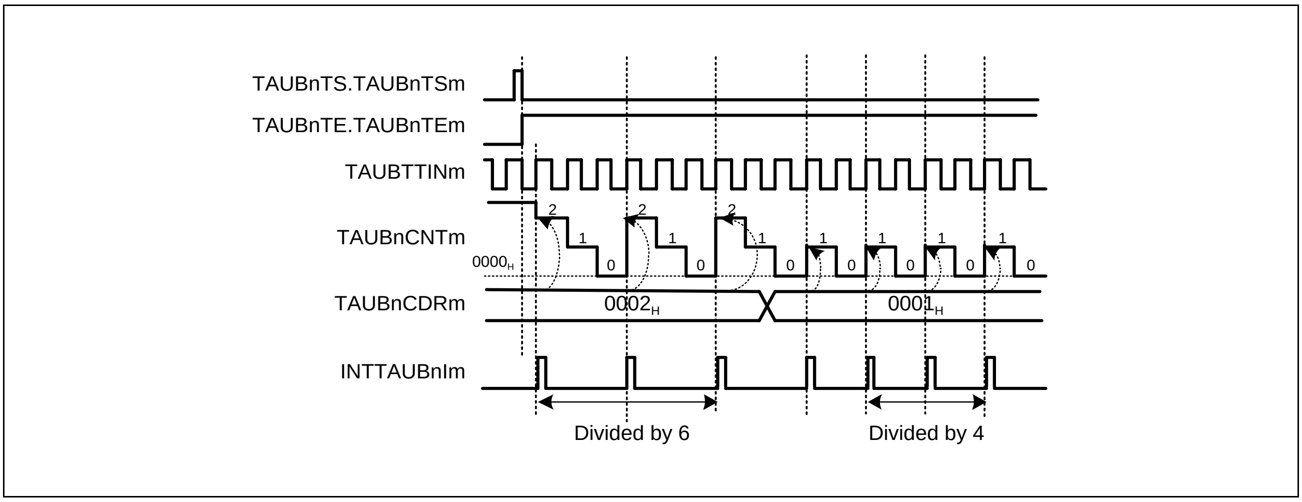


Figure 31.37 General Timing Diagram for Clock Divide Function

31.12.3.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.64 Contents of the TAUBnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.65 Contents of the TAUBnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Clock Divide Function. Therefore, these registers must be set to 0.

Table 31.66 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.3.4 Operating Procedure for Clock Divide Function

Table 31.67 Operating Procedure for Clock Divide Function

		Operation	Status of TAUBn
	Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.64, Contents of the TAUBnCMORm Register for Clock Divide Function** and **Table 31.65, Contents of the TAUBnCMURm Register for Clock Divide Function** Set the value of the TAUBnCDRm register.	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 is set to 1, INTTAUBnIm is generated.
	During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	When a TAUBTTINm input edge is detected, TAUBnCNTm counts down. When the counter reaches 0000_H: ● TAUBnCNTm loads the TAUBnCDRm value and continues count operation ● INTTAUBnIm is generated
	Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.3.5 Specific Timing Diagrams

(1) TAUBnCDRm = 0000$_H$

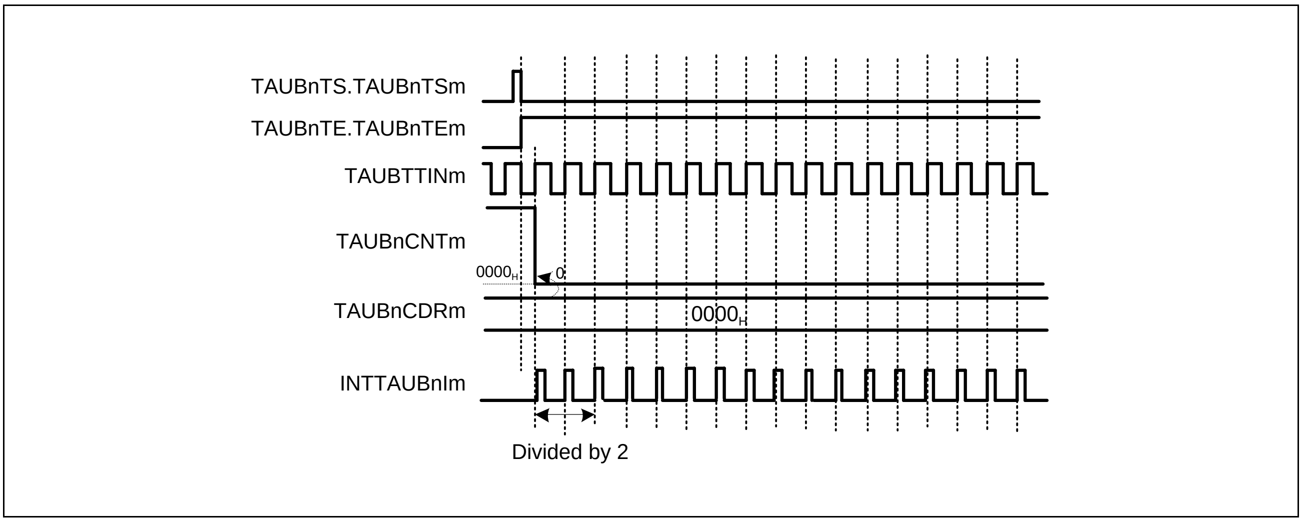


Figure 31.38 TAUBnCDRm = 0000$_H$, TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01$_B$

- If TAUBnCDRm is 0000$_H$, TAUBnCNTm is also always 0000$_H$.
- INTTAUBnIm is generated every count clock.

Figure 31.38, TAUBnCDRm = 0000H, TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01B is an image of the operation timing. Actually, there is a delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn.

(2) Operation restart

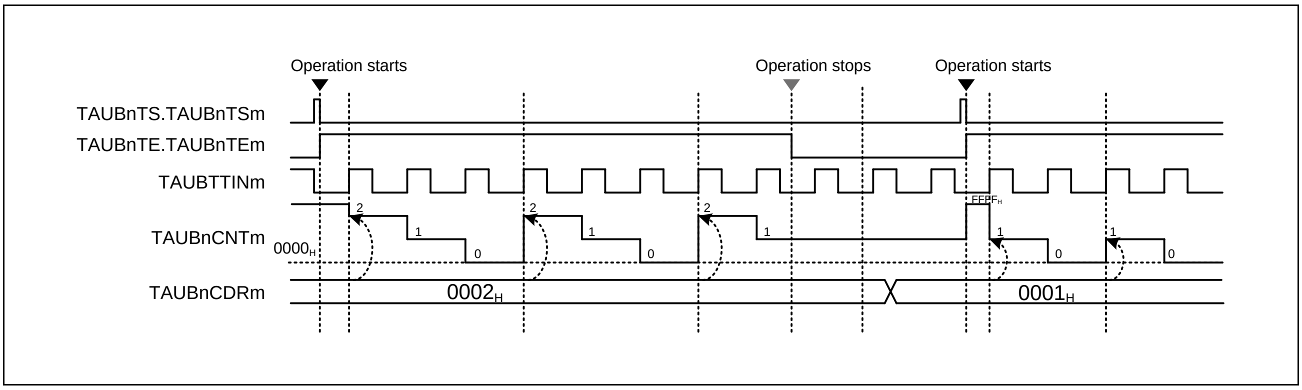


Figure 31.39 Operation Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01$_B$)

(3) Forced restart

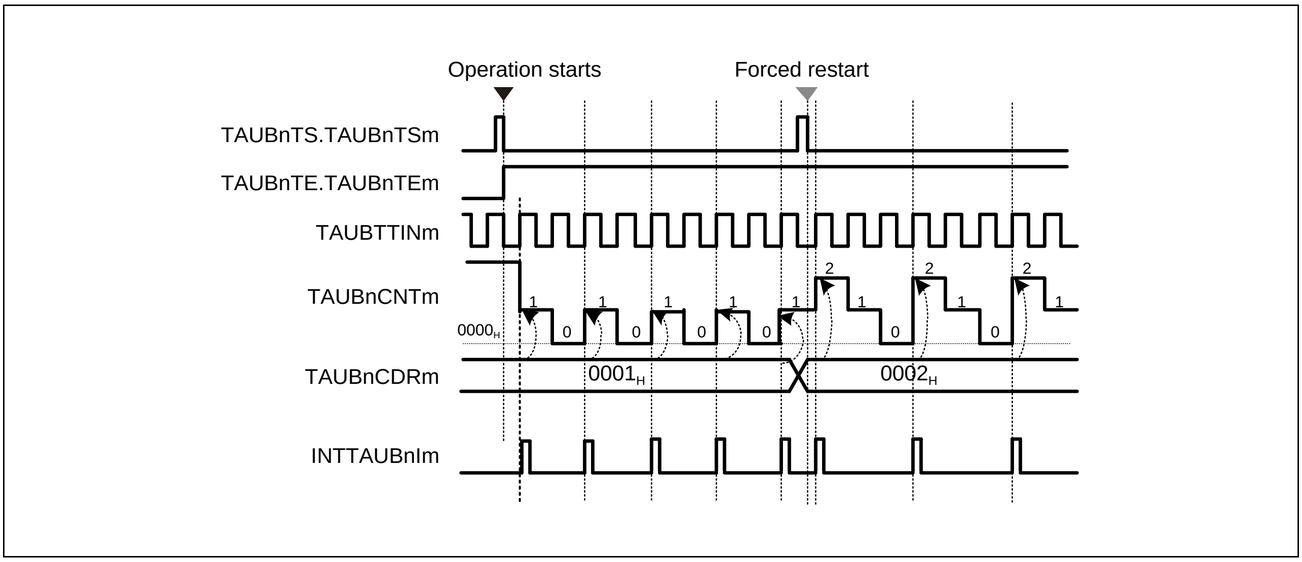


Figure 31.40 Forced Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To forcibly restart the counter.

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- The value of TAUBnCDRm is written to TAUBnCNTm and the count operation restarts.

31.12.4 External Event Count Function

31.12.4.1 Overview

Summary

This function is used as an event timer. It generates an interrupt (INTTAUBnIm) when a specific number of valid edges of TAUBTTINm input are detected.

Prerequisites

- The operation mode must be set to event count mode, see **Table 31.68, Contents of the TAUBnCMORm Register for External Event Count Function**
- TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When the counter starts, the current value of TAUBnCDRm is written to TAUBnCNTm.

When an effective TAUBTTINm input edge is detected, the value of TAUBnCNTm reduces by 1. TAUBnCNTm retains this value until a valid TAUBTTINm input edge is detected or the counter is restarted.

When effective edges are detected (TAUBnCDRm + 1) times, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues to operate.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

31.12.4.2 Equations

Number of valid edges,
detected before INTTAUBnIm is generated = TAUBnCDRm + 1

31.12.4.3 Block Diagram and General Timing Diagram

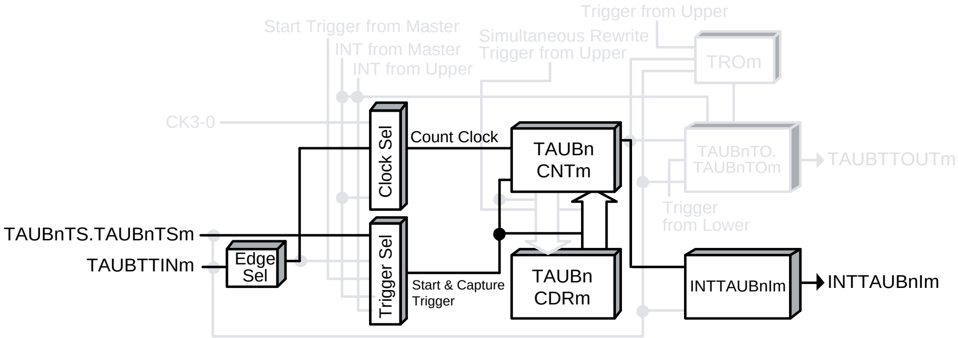


Figure 31.41 Block Diagram for External Event Count Function

The following settings apply to the general timing diagram.

- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

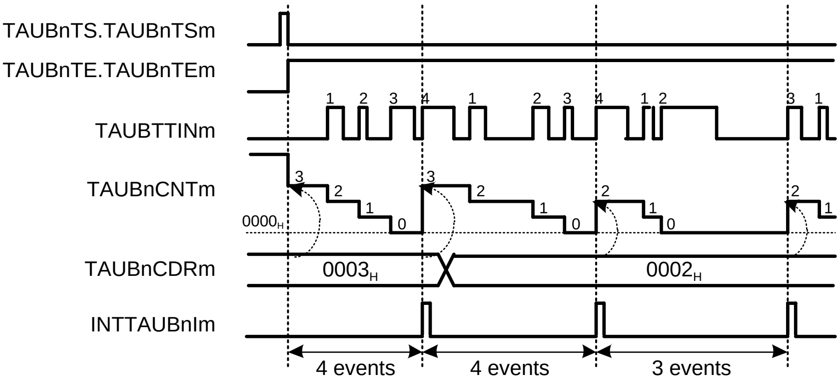


Figure 31.42 General Timing Diagram for External Event Count Function

31.12.4.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.68 Contents of the TAUBnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0011_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.69 Contents of the TAUBnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the External Event Count Function.

Therefore, these registers must be set to 0.

Table 31.70 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.4.5 Operating Procedure for External Event Count Function

Table 31.71 Operating Procedure for External Event Count Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.68, Contents of the TAUBnCMORm Register for External Event Count Function** and **Table 31.69, Contents of the TAUBnCMURm Register for External Event Count Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value and waits for detection of the TAUBTTINm input edge.
During operation	Detection of TAUBTTINm edges. The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at any time.	TAUBnCNTm performs count-down operation each time a TAUBTTINm input edge is detected. When effective edges are detected (TAUBnCDRm + 1) times: ● TAUBnCNTm loads the TAUBnCDRm value and continues count operation ● INTTAUBnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

31.12.4.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

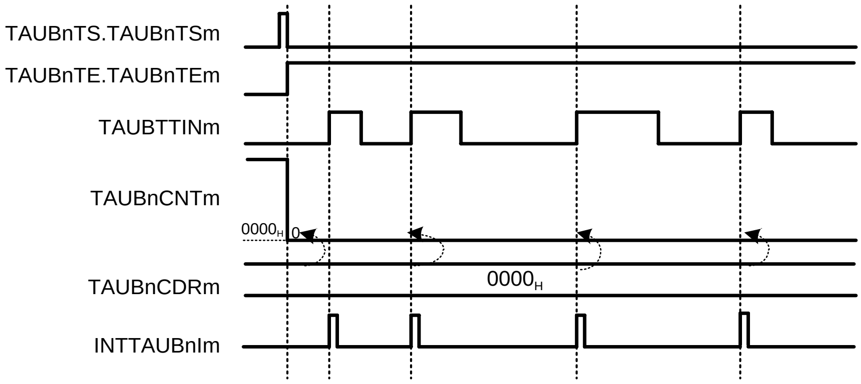


Figure 31.43 TAUBnCDRm = 0000_H, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If 0000_H = TAUBnCDRm, 0000_H is loaded to TAUBnCNTm every time a valid TAUBTTINm input edge is detected.

This means, INTTAUBnIm is generated every time a valid TAUBTTINm input edge is detected.

(2) Operation stop and restart

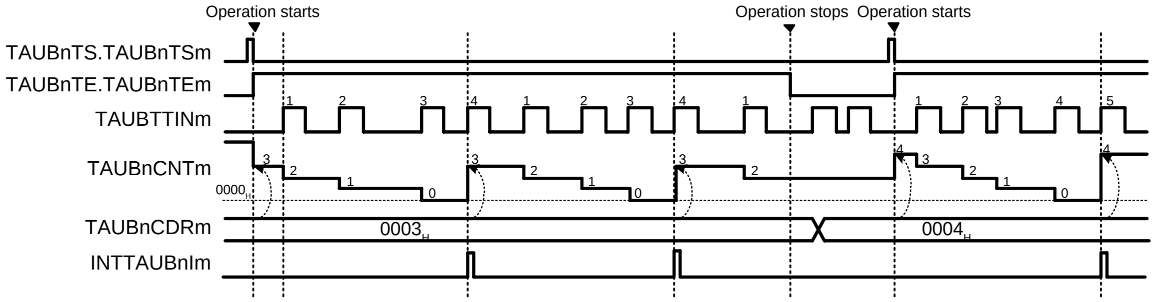


Figure 31.44 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained. TAUBTTINm continues and TAUBnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm loads the TAUBnCDRm value and restarts count operation.

(3) Forced restart

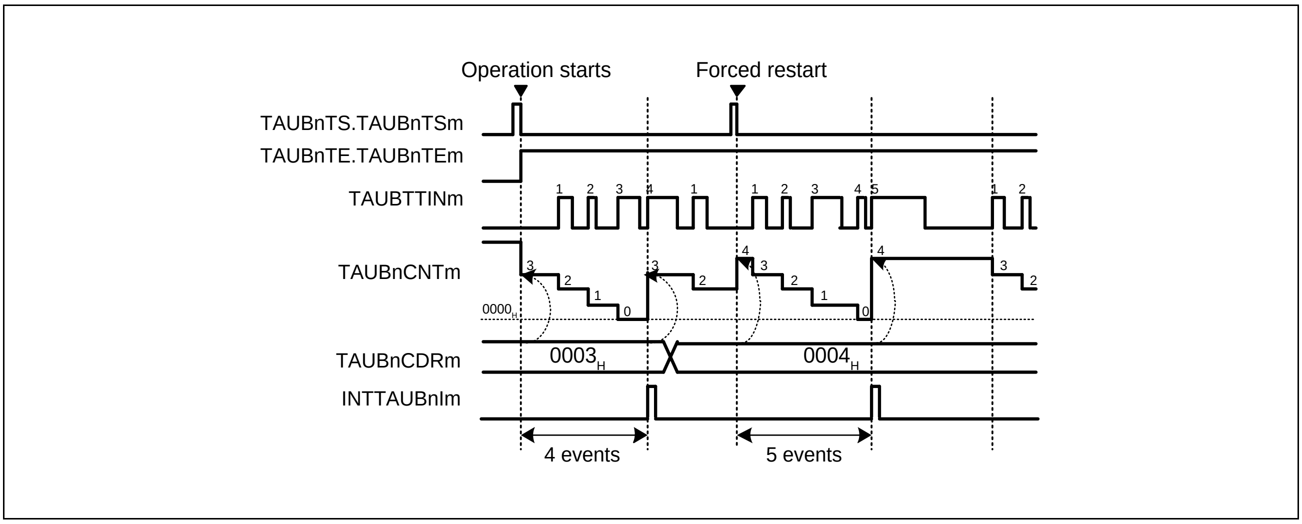


Figure 31.45 Forced Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

A forced restart applies the new TAUBnCDRm value to TAUBnCNTm immediately.

- The counter can be restarted (without stopping it first), by setting TAUBnTS.TAUBnTSm to 1 during operation.
- The value of TAUBnCDRm is loaded to TAUBnCNTm and the counter awaits the next valid TAUBTTINm input edge.

31.12.5 One-Pulse Output Function

31.12.5.1 Overview

Summary

This function generates an interrupt (INTTAUBnIm) when a valid TAUBTTINm input edge is detected and subsequently, in a specific interval. TAUBTTINm input signal pulses that occur within the defined interval are ignored.

Prerequisites

- The operation mode must be set to pulse one-count mode. (See **Table 31.72, Contents of the TAUBnCMORm Register for One-Pulse Output Function**).
- Trigger detection must be disabled during counting (TAUBnCMORm.TAUBnMD0 = 0).

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input edge is detected. The value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from the TAUBnCDRm value, and an interrupt is generated.

When the counter reaches 0001_H an interrupt is generated. The counter stops at 0000_H and awaits the next effective TAUBTTINm input edge.

When the counter is counting down, further TAUBTTINm input signals are ignored, i.e. the counter does not reset.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

31.12.5.2 Equations

Interval between TAUBTTINm and INTTAUBnIm = count clock cycle × TAUBnCDRm

31.12.5.3 Block Diagram and General Timing Diagram

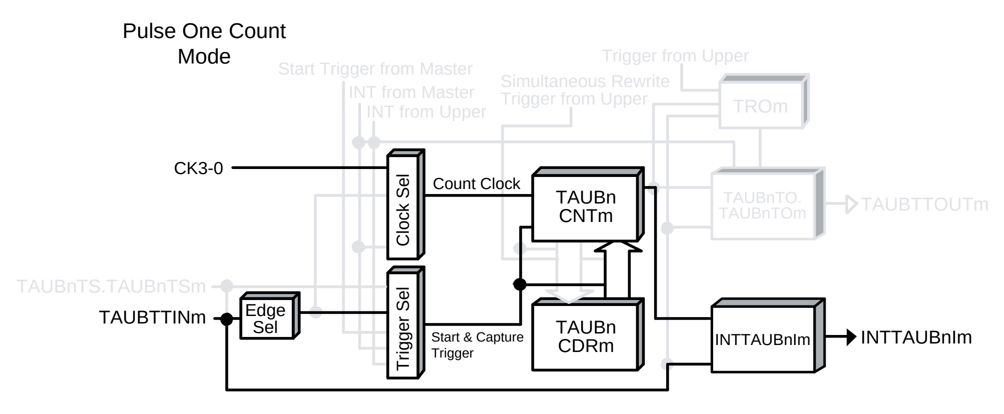


Figure 31.46 Block Diagram for One-Pulse Output Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

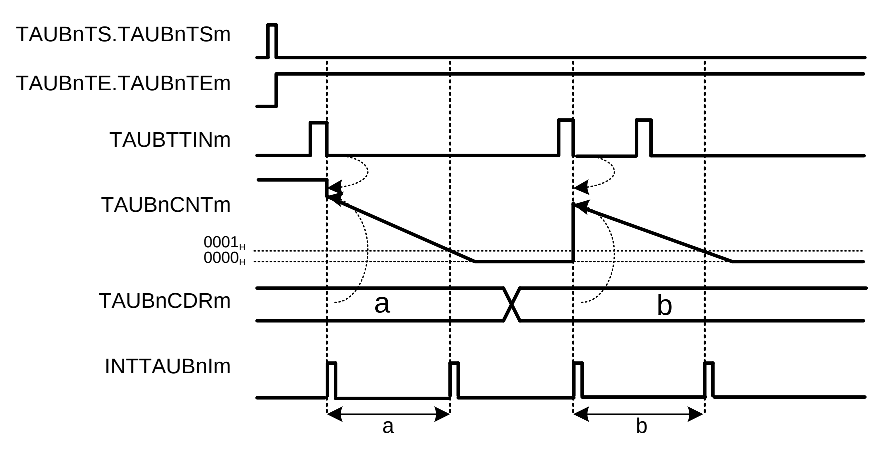


Figure 31.47 General Timing Diagram for One-Pulse Output Function

31.12.5.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.72 Contents of the TAUBnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.73 Contents of the TAUBnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the One-Pulse Output Function.
Therefore, these registers must be set to 0.

Table 31.74 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.5.5 Operating Procedure for One-Pulse Output Function

Table 31.75 Operating Procedure for One-Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.72, Contents of the TAUBnCMORm Register for One-Pulse Output Function** and **Table 31.73, Contents of the TAUBnCMURm Register for One-Pulse Output Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	INTTAUBnIm is generated when TAUBnCNTm starts. TAUBnCNTm counts down. When the counter reaches 0001_H, INTTAUBnIm is generated. TAUBnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUBnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.6 TAUBTTINm Input Pulse Interval Measurement Function

31.12.6.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUBnCSRm.TAUBnOVF to measure the interval of the TAUBTTINm input signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 31.76 Effects of an Overflow

TAUBnCMORm. TAUBnCOS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input is then Detected	
	TAUBnCDRm	TAUBnCSRm. TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm. TAUBnOVF
00	Unchanged	0	TAUBnCNTm loaded to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm set to 0, TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] is 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the interval of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

The function can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retains its value. While the function is stopped, TAUBTTINm input valid edge detection and TAUBnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the interrupt at start or restart is not generated.

NOTE

When TAUBnCMORm.TAUBnCOS[1:0] = 10_B or 11_B, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

31.12.6.2 Equations

TAUBTTINm input pulse interval = count clock cycle ×

[(TAUBnCSRm.TAUBnOVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

31.12.6.3 Block Diagram and General Timing Diagram

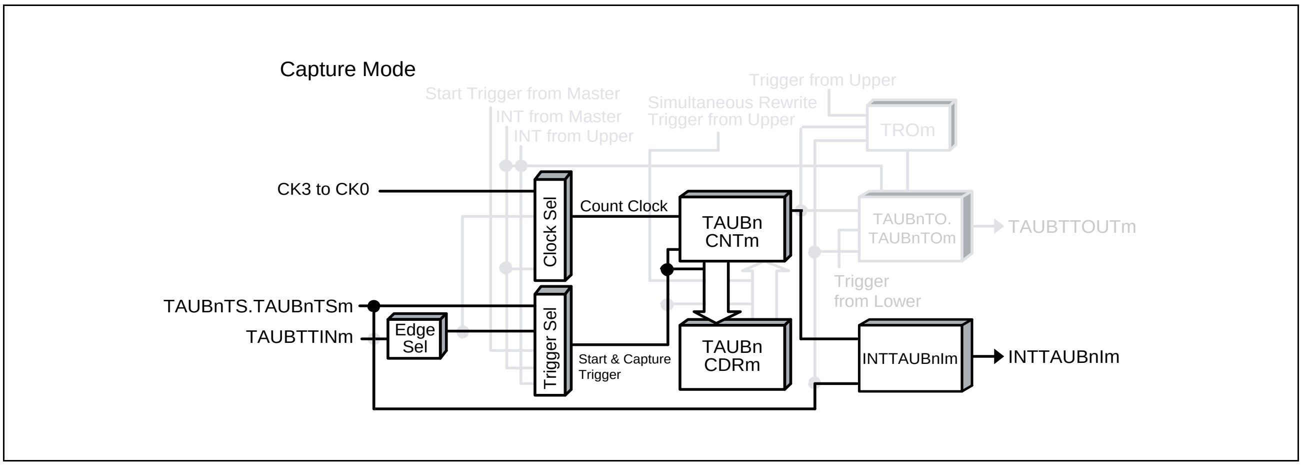


Figure 31.48 Block Diagram for TAUBTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)
- When a valid TAUBTTINm input is detected after an overflow TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

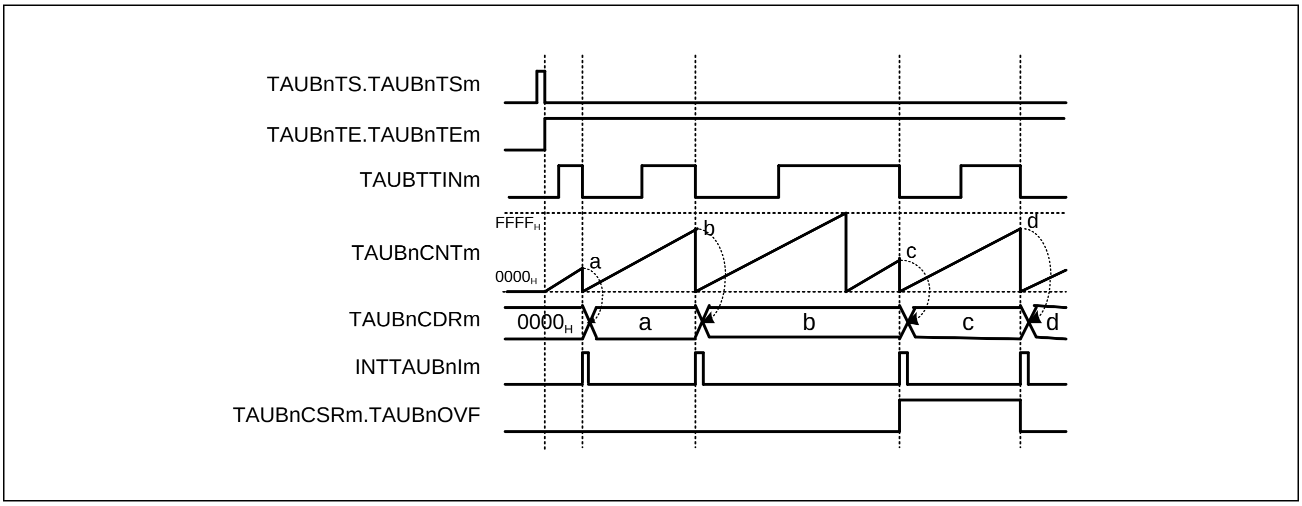


Figure 31.49 General Timing Diagram for TAUBTTINm Input Pulse Interval Measurement Function

31.12.6.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.77 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	See **Table 31.76, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0010_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.78 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 31.79 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.6.5 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

Table 31.80 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.77, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function** and **Table 31.78, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm is cleared to 0000_H. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	Detection of TAUBTTINm edges. The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time. The TAUBnCSCm.TAUBnCLOV bit can be set to 1. (The TAUBnCSRm.TAUBnOVF bit can be cleared to 0.)	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: ● TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and returns to 0000_H ● INTTAUBnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

Restart operation

31.12.6.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

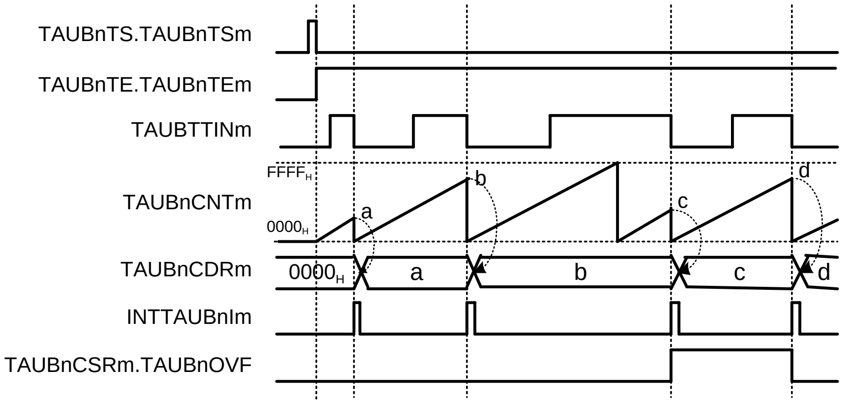


Figure 31.50 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is loaded to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

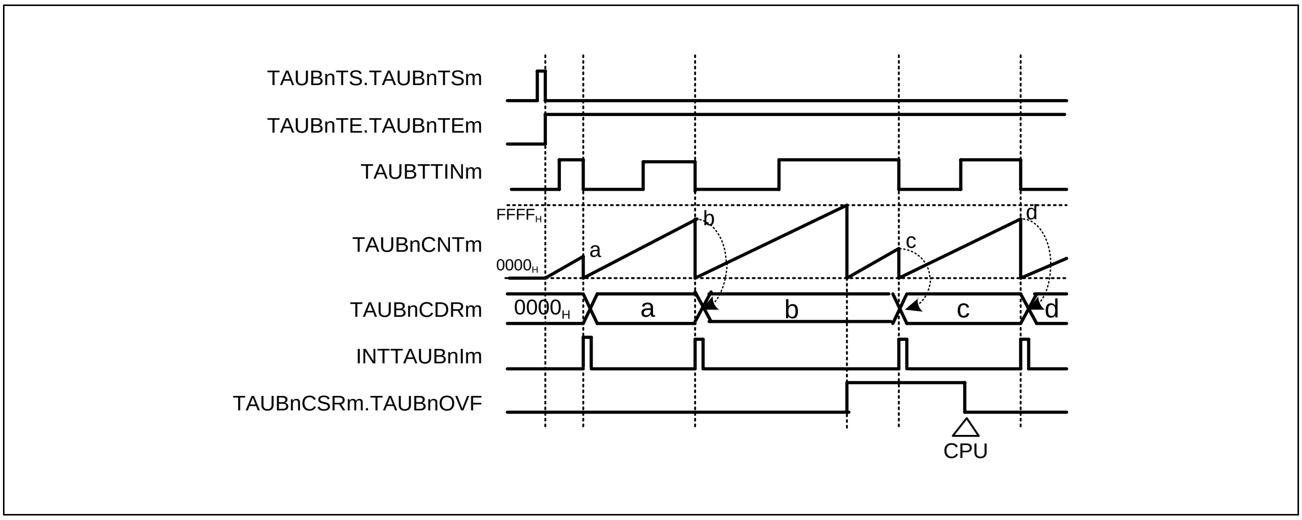


Figure 31.51 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command. (TAUBnCSCm.TAUBnCLOV bit = 1)

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

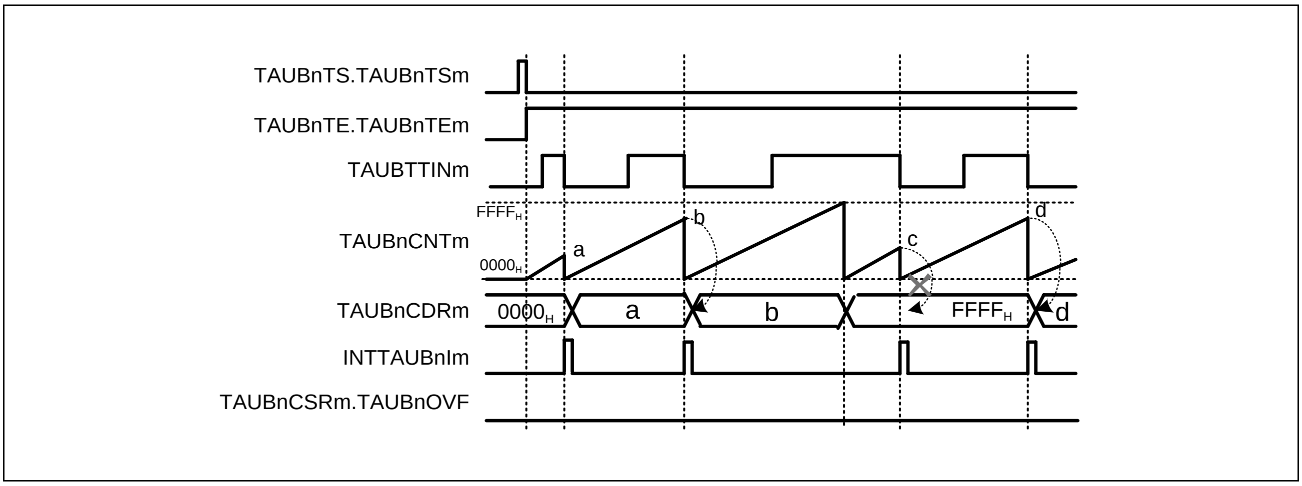


Figure 31.52 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

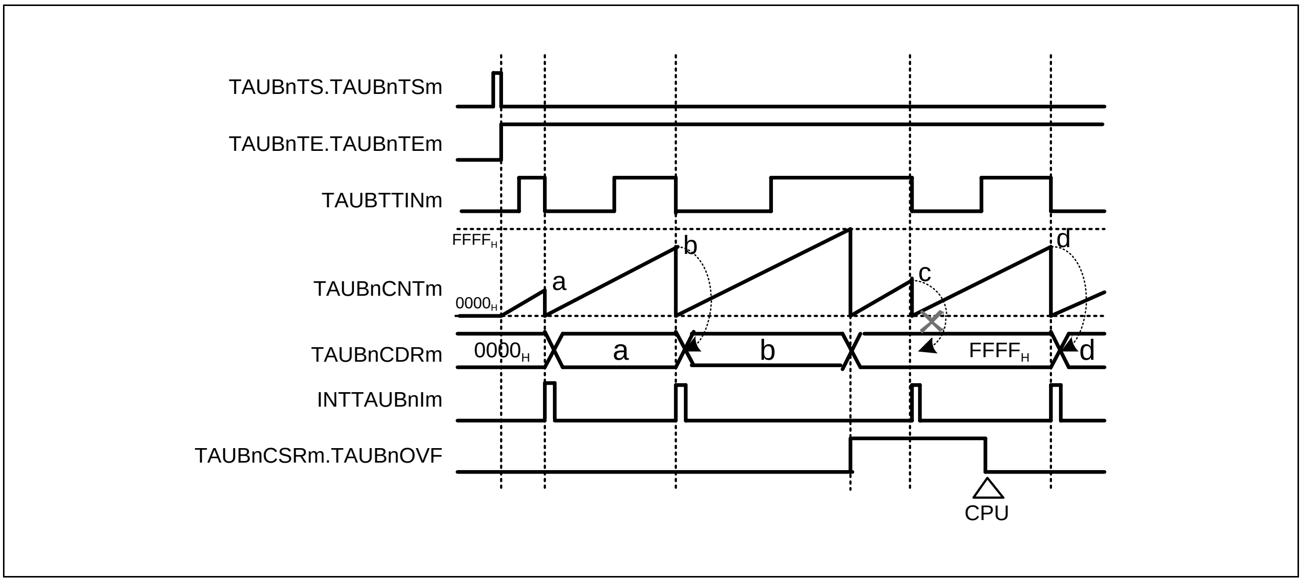


Figure 31.53 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When rising and falling edge detection are selected (TAUBnCMORm.TAUBnMD0 = 1)

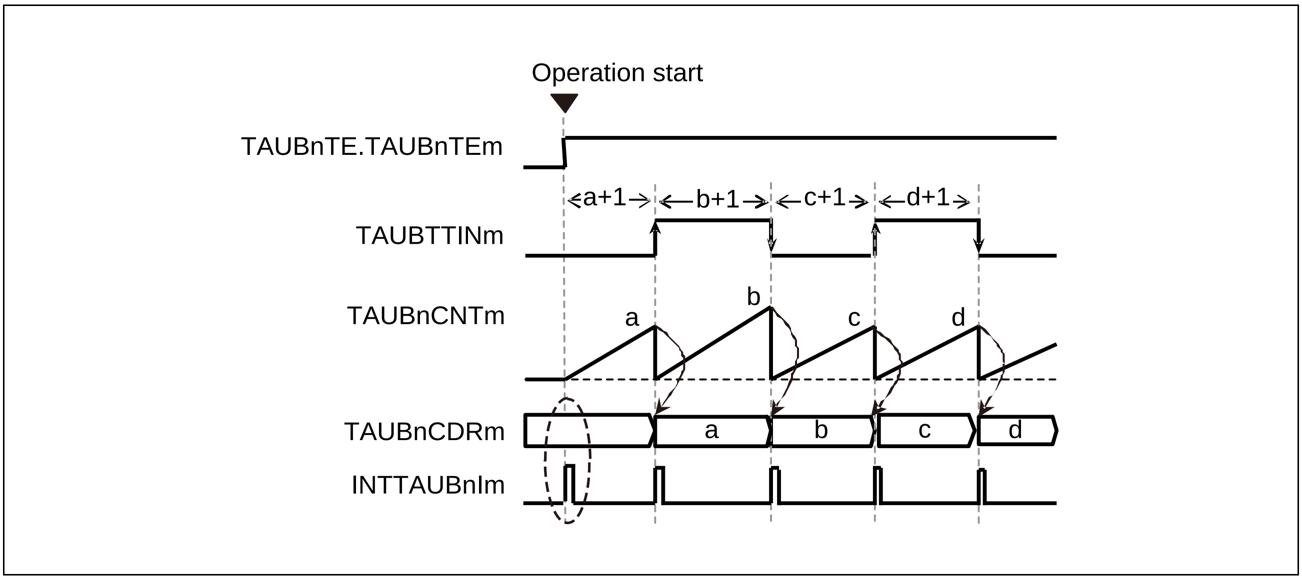


Figure 31.54 TAUBnCMORm.TAUBnMD0 = 1

Setting TAUBnCMURm.TAUBnTIS[1:0] to 10_B (detection of both edges selected) measures the TAUBTTINm rising and falling edge intervals.

(6) Operation stop and operation restart (TAUBnCMORm.TAUBnMD0 = 0)

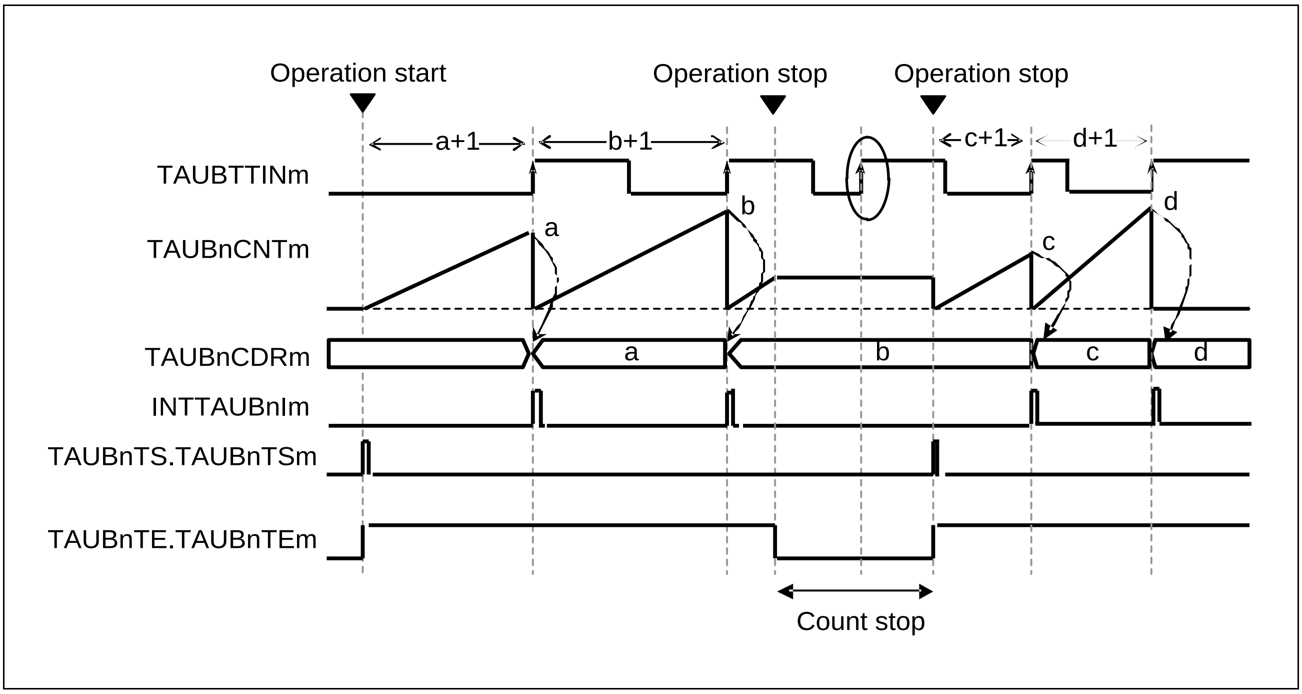


Figure 31.55 Operation Stop and Operation Restart (TAUBnCMORm.TAUBnMD0 = 0)

Setting TAUBnTT.TAUBnTTm to 1 clears TAUBnTE.TAUBnTEm to 0, which stops the count operation. At this time, TAUBnCNTm retains the status and stops.

When TAUBnTE.TAUBnTEm retains 0 (operation stopped), TAUBTTINm input is ignored (edge detection is ignored and capture operation is not performed).

Setting TAUBnTS.TAUBnTSm to 1 clears the counter to 0000_H and restarts count-up operation.

31.12.7 TAUBTTINm Input Signal Width Measurement Function

31.12.7.1 Overview

Summary

This function measures the width of a TAUBTTINm signal by starting counting on one edge of the TAUBTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm start edge is detected, the counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm stop edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter retains its value (TAUBnCDRm + 1) and awaits the next valid TAUBTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 31.81 Effects of an Overflow

TAUBnCMORm. TAUBnCOS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input Stop Edge is Detected	
	TAUBnCDRm	TAUBnCSRm. TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm. TAUBnOVF
00	Unchanged	0	TAUBnCNTm written to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm stops counting TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] = 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the width of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUBnCMORm.TAUBnCOS[1] = 1, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

31.12.7.2 Equations

TAUBTTINm input signal width = count clock cycle ×

[(TAUBnCSRm.OVF × (FFFF$_H$ + 1)) + TAUBnCDRm capture value + 1]

31.12.7.3 Block Diagram and General Timing Diagram

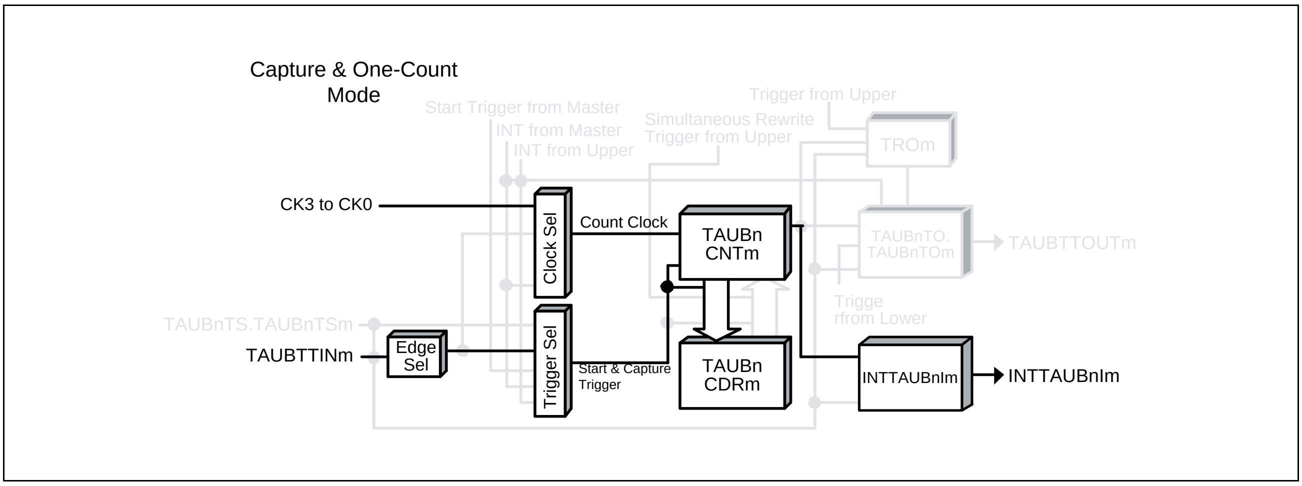


Figure 31.56 Block Diagram for TAUBTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11$_B$)
- When a valid TAUBTTINm input is detected after an overflow, TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00$_B$)

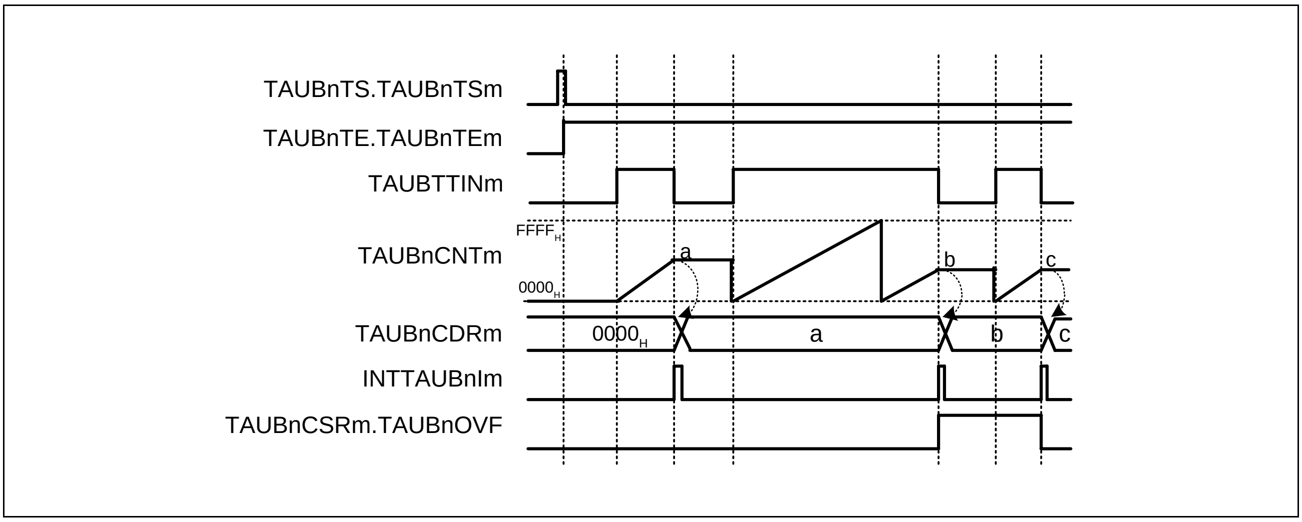


Figure 31.57 General Timing Diagram for TAUBTTINm Input Signal Width Measurement Function

31.12.7.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.82 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	See **Table 31.81, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0110_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.83 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 31.84 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.7.5 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

Table 31.85 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

		Operation	Status of TAUBn
	Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.82, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function** and **Table 31.83, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a TAUBTTINm start edge is detected, TAUBnCNTm start edge to count up.
	During operation	The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time. TAUBnCSCm.TAUBnCLOV bit can be set to 1.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: ● TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and retains its value ● INTTAUBnIm is then generated. ● The count stops at the value transferred to TAUBnCDRm + 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. Afterwards, this procedure is repeated.
	Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

31.12.7.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

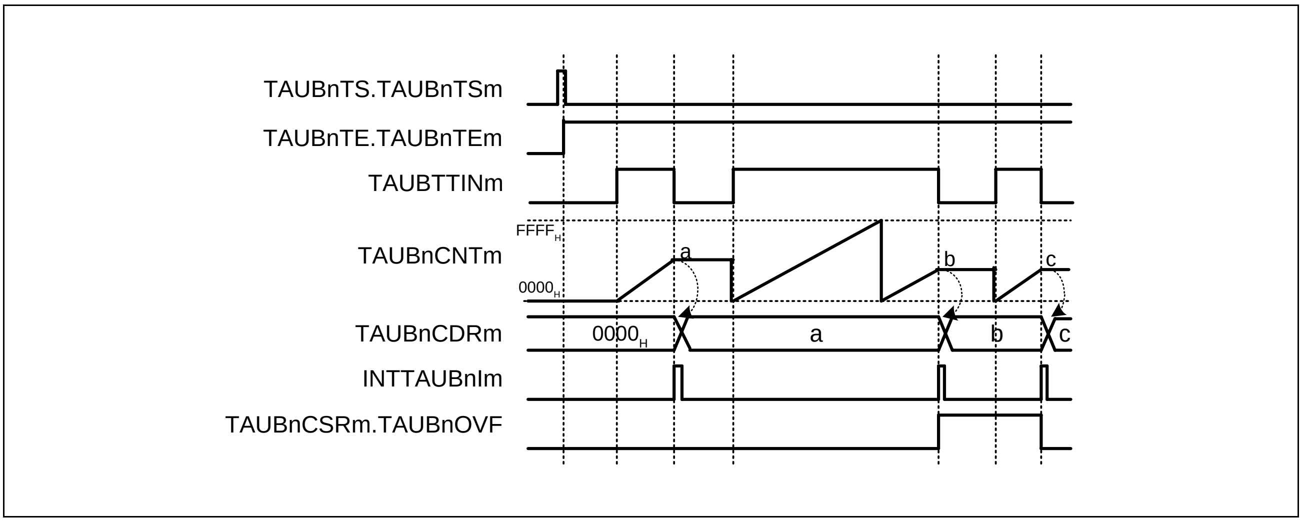


Figure 31.58 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

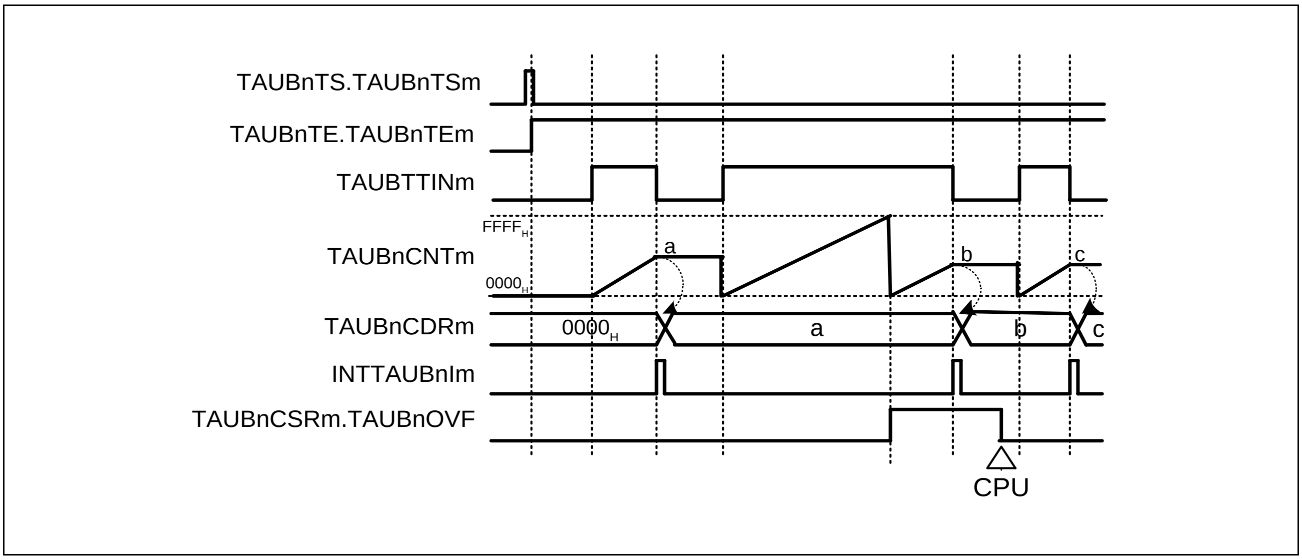


Figure 31.59 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command (The TAUBnCSCm.TAUBnCLOV bit = 1).

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

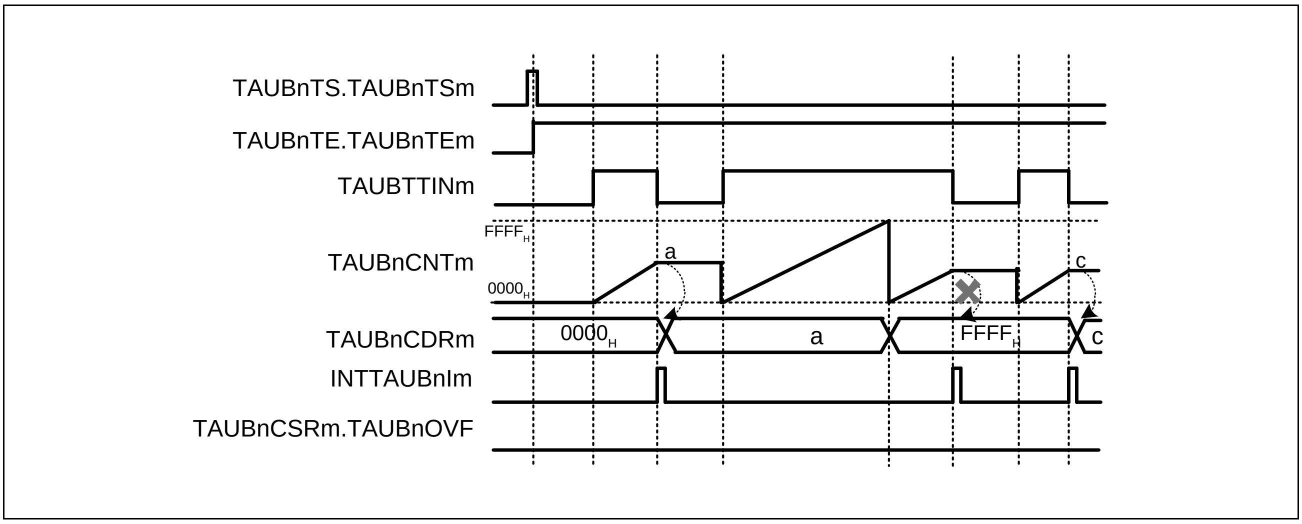


Figure 31.60 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

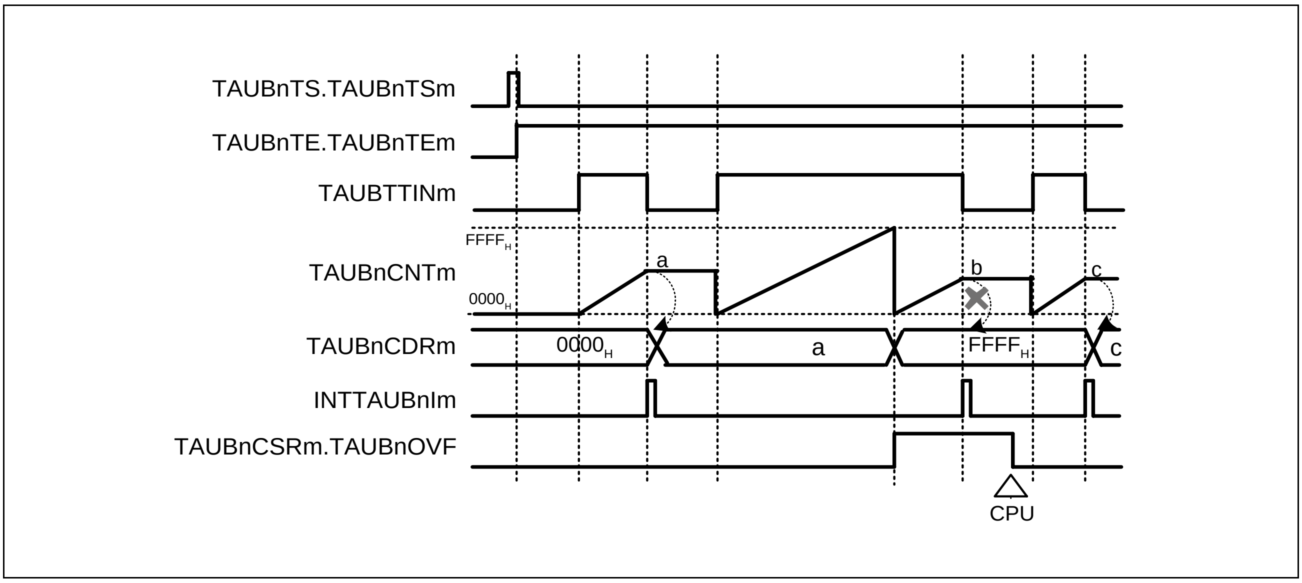


Figure 31.61 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When an overflow occurs (high width measurement)

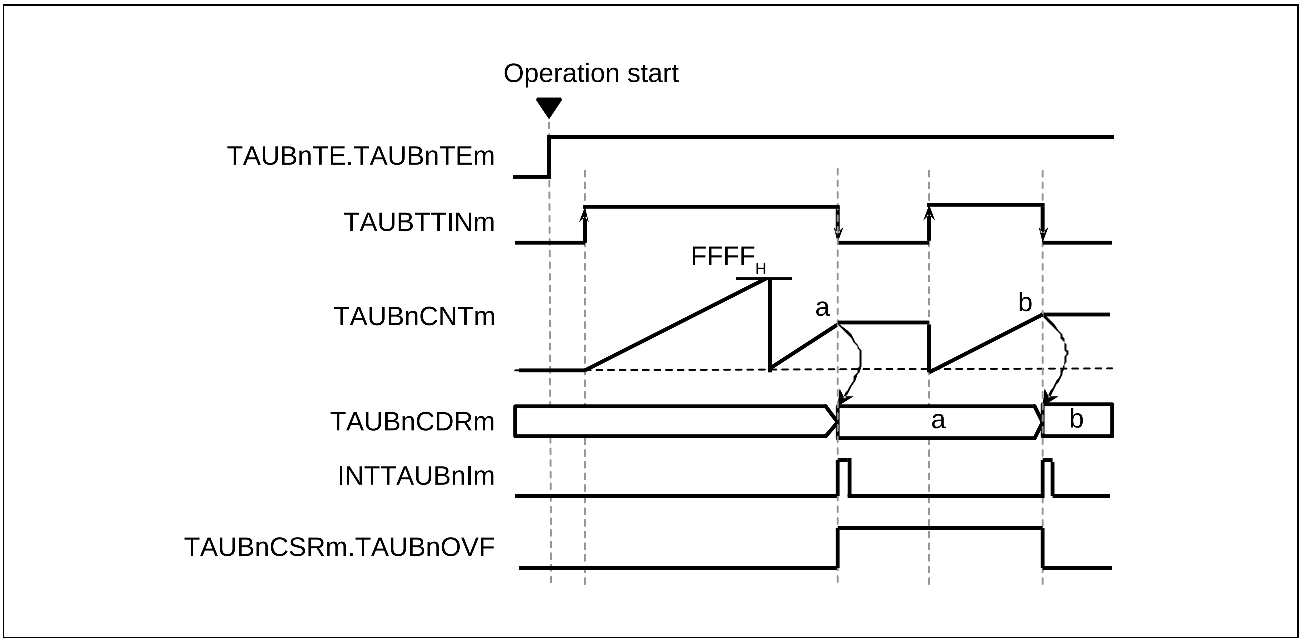


Figure 31.62 When an Overflow Occurs

When a capture trigger is input after the counter value has overflowed, the counter value is transferred to TAUBnCDRm and at the same time TAUBnCSRm.TAUBnOVF is set to 1.

TAUBnCSRm.TAUBnOVF is kept at 1 until the next capture trigger occurs.

If the next capture trigger is not accompanied by an overflow, TAUBnCSRm.TAUBnOVF is cleared to 0.

TAUBTTINm input signal width (example when TAUBnCSRm.TAUBnOVF is 1 and TAUBnCDRm is a)
= count clock cycle × ((10000_H × TAUBnCSRm.TAUBnOVF) + (TAUBnCDRm capture value + 1))
= count clock cycle × ((10000_H × 1) + (a+1))
= count clock cycle × (10000_H + a+1)

31.12.8 TAUBTTINm Input Position Detection Function

31.12.8.1 Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUBTTINm signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter starts to count from 0000_H. When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits.

Conditions

If the TAUBnCMORm.MD0 bit is set to 0, the first interrupt after a start or restart is not generated.

31.12.8.2 Equations

Function duration at a TAUBTTINm input pulse =

count clock cycle × (TAUBnCDRm capture value + 1)

31.12.8.3 Block Diagram and General Timing Diagram

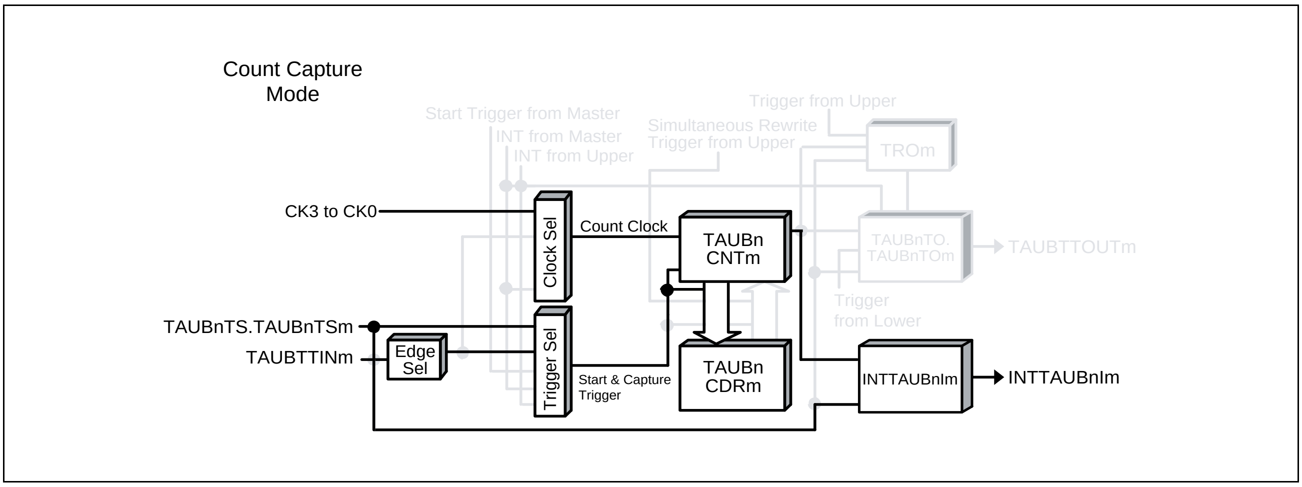


Figure 31.63 Block Diagram for TAUBTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

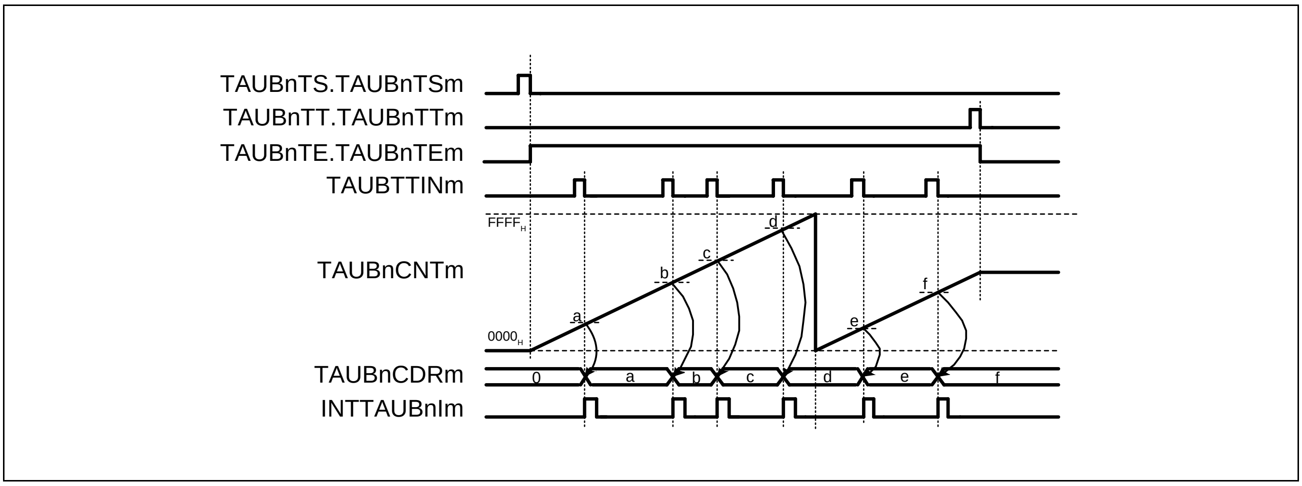


Figure 31.64 General Timing Diagram for TAUBTTINm Input Position Detection Function

31.12.8.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.86 Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1011_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.87 Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 31.88 Simultaneous Rewrite Settings for TAUBTTINm Input Position Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.8.5 Operating Procedure for TAUBTTINm Input Position Detection Function

Table 31.89 Operating Procedure for TAUBTTINm Input Position Detection Function

Restart operation

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.86, Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function** and **Table 31.87, Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: ● TAUBnCNTm transfers (captures) its value to TAUBnCDRm ● INTTAUBnIm is output. ● The counter value is not cleared to 0000_H and TAUBnCNTm continues count operation. Afterwards, this procedure is repeated. If TAUBnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.8.6 Specific Timing Diagrams

(1) Operation stop and restart

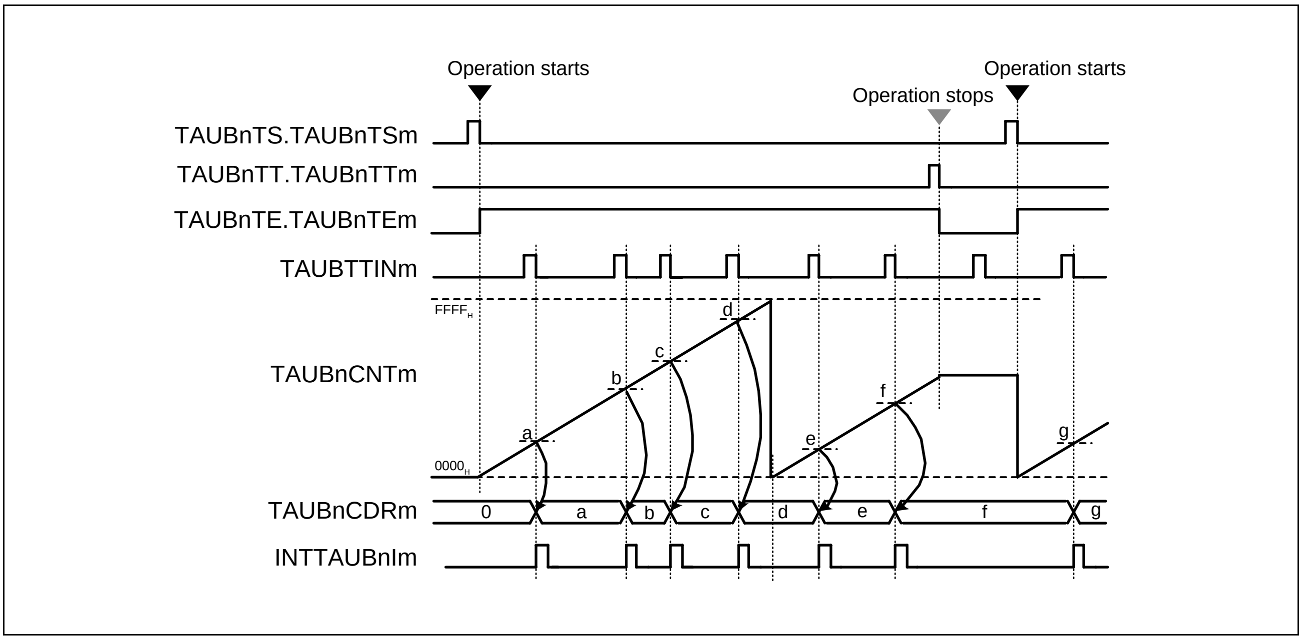


Figure 31.65 Operation Stop and Restart (TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

31.12.9 TAUBTTINm Input Period Count Detection Function

31.12.9.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter awaits a valid TAUBTTINm input edge.

When a valid TAUBTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter stops and retains its value (TAUBnCDRm + 1) until the next valid TAUBTTINm input start edge is detected.

When a next valid TAUBTTINm input start edge is detected, the counter restarts from the value retained while stopping.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits.
2. As this function is to measure the TAUBTTINm input signal width, setting TAUBnTS.TAUBnTSm to 1 is disabled while TAUBnTE.TAUBnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

31.12.9.2 Equations

Cumulative TAUBTTINm input width =

count clock cycle × (TAUBnCDRm capture value + 1)

31.12.9.3 Block Diagram and General Timing Diagram

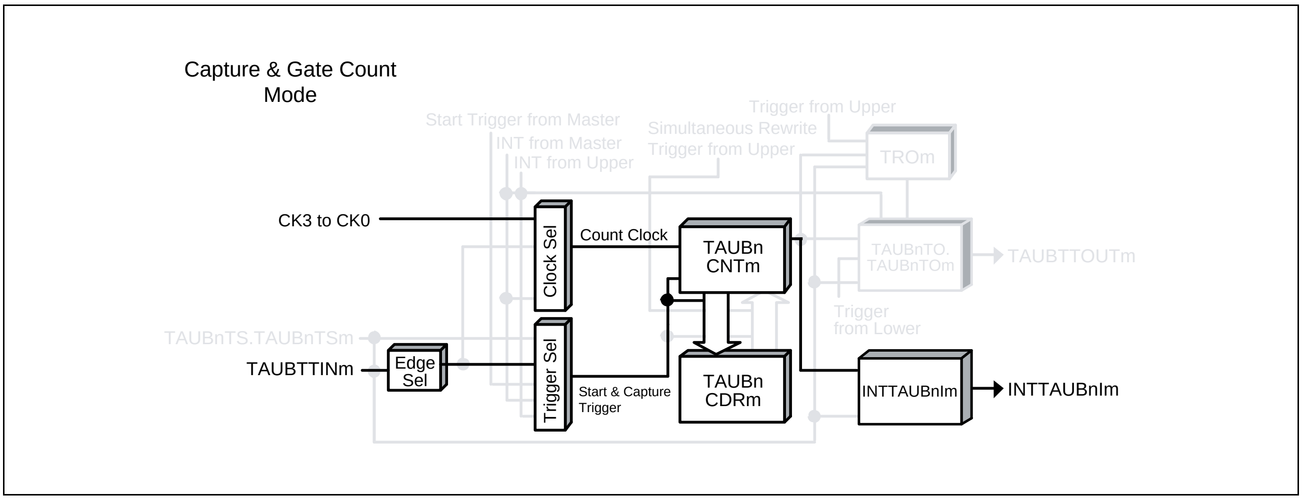


Figure 31.66 Block Diagram for TAUBTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

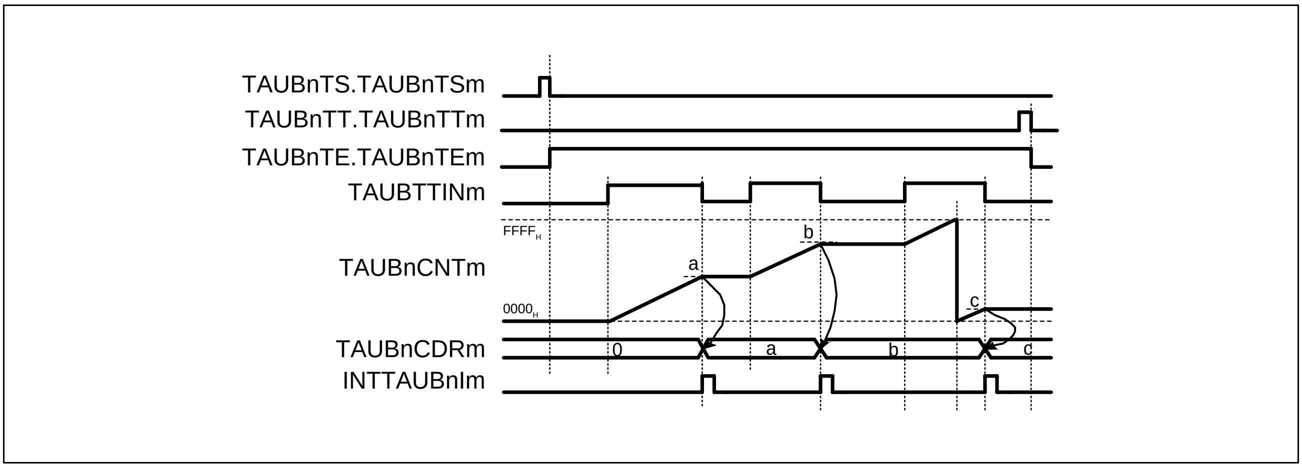


Figure 31.67 General Timing Diagram for TAUBTTINm Input Period Count Detection Function

31.12.9.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.90 Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1101_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.91 Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 31.92 Simultaneous Rewrite Settings for TAUBTTINm Input Period Count Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function

Table 31.93 Operating Procedure for TAUBTTINm Input Period Count Detection Function

Restart operation

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.90, Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function** and **Table 31.91, Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	Detection of TAUBTTINm edges. The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time.	When a TAUBTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUBnCNTm starts to count up from the stop value. When TAUBnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUBnCDRm and INTTAUBnIm is generated. Counting stops at the “value transferred to TAUBnCDRm + 1” value and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When TAUBnCNTm reaches FFFF_{H}, the counter restarts from 0000_{H}. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.9.6 Specific Timing Diagrams

31.12.9.7 Operation stop and restart

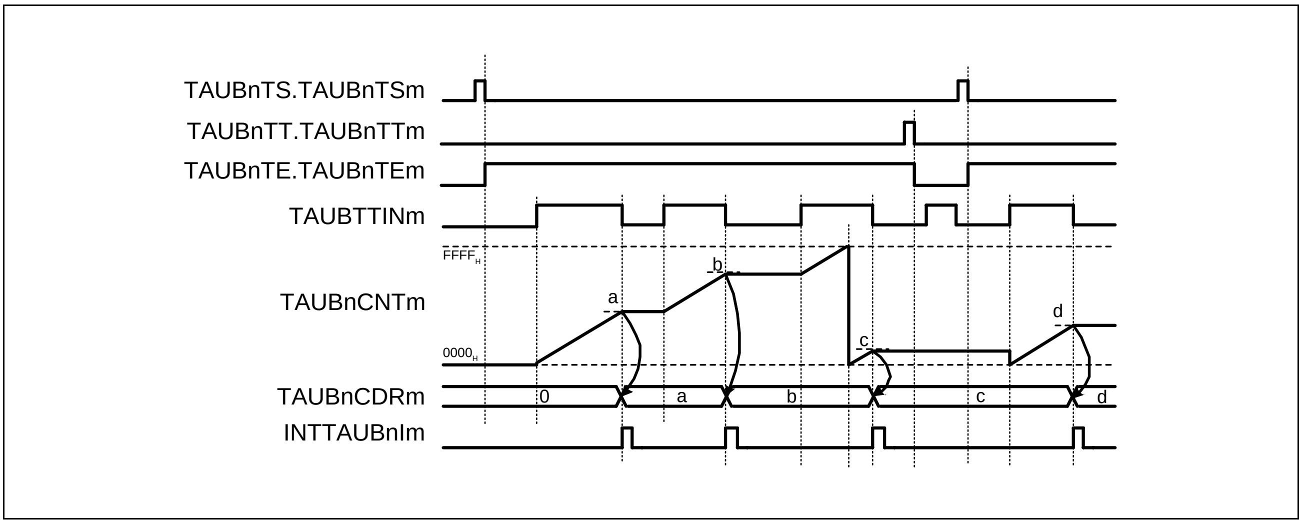


Figure 31.68 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

31.12.10 TAUBTTINm Input Pulse Interval Judgment Function

31.12.10.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUBnCNTm) and the value in the channel data register (TAUBnCDRm) when a TAUBTTINm input pulse occurs. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid edge is detected or TAUBnTS.TAUBnTSm is set to 1, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. TAUBnCNTm reloads the value of TAUBnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUBTTINm valid edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:

- If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
- If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.

31.12.10.2 Block Diagram and General Timing Diagram

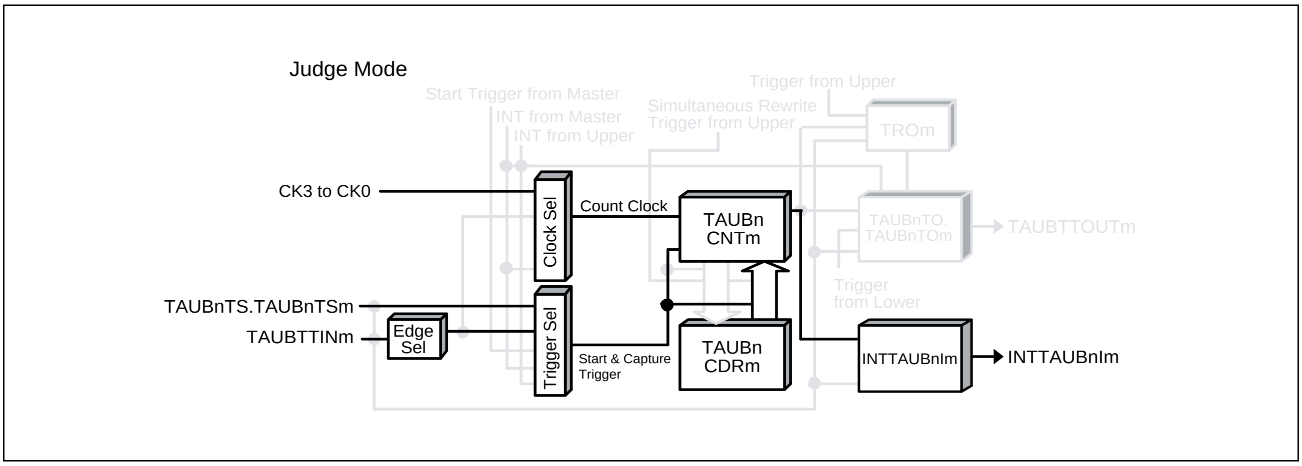


Figure 31.69 Block Diagram for TAUBTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

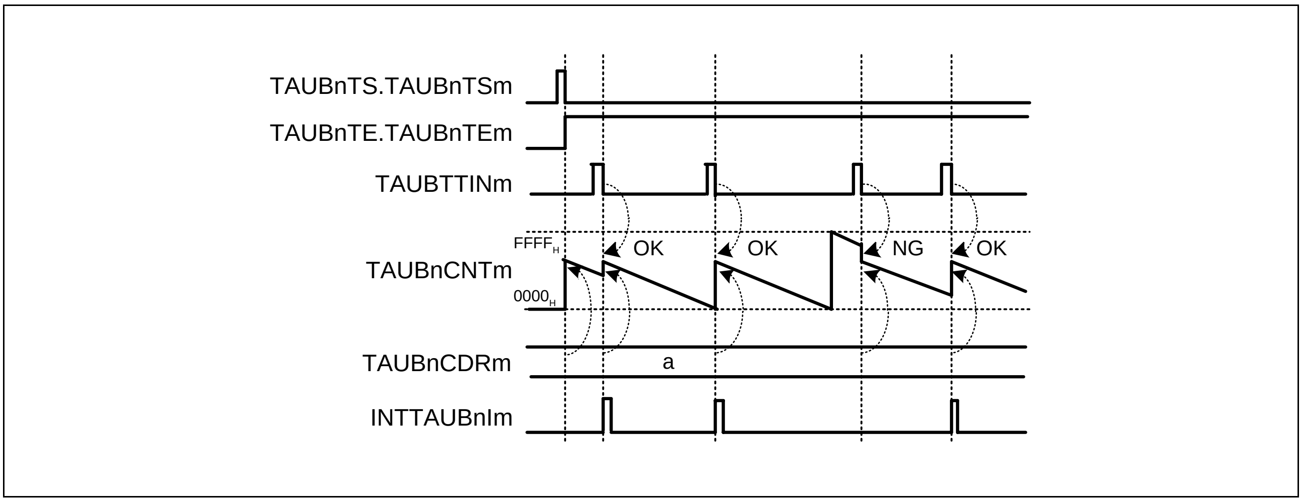


Figure 31.70 General Timing Diagram for TAUBTTINm Input Pulse Interval Judgment Function

31.12.10.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.94 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0001_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.95 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Judgment Function. Therefore, these registers must be set to 0.

Table 31.96 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), sets these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

Table 31.97 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.94, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function** and **Table 31.95, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value.
During operation	The following register can be changed at any time: TAUBnCDRm register	When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. If a TAUBTTINm input edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

31.12.11 TAUBTTINm Input Signal Width Judgment Function

31.12.11.1 Overview

Summary

This function compares the count value (TAUBnCNTm) for the high or low level width of a TAUBTTINm input signal and the TAUBnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUBnIm.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm input start edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid stop edge is detected, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. The counter TAUBnCNTm retains its value until the next TAUBTTINm valid start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUBTTINm stop edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

- The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:
  - If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
  - If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.
- The TAUBnCMURm.TAUBnTIS[1:0] bits specify the type of width measurement:
  - For high width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 11_B) the start edge is a rising TAUBTTINm edge and the stop edge is a falling TAUBTTINm edge.
  - For low width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 10_B) the start edge is a falling TAUBTTINm edge and the stop edge is a rising TAUBTTINm edge.
- Setting TAUBnTS.TAUBnTSm to 1 is prohibited during operation.

31.12.11.2 Block Diagram and General Timing Diagram

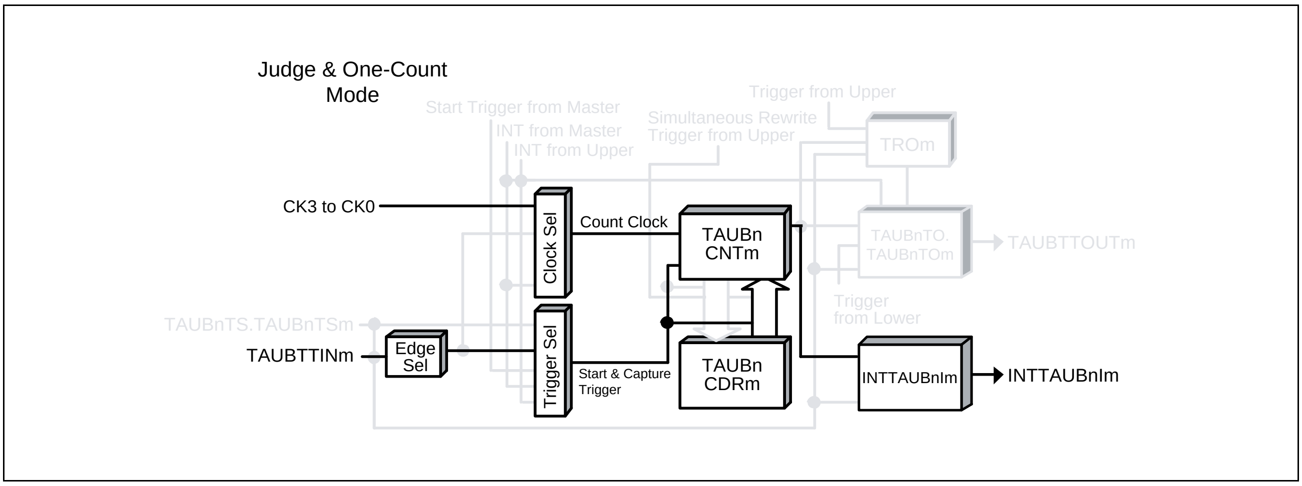


Figure 31.71 Block Diagram for TAUBTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm (TAUBnCMORm.TAUBnMD0 = 0)
- TAUBTTINm valid start edge = rising edge, TAUBTTINm valid stop edge = falling edge (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

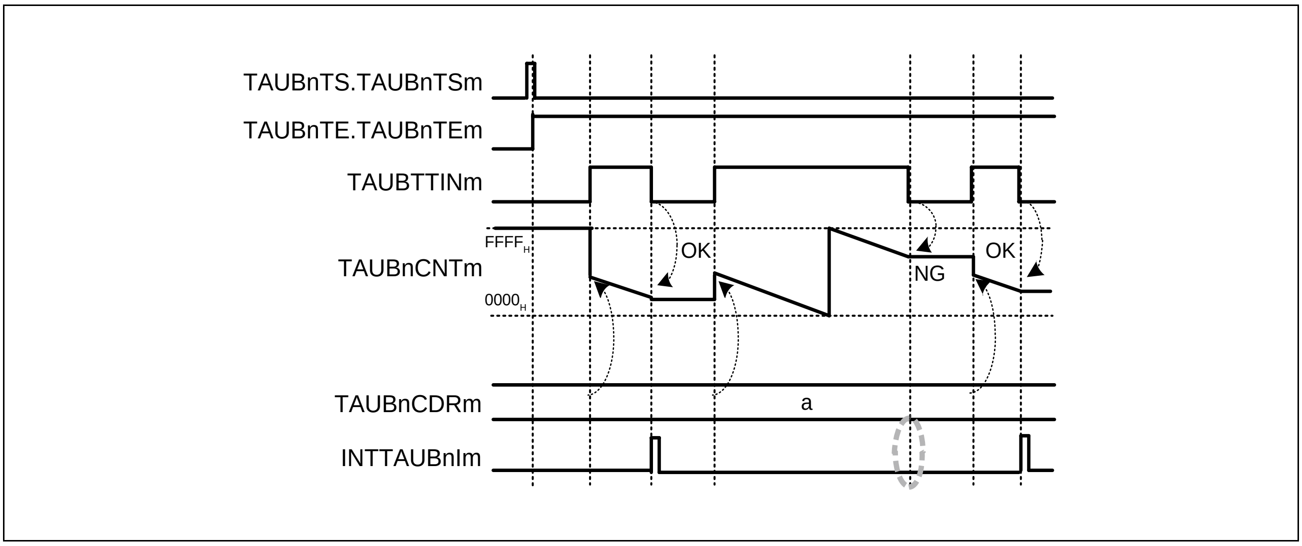


Figure 31.72 General Timing Diagram for TAUBTTINm Input Signal Width Judgment Function

31.12.11.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.98 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0111_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.99 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Judgment Function. Therefore, these registers must be set to 0.

Table 31.100 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), sets these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

Table 31.101 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

		Operation	Status of TAUBn
	Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.98, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function** and **Table 31.99, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
	During operation	The following register can be changed at any time: ● TAUBnCDRm register	If a TAUBTTINm start edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. Afterwards, this procedure is repeated.
	Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

31.12.12.1 Overview

Summary

This function measures the width of an individual TAUBTTINm input signal. An interrupt is generated if the TAUBTTINm input width is longer than $FFFF_H + 1$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is written to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUBTTINm input start edge is detected, TAUBnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

31.12.12.2 Block Diagram and General Timing Diagram

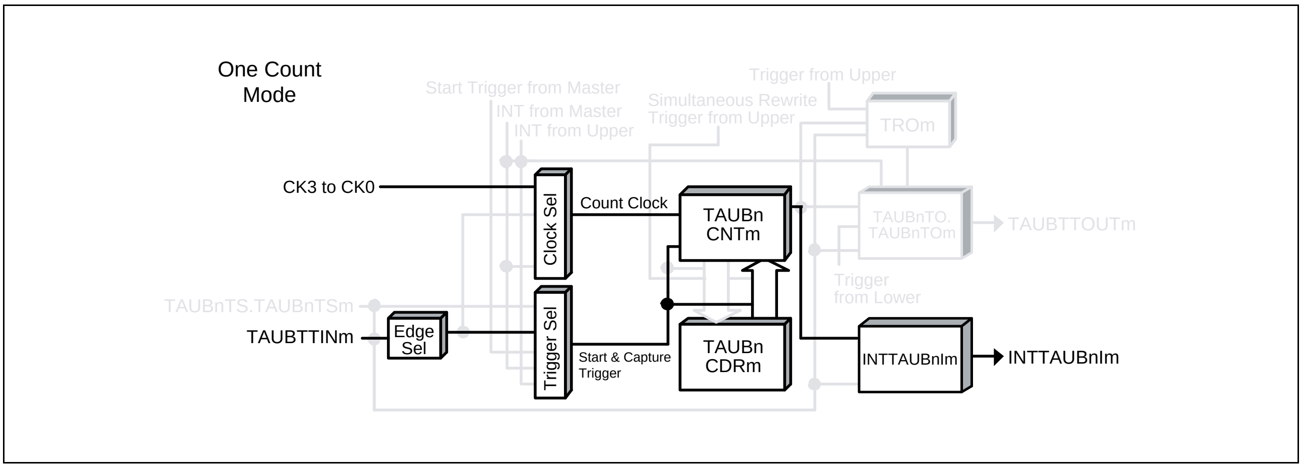


Figure 31.73 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

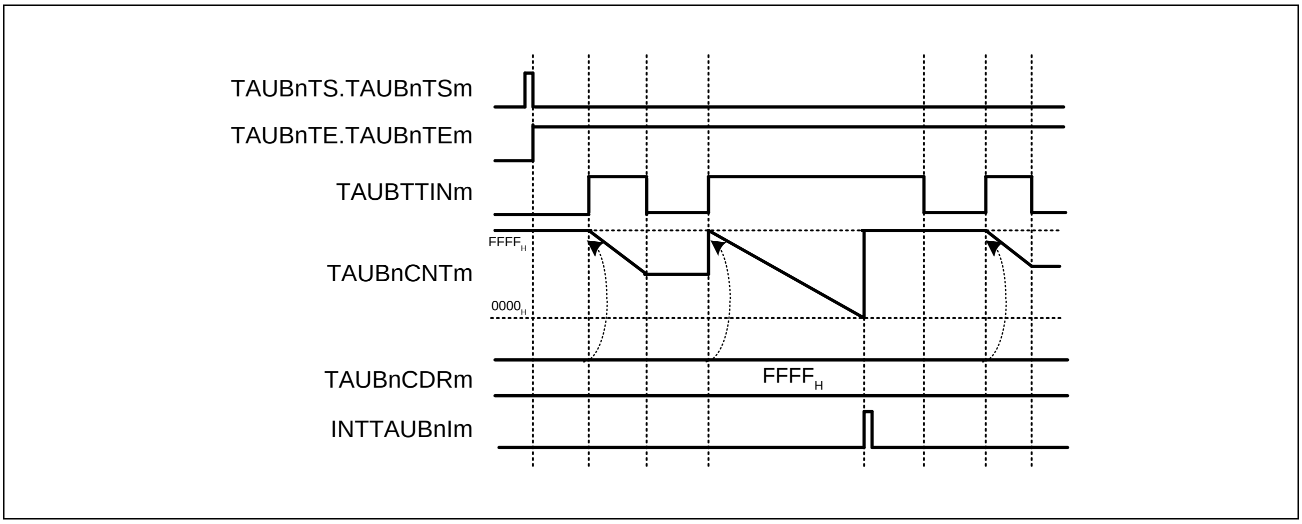


Figure 31.74 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

31.12.12.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.102 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 0_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.103 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 31.104 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Table 31.105 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

		Operation	Status of TAUBn
	Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.102, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** and **Table 31.103, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** Set the value of the TAUBnCDRm register to FFFF$_H$.	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value (FFFF$_H$).
	During operation	The TAUBnCNTm register can be read at any time.	TAUBnCNTm counts down. When the counter reaches 0000$_H$: INTTAUBnIm is generated When a TAUBTTINm input stop edge is detected during count operation: TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: The TAUBnCDRm value (FFFF$_H$) is loaded to TAUBnCNTm again and the counter starts to count down. Afterwards, this procedure is repeated.
	Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

31.12.13.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal. An interrupt is generated and an overflow interrupt can be output if the cumulative TAUBTTINm input width is longer than $FFFF_H$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is written to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUBTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is written to TAUBnCNTm and the counter continues to count down until a TAUBTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

31.12.13.2 Block Diagram and General Timing Diagram

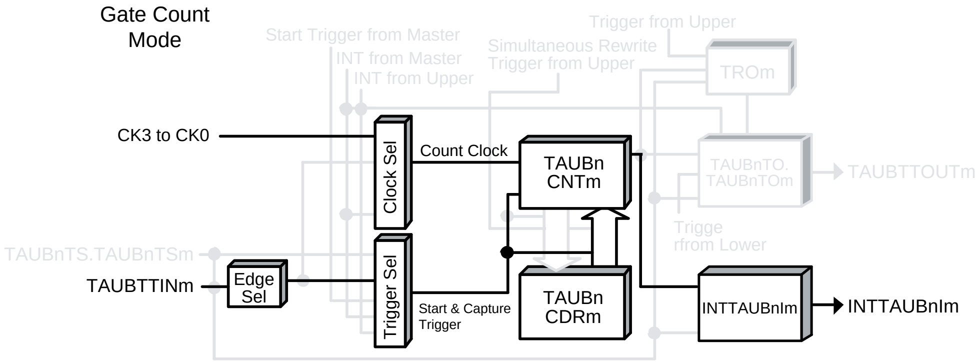


Figure 31.75 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

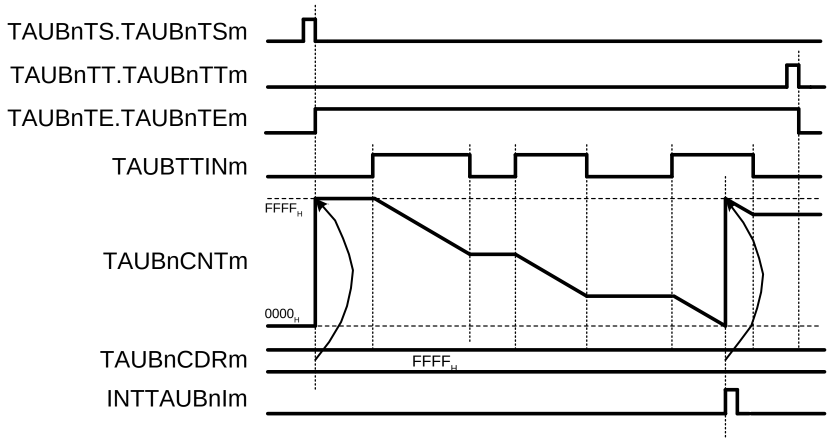


Figure 31.76 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

31.12.13.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.106 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.107 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 31.108 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

31.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Table 31.109 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Restart operation

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 31.106, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)** and **Table 31.107, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)** Set the value of the TAUBnCDRm register to FFFF_H.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1 TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1. TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value (FFFF_H).
During operation	The TAUBnCNTm register can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: INTTAUBnIm is generated TAUBnCNTm reloads the TAUBnCDRm value (FFFF_H) and continues count operation When a TAUBTTINm input stop edge is detected during count operation: TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: TAUBnCNTm starts to count down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.13 Independent Channel Simultaneous Rewrite Functions

The following describes functions that carry out simultaneous rewrite:

31.13.1 Simultaneous Rewrite Trigger Generation Function Type 1

31.13.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals. The upper channel is for generating the simultaneous rewrite trigger (TAUBnRDC.TAUBnRDCm = 1), and the lower channels are for conducting simultaneous rewrite when triggered from the upper channel (TAUBnRDC.TAUBnRDCm = 0).

Prerequisites

- Two (or more) channels that are lower than the channel used as the upper channel, each with simultaneous rewrite enabled (TAUBnRDE.TAUBnRDEm = 1)
- The operation mode of the upper channel must be set to interval timer mode, see **Table 31.110, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1.**
- For the operation modes that can be set to the lower channels, see **Table 31.53, Channel Functions and the Methods They Use for Simultaneous Rewrite.**
- In this function, TAUBTTOUTm is not used for all the channels.

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) of the upper and lower channel(s) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of the data register buffer of the upper channel (TAUBnCDRm buf) is written to the counter (TAUBnCNTm) and the counter starts to count down from this value.
The counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

When a counter reaches 0000_H, an interrupt is generated from the channel.
The corresponding TAUBnCNTm then reloads the current TAUBnCDRm buffer value and subsequently continues operation.

If the channel where the interrupt occurs is specified as the trigger channel for simultaneous rewrite (TAUBnRDC.TAUBnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUBnRSF.TAUBnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Condition

- The channel which is monitored for INTTAUBnIm is specified by setting TAUBnRDC.TAUBnRDCm = 1 for the corresponding channel. The TAUBnRDC.TAUBnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.

31.13.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUBnCDRm + 1)

To control simultaneous rewrite, the following condition must be satisfied:

[For PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[For triangle PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUBnCDRm + 1 and value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that the cycle for the triangle PWM is twice the cycle for the PWM

31.13.1.3 Block Diagram and General Timing Diagram

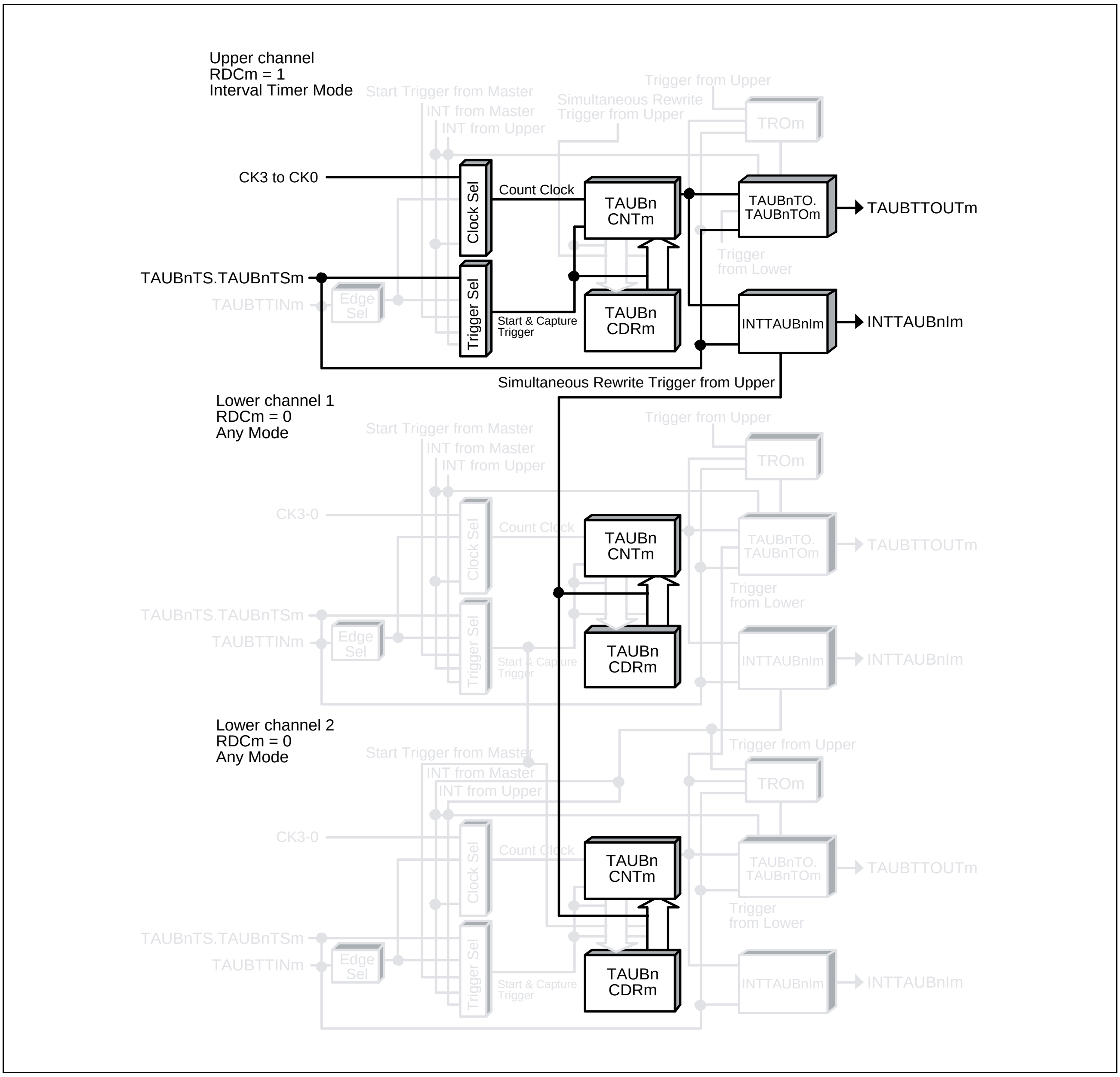


Figure 31.77 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

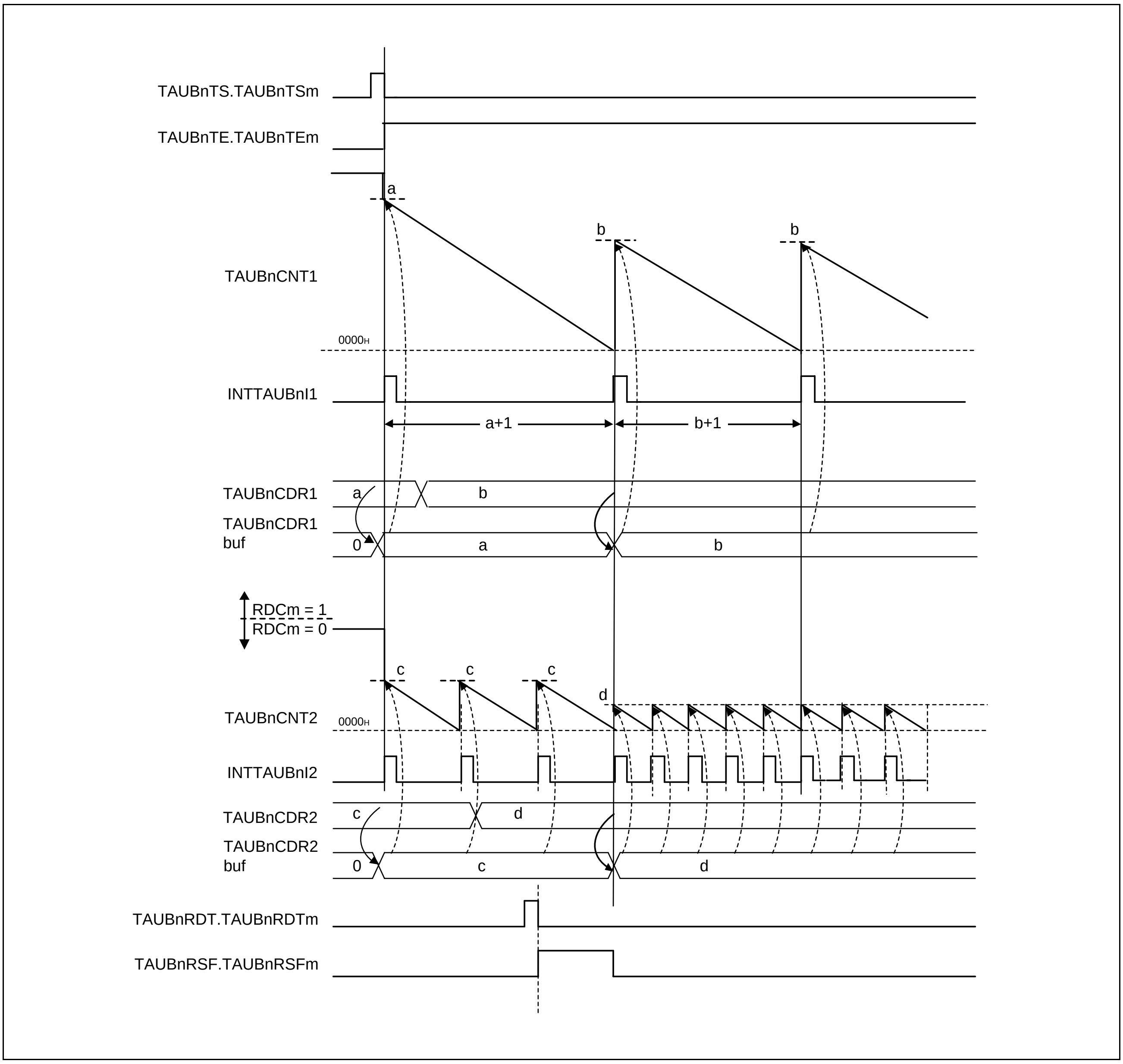


Figure 31.78 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

31.13.1.4 Register Settings for The Upper Channel

(1) TAUBnCMORm for the upper channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.110 Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the upper channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.111 Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the upper channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function. However, it can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite for the upper channel

Table 31.112 Simultaneous Rewrite Settings for the Upper Channel in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	1: Channel is monitored for an INTTAUBnIm signal that is used as the simultaneous rewrite trigger

31.13.1.5 Register Settings for the Lower Channel(s)

(1) TAUBnCMORm for the lower channel(s)

For the TAUBnCMORm register of the lower channels, follow the TAUBnCMORm register settings for the operation mode that can be set. (See **Table 31.53, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

(2) TAUBnCMURm for the lower channel(s)

For the TAUBnCMURm register of the lower channels, follow the TAUBnCMURm register settings for the operation mode that can be set. (See **Table 31.53, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

(3) Channel output mode for the lower channel(s)

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 31.53, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

(4) Simultaneous rewrite for the lower channel(s)

Table 31.113 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 31.114 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Restart operation

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers for the upper channel as described in **Table 31.110, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** and **Table 31.111, Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** Set the TAUBnCMORm and TAUBnCMURm registers for the lower channel as described in **Section 31.13.1.5, Register Settings for the Lower Channel(s)** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	TAUBnRDT.TAUBnRDTm, TAUBnCDRm.TAUBnCDR can be changed. TAUBnRSF.TAUBnRSFm can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: TAUBnCNTm reloads the TAUBnCDRm value and continues count operation INTTAUBnIm is generated Simultaneous rewrite is controlled when INTTAUBnIm is generated from the channel where TAUBnRDC.TAUBnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

31.14 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the Timer Array Unit B. For a general overview of synchronous channel operation, see Overview.

31.14.1 PWM Output Function

31.14.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the pulse width (duration) of the TAUBTTOUTm to be set. The pulse cycle is set in the master channel. The pulse width is set in the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 31.115, Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function**.
- The operation mode of the slave channel(s) must be set to one-count mode, see **Table 31.118, Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function**.
- TAUBTTOUTm is not used for the master channel of this function.
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode 1.

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) toggles, which realizes a PWM output.

- Master channel:

When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter loads the TAUBnCDRm value and counts down.

- Slave channel:

INTTAUBnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value. The TAUBTTOUTm signal is set to the active level.

When the counter reaches 0000_H, i.e. duty time has elapsed, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset to the inactive level. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Set Simultaneous rewrite with this function. Please see **Section 31.6, Simultaneous Rewrite**.

31.14.1.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUBnCDRm (slave) / (TAUBnCDRm (master) + 1)) × 100

- Duty cycle = 0%
  TAUBnCDRm (slave) = 0000_H
- Duty cycle = 100%
  TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

31.14.1.3 Block Diagram and General Timing Diagram

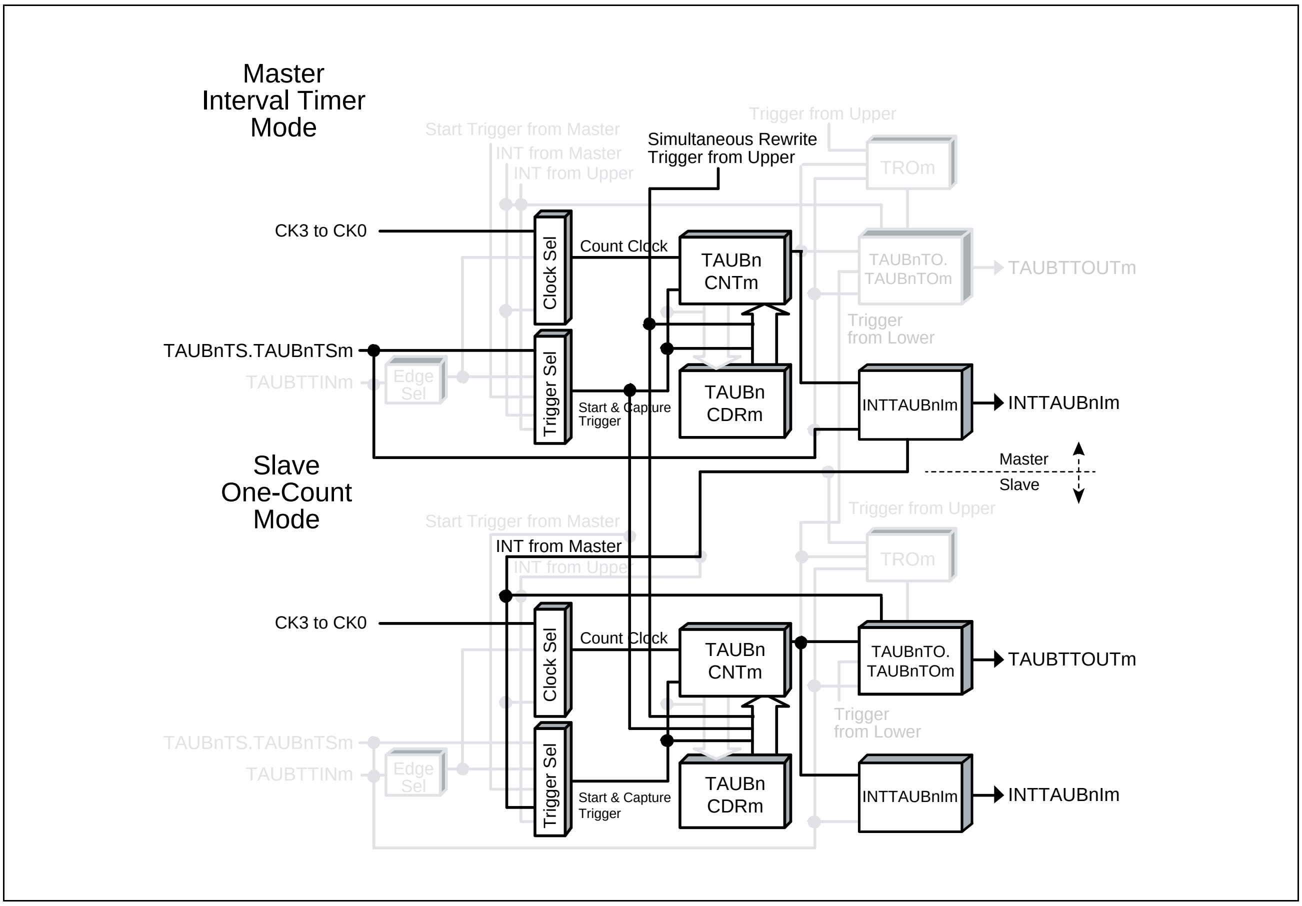


Figure 31.79 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

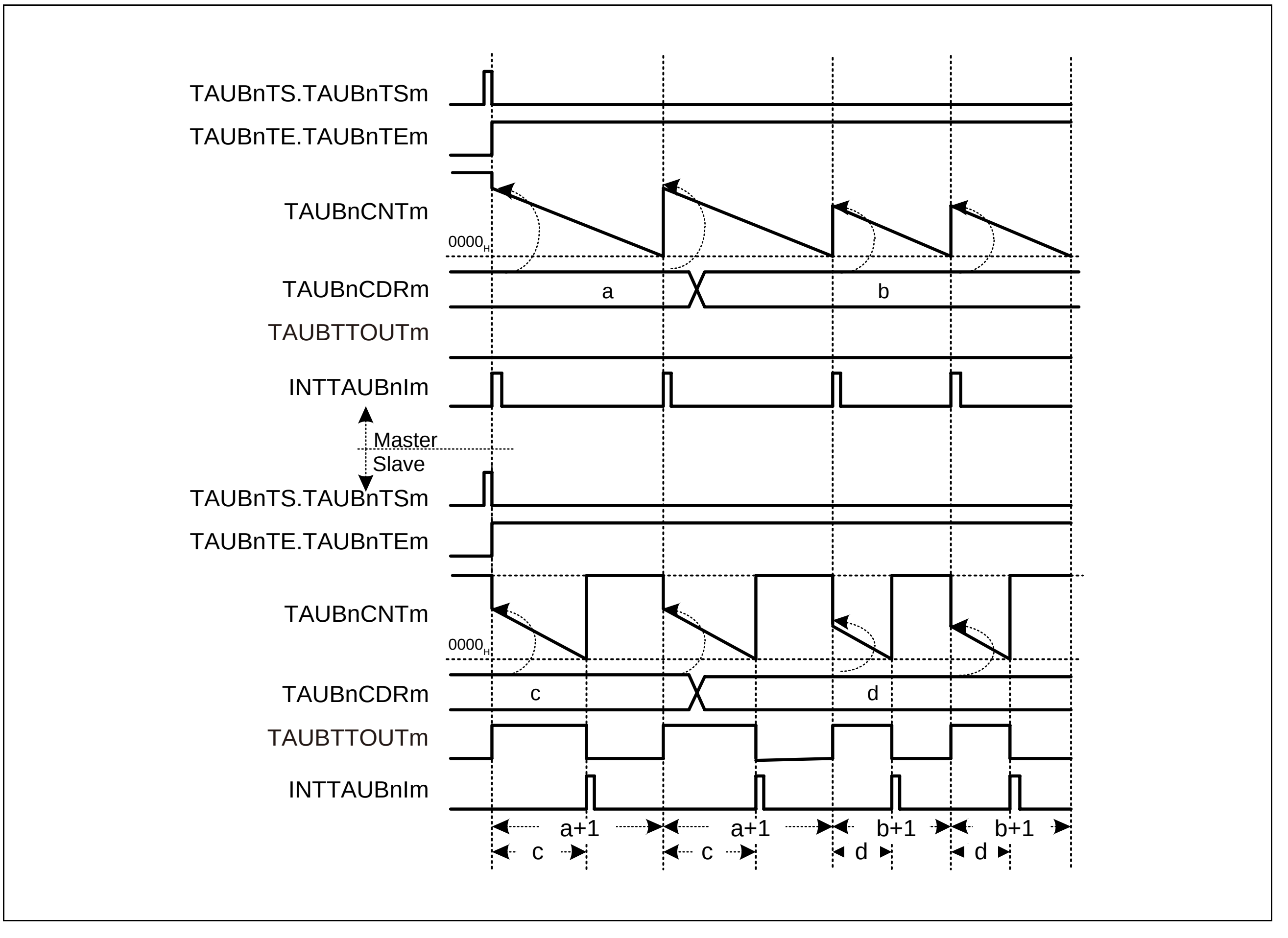


Figure 31.80 General Timing Diagram for PWM Output Function

NOTE

- The interval between the start of the count and an interrupt being generated is the value of corresponding TAUBnCDRm + 1.
- TAUBTTOUTm of the slave channel will rise with a delay of one count clock after the rising of INTTAUBnIm of the master channel.

31.14.1.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.115 Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.116 Contents of the TAUBnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.117 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

When used in TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel operating in **Section 31.13.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Configure the operation following the conditions below.

- The channel set to Simultaneous Rewrite Trigger Output Function Type 1: TAUBnRDCm = 1, TAUBnRDSm = 1
  The setting value of TAUBnCDRm to this channel is as follows.
  = ((setting value of TAUBnCDRm of the master channel subject to simultaneous rewrite + 1) × number of interrupts) – 1
- Master channel: TAUBnRDCm = 0, TAUBnRDSm = 1
- Slave channel: TAUBnRDCm = 0, TAUBnRDSm = 1

Although the value of duty exceeds 100% when the setting value of TAUBnCDRm (slave) > the setting value of TAUBnCDRm (master) + 1, the output will be aggregated to 100%.

31.14.1.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.118 Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.119 Contents of the TAUBnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 31.120 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.121 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.1.6 Operating Procedure for PWM Output Function

Table 31.122 Operating Procedure for PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.1.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel loads TAUBnCDRm and counts down. When the counter reaches 0000_H: ● INTTAUBnIm (master) is generated ● TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation ● TAUBnCNTm (slave) loads the TAUBnCDRm value and counts down ● TAUBTTOUTm (slave) is set to the active level When TAUBnCNTm (slave) reaches 0000_H: ● The counter of TAUBnCNTm (slave) stops. ● INTTAUBnIm (slave) is generated ● TAUBTTOUTm (slave) is set to the inactive level
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

31.14.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

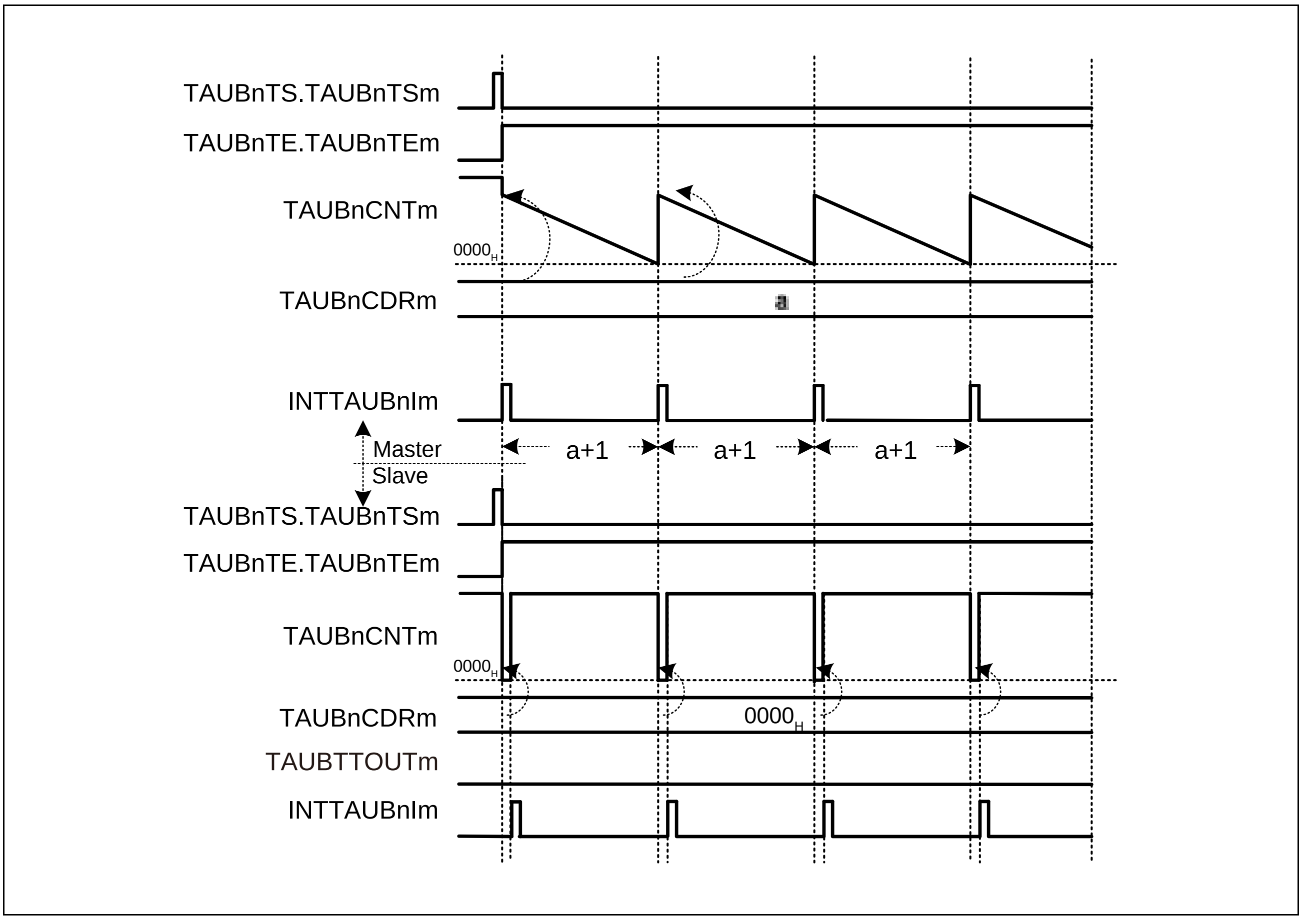


Figure 31.81 TAUBnCDRm (slave) = 0000_H,
Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUBnIm), 0000_H is written to TAUBnCNTm (slave). As a result, a slave channel interrupt (INTTAUBnIm) is generated at the same time and TAUBTTOUTm remains inactive.
- TAUBnCNTm (slave) generates an interrupt every time the value of TAUBnCDRm is loaded.

(2) Duty cycle = 100%

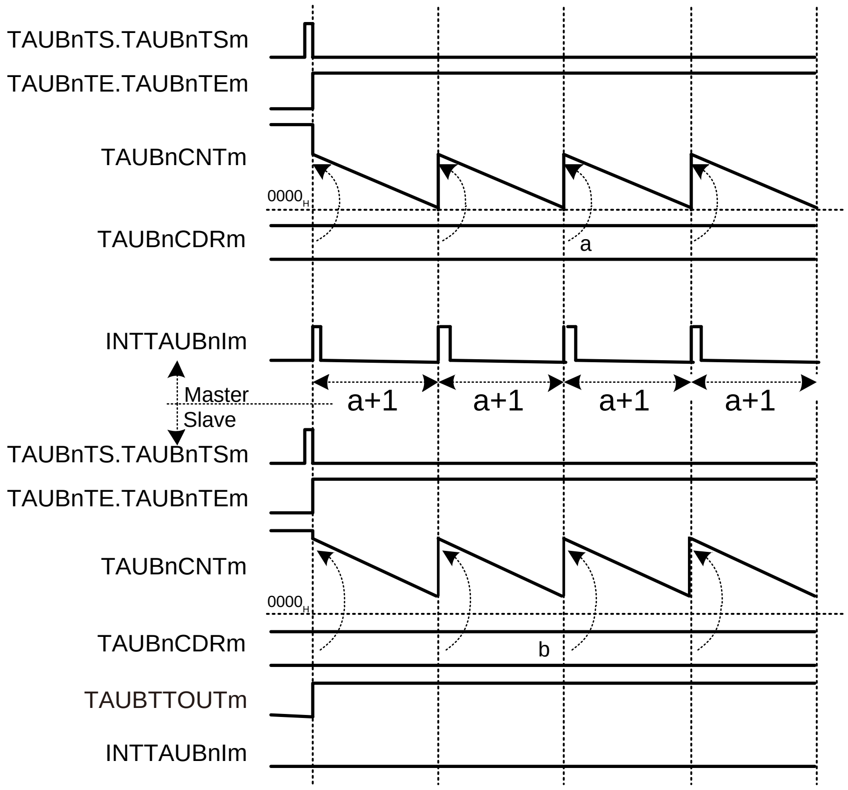


Figure 31.82 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1,
Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

If the value TAUBnCDRm (slave) is higher than the value TAUBnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUBTTOUTm remains at active state.

(3) Operation stop and restart

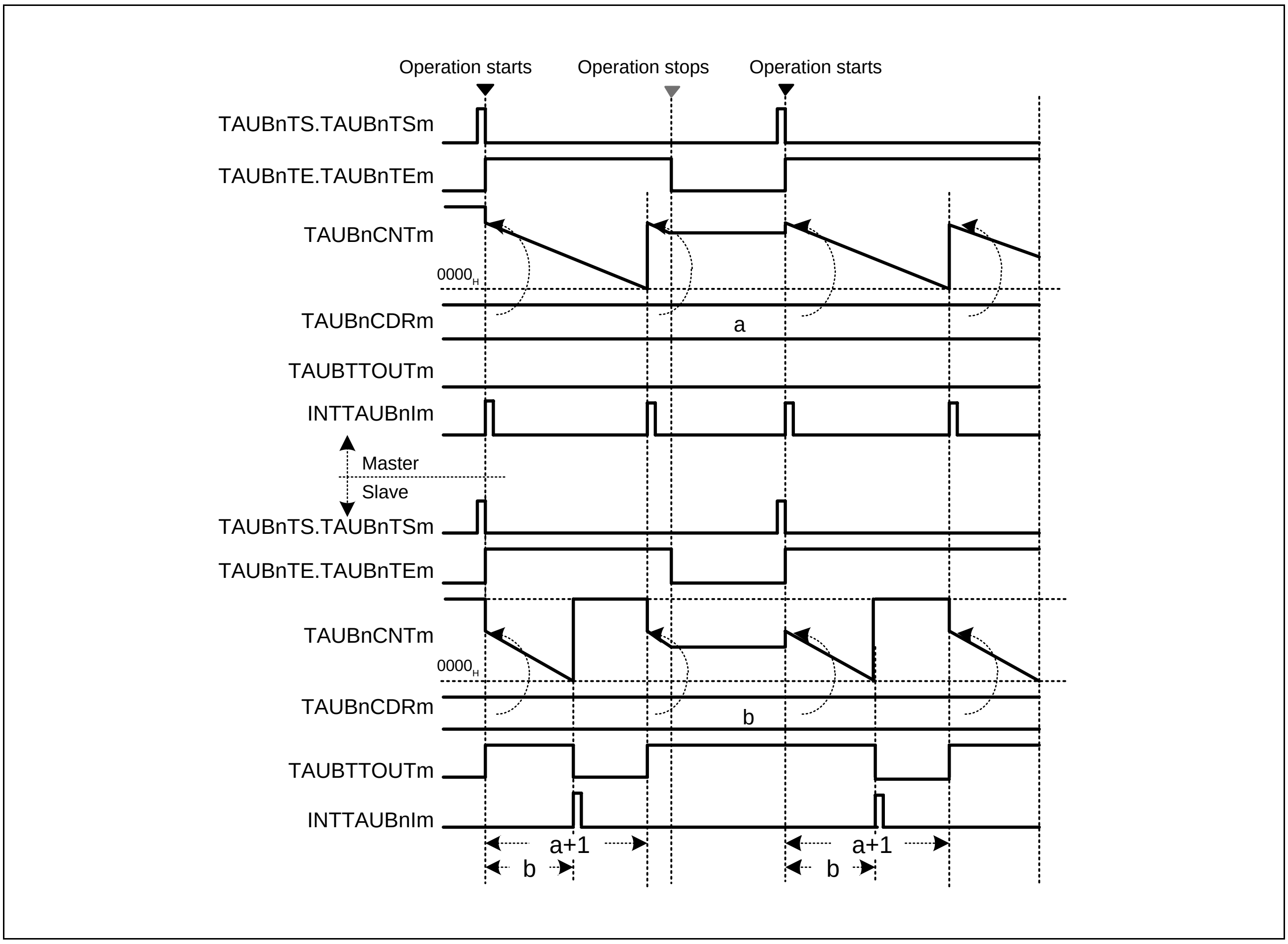


Figure 31.83 Operation Stop and Restart, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm of the master and slave channel(s) to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm of master and slave channel(s) to 1. TAUBnCNTm of master and slave channel reload the current values of TAUBnCDRm and start to count down from these values.

(4) Operation stop and restart (Slave output, Initialization)

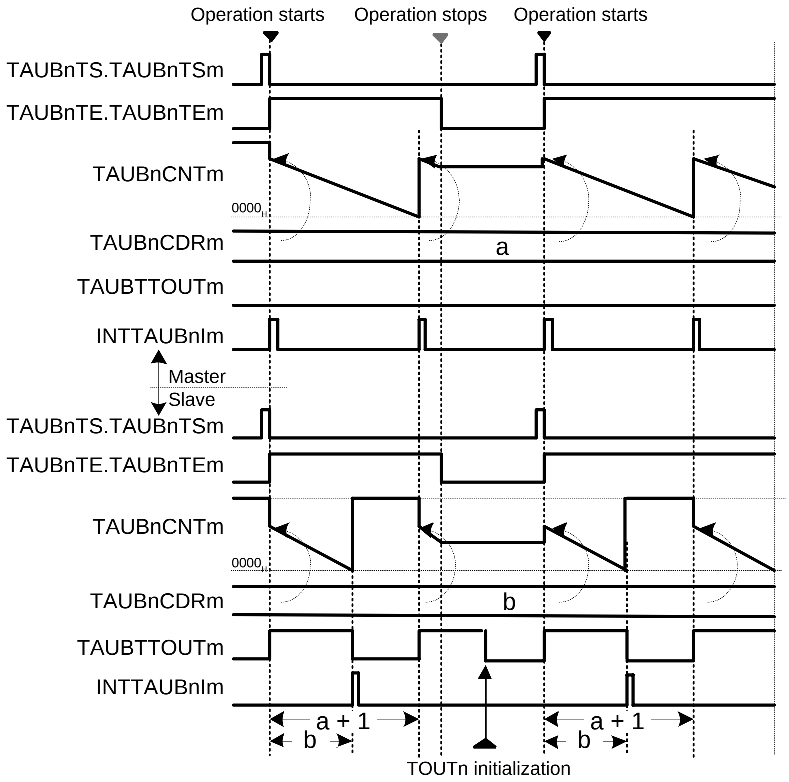


Figure 31.84 Operation Stop and Restart (Slave Output, Initialization)

When TAUBnTOE.TAUBnTOEm of the slave channel is set to 0 while TAUBnTE.TAUBnTEm = 0 and the inactive level of TAUBTTOUTm is written in the TAUBnTO.TAUBnTOm, the output level of TAUBTTOUTm (slave channel) becomes active when INTTAUBnIm is issued when the count operation is started after restart.

31.14.2 One-Shot Pulse Output Function

31.14.2.1 Overview

Summary

This function outputs a signal pulse with a defined pulse width and a specific delay time compared to an external input signal pulse by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to one-count mode, see **Table 31.123, Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**
- The operation mode of the slave channel must be set to pulse one-count mode, see **Table 31.126, Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel must be set to independent channel output mode 2.
- TAUBTTINm (master) has to be detected while TAUBnCNTm (master) and TAUBnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUBTTINm (slave).

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) for master and slave channels to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

- Master channel:
  When the next valid TAUBTTINm input edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm. The counter starts to count down from this value. If TAUBnCMORm.TAUBnMD0 = 0, a trigger (TAUBTTINm) which is detected within the delay time is ignored.
  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUBTTINm input edge.
- Slave channel:
  INTTAUBnIm generated on the master channel triggers the counter of the slave channel. The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value.
  An interrupt is generated and the TAUBTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter remains at 0000_H and awaits the next INTTAUBnIm of the master channel.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

The counter of the master channel can be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

NOTES

1. If a forced restart of the counter is executed during operation, the width of the output signal does not correspond to the value of TAUBnCDRm (slave).
2. The TAUBTTINm input signal is sampled at the frequency of the operating clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

Conditions

- If TAUBnCMORm.TAUBnMD0 of the master channel is set to 0, during counting detected TAUBTTINm input edges are ignored.
- Simultaneous rewrite can be used with this function. Please see **Section 31.6, Simultaneous Rewrite**.

31.14.2.2 Equations

Delay from trigger input to pulse output

= (TAUBnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUBnCDRm (slave)) × count clock cycle

31.14.2.3 Block Diagram and General Timing Diagram

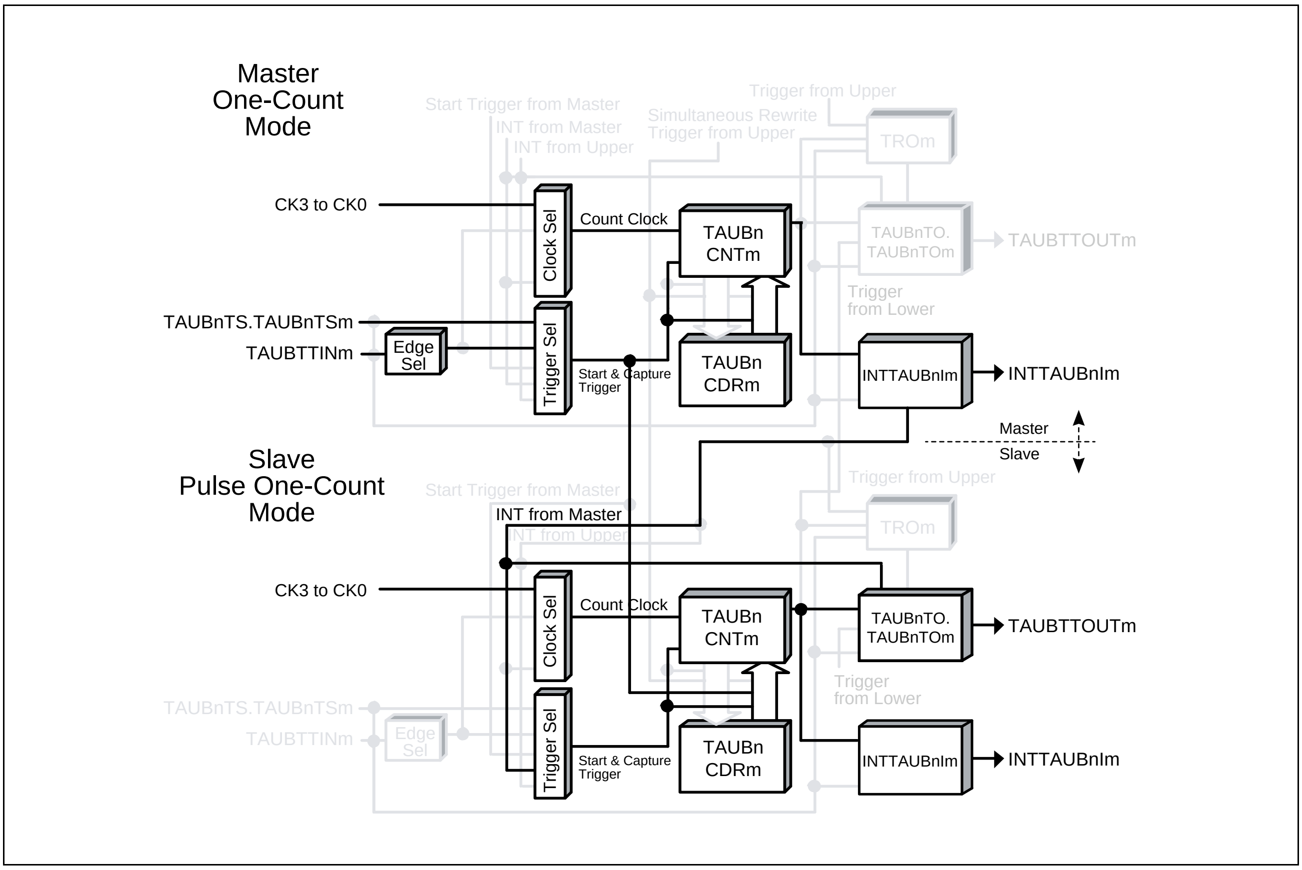


Figure 31.85 Block Diagram for One-Shot Pulse Output Function

The following settings apply to the general basic diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

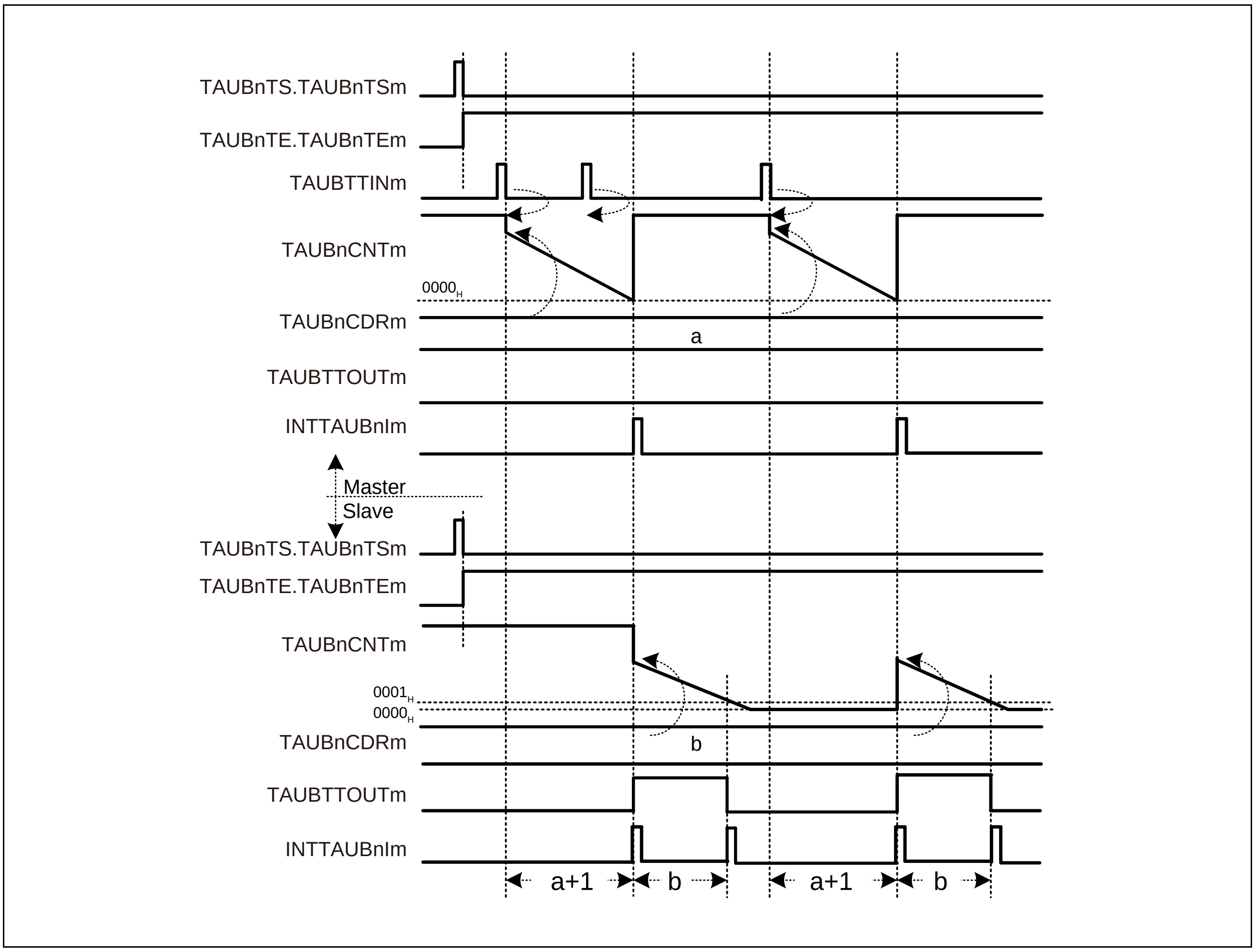


Figure 31.86 General Timing Diagram for One-Shot Pulse Output Function

31.14.2.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.123 Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_{B}.
11	TAUBnMAS	Write 1_{B}.
10 to 8	TAUBnSTS[2:0]	Write 001_{B}.
7, 6	TAUBnCOS[1:0]	Write 00_{B}.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_{B}.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.124 Contents of the TAUBnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.125 Simultaneous Rewrite Settings for the Master Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.2.5 Register Settings for the Slave Channel

(1) TAUBnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.126 Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the MD0 bit of the master and slave channels must be identical.

(2) TAUBnCMURm for the slave channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.127 Contents of the TAUBnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel

Table 31.128 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	1: Enables independent channel output mode
TAUBnTOM.TAUBnTOMm	0: Independent channel output
TAUBnTOC.TAUBnTOCm	1: Operating mode 2
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	0: Disables dead time operation
TAUBnTDL.TAUBnTDLm	0: When dead time operation is disabled (TAUBnTDE.TAUBnTDEm = 0), set these bits to 0

(4) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.129 Simultaneous Rewrite Settings for the Slave Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 31.130 Operating Procedure for One-Shot Pulse Output Function

		Operation	Status of TAUBn
	Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.2.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.2.5, Register Settings for the Slave Channel**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUBTTINm input.
	During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	When a valid TAUBTTINm input edge is detected, TAUBnCNTm of the master channel loads the TAUBnCDRm value and counts down. When the counter reaches 0000_H: ● INTTAUBnIm (master) is generated ● TAUBnCNTm (master) is reset to $FFFF_H$ and waits for the next valid TAUBTTINm input edge. ● TAUBnCNTm (slave) reloads the TAUBnCDRm value and starts to count down ● INTTAUBnIm (slave) is generated ● TAUBTTOUTm (slave) is set to the active level. When TAUBnCNTm (slave) reaches 0001_H: ● The counter of TAUBnCNTm (slave) stops. ● INTTAUBnIm (slave) is generated ● TAUBTTOUTm (slave) is set to an inactive level.
	Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

31.14.2.7 Specific Timing Diagrams

(1) TAUBnCDRm (master) = 0000$_H$

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00$_B$)

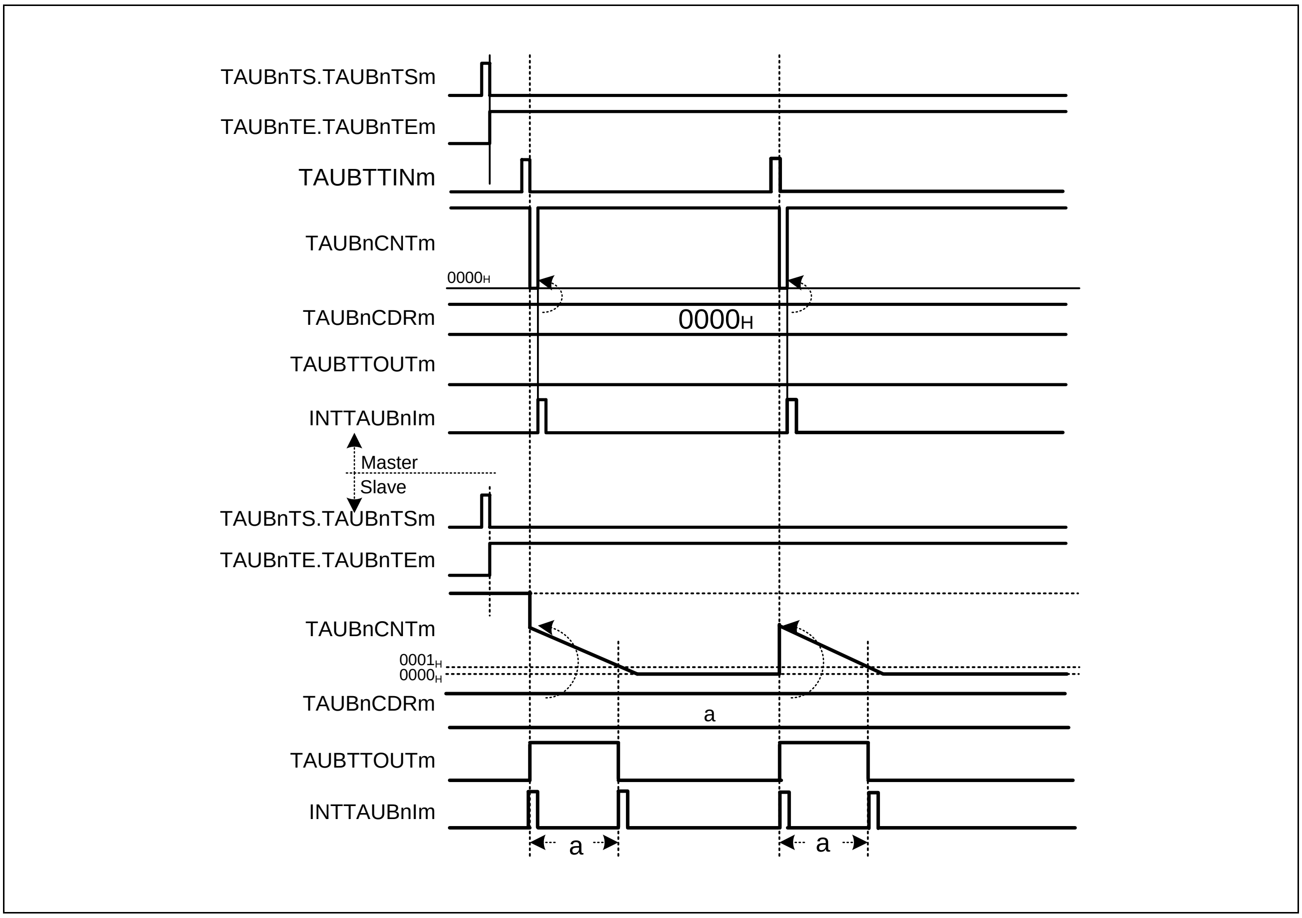


Figure 31.87 TAUBnCDRm (Master) = 0000$_H$

- When a valid TAUBTTINm input edge is detected, the value 0000$_H$ is written to TAUBnCNTm (master). The counter is set to 0000$_H$ for one count and returns to FFFF$_H$.
  Thus, the slave channel starts to count down one count clock later to TAUBTTINm (master).

(2) TAUBnCDRm (slave) = 0000$_H$

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00$_B$)

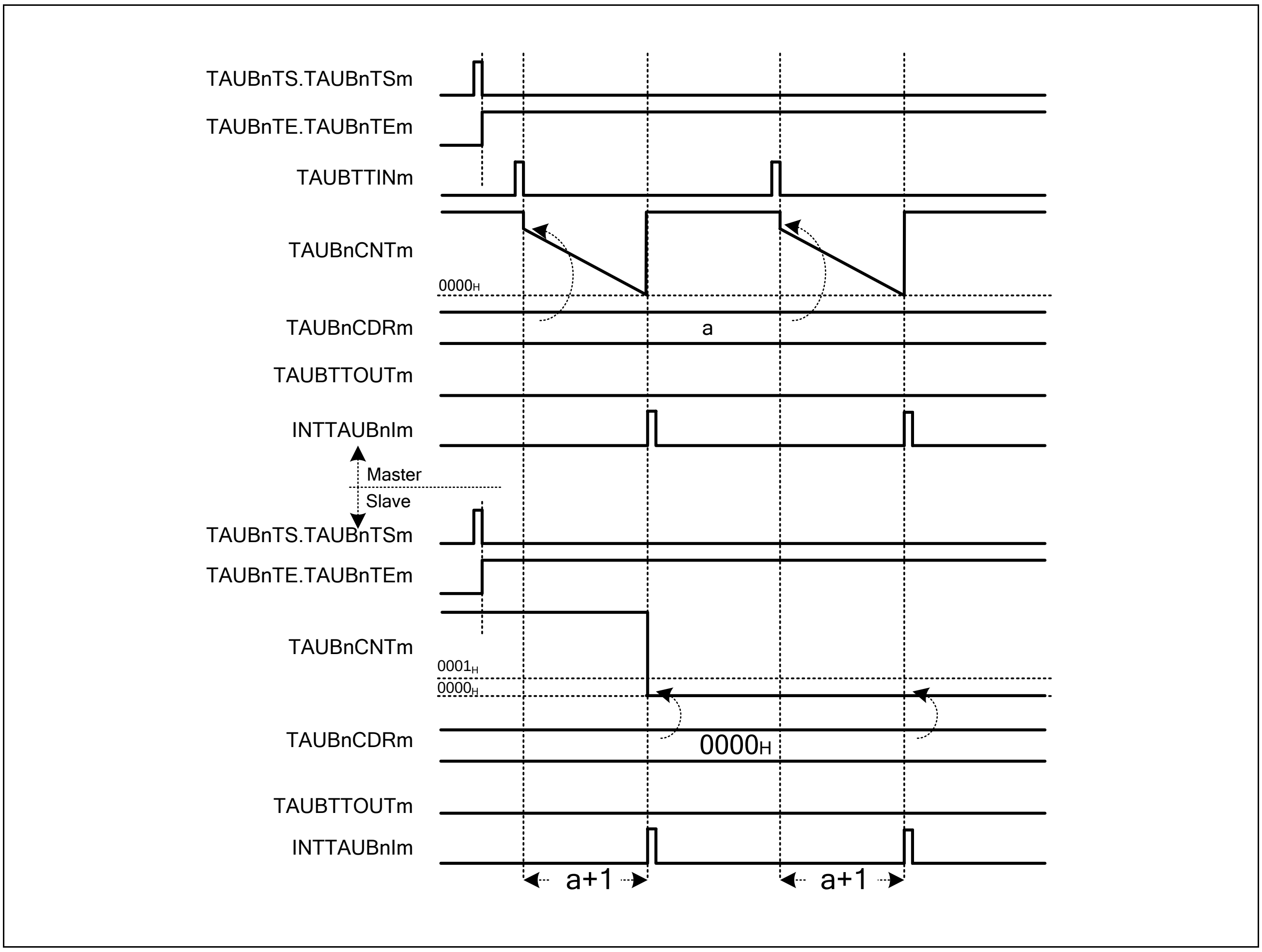


Figure 31.88 TAUBnCDRm (Slave) = 0000$_H$

- TAUBTTOUTm remains at inactive state, because the pulse width is zero.

(3) TAUBnCMORm.TAUBnMD0 = 0

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

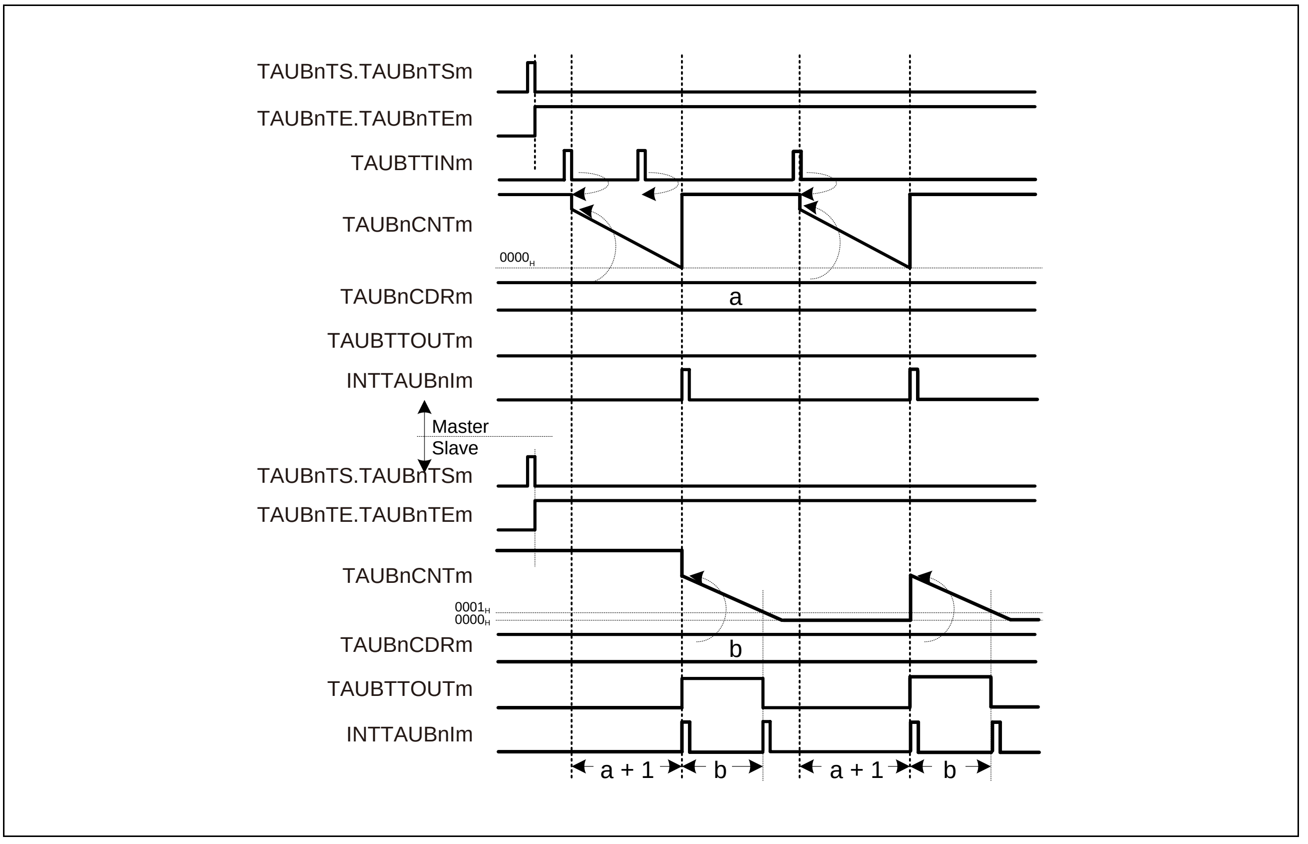


Figure 31.89 TAUBnCMORm.TAUBnMD0 = 0

- Even when an effective edge is input to TAUBTTINm while the counter of the master channel counts down, the counter continues counting down.

(4) TAUBnCMORm.TAUBnMD0 = 1

The following settings apply to this diagram.

- Start trigger detection enabled during counting (TAUBnCMORm.TAUBnMD0 = 1)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

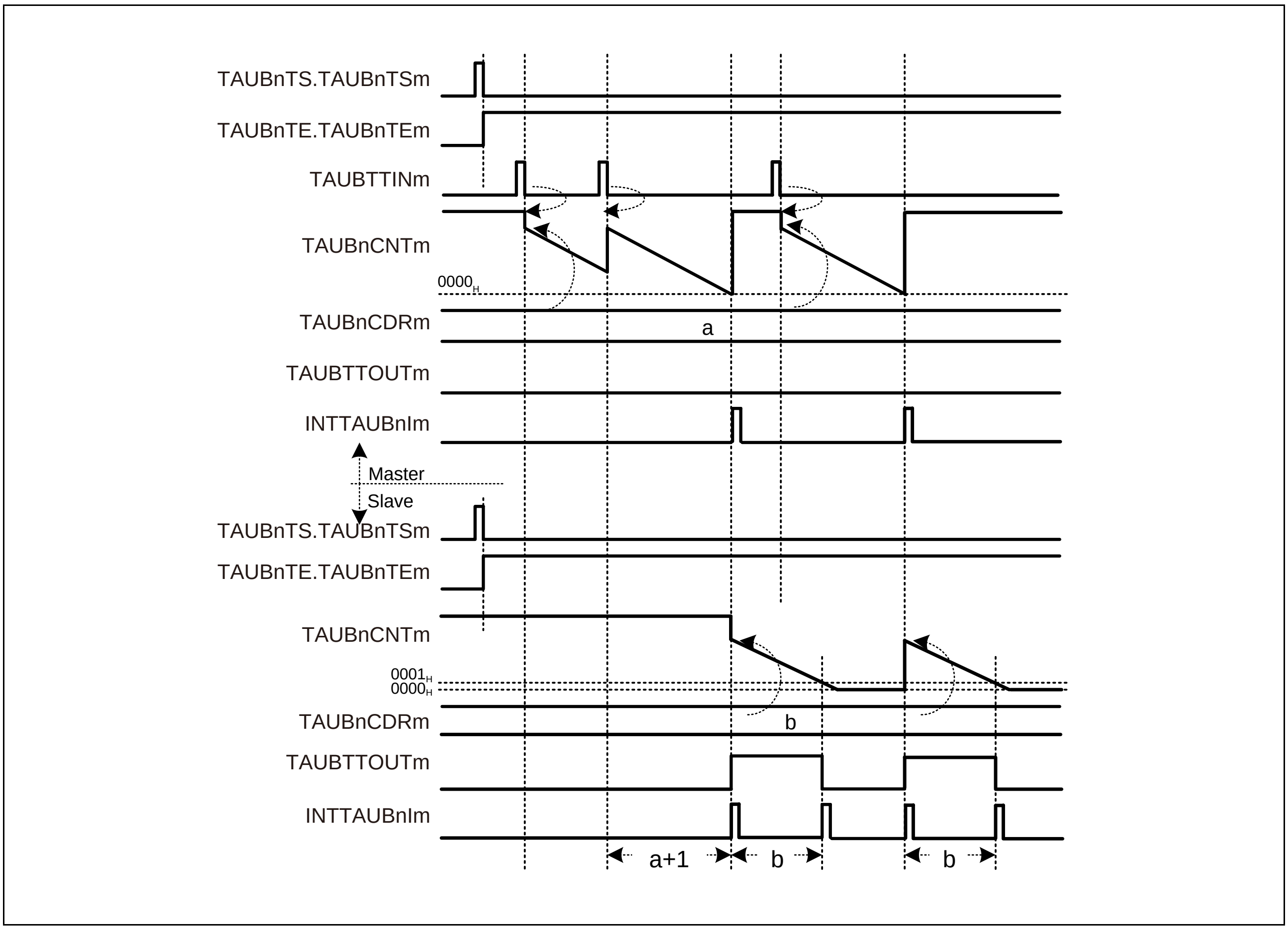


Figure 31.90 TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected while the counter of the master channel counts down, TAUBnCNTm reloads the value of TAUBnCDRm. The counter restarts to count down.

This means the delay for INTTAUBnIm generation interval is extended by the value of TAUBnCNTm at the time when a valid TAUBTTINm input edge is detected.

(5) Operation stop and restart

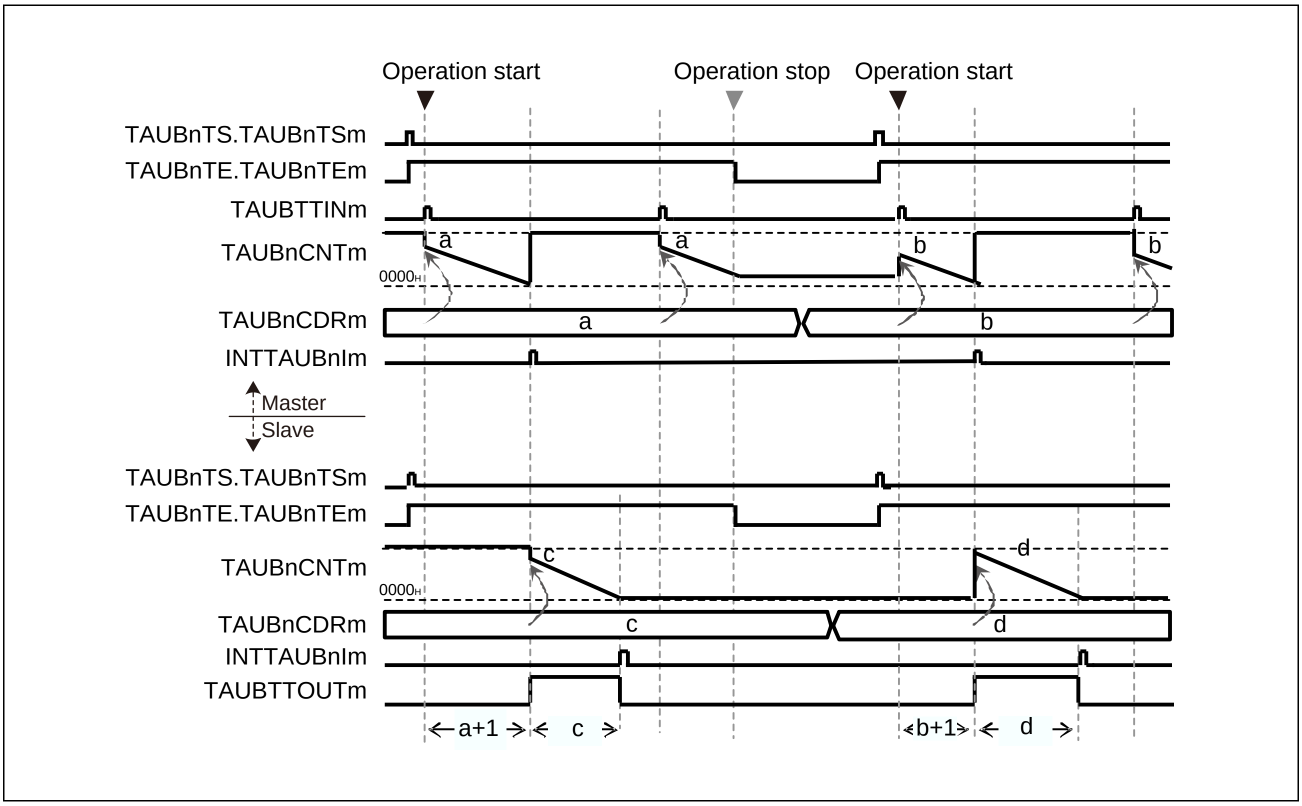


Figure 31.91 Stopping and Restarting the Operation

Setting TTm of the master and slave channels to 1 clears TAUBnTE.TAUBnTEm to 0, thereby stopping the count operation. If this happens, TAUBnCNTm and TAUBTTOUTm stop operation with the values retained.

Setting TAUBnTS.TAUBnTSm of the master and slave channels to 1 concurrently sets TAUBnTE.TAUBnTEm to 1.

When the start trigger is detected while the TAUBnTE.TAUBnTEm is set to 1, the TAUBnCDRm value is transferred to TAUBnCNTm and the operation restarts.

(6) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

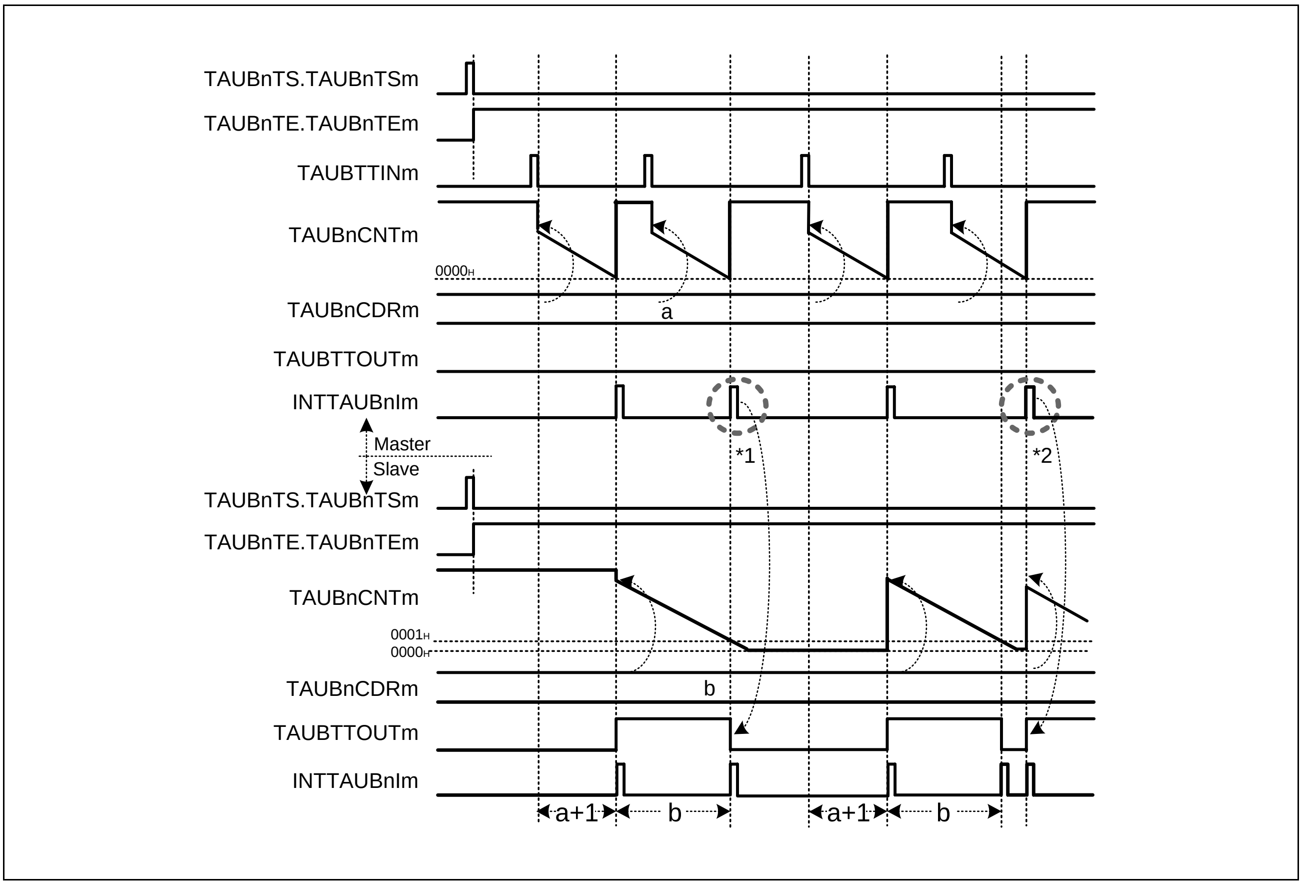


Figure 31.92 Input Interval of TAUBTTINm ≤ Delay Time + Pulse Width + 1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached*[1], the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUBnCDRm (slave) is reloaded. An interrupt is generated and TAUBTTOUTm toggles. If TAUBnCNTm (master) has started to count down while the TAUBnCNTm (slave) is still counting*[2], TAUBTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

31.14.3 Delay Pulse Output Function

31.14.3.1 Overview

Summary

This function outputs two signals. The reference signal has a defined pulse width and pulse cycle specified using the master channel and slave channel 1.
Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by amount specified in slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1.
  The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified in slave channel 2.

Prerequisites

- Four channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 31.131, Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function**.
- The operation mode of slave channels 1 and 2 must be set to one-count mode, see **Table 31.134, Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function** and **Table 31.138, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function**.
- The operation mode of slave channel 3 must be set to pulse one-count mode, see **Table 31.141, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function**
- TAUBTTOUTm is not used for the master channel and slave channel 2
- The channel output mode of slave channel 1 must be set to synchronous channel output mode 1.
- The channel output mode of slave channel 3 must be set to independent channel output mode 1.

Description

The counters of the channel group are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm to 1, enabling count operation.

- Master channel:

  The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value. INTTAUBnIm is generated on the master channel.
  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter reloads the TAUBnCDRm value and counts down.

- Slave channels 1 and 2:

  When the slave channels 1 and 2 detect an interrupt from the master channel, they start to count down from the current value of TAUBnCDRm. The TAUBTTOUTm signal (slave 1) is set.

  - Slave channel 1:
    When the counter of slave channel 1 reaches 0000_H (duty time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.

  - Slave channel 2:
    When the counter of slave channel 2 reaches 0000_H, delay time has elapsed and INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.
    INTTAUBnIm (slave channel 2) triggers the counter of slave channel 3

- Slave channel 3:

  When slave channel 3 detects an interrupt from slave channel 2, it starts to count down from the current value of TAUBnCDRm. INTTAUBnIm is generated and the TAUBTTOUTm signal (slave 3) is set.
  When the counter of slave channel 3 reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset.
  The output from slave channel 3 is the delayed PWM pulse.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 31.6, Simultaneous Rewrite**.

31.14.3.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUBnCDRm (slave 1)) × count clock cycle

Delay = (TAUBnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUBnCDRm (slave 3)) × count clock cycle

Where the setting of the delay is within the following range:

$0000_H \leq$ TAUBnCDRm (slave 2) < TAUBnCDRm (master)

NOTES

1. The output waveform of TAUBTTOUTm (slave 3) is the output waveform of TAUBTTOUTm (slave 1) delayed for the delay generated by slave 2. It cannot be delayed for more than the pulse cycle.
2. When INTTAUBnIm of slave 2 occurs while slave 3 is counting, slave 3 restarts the operation. Therefore, the output waveform of TAUBTTOUTm (slave 3) retains the active level. (In this case, TAUBTTOUTm (slave 3) cannot output the waveform of the delayed basic pulse of TAUBTTOUTm (slave 1).)

31.14.3.3 Block Diagram and General Timing Diagram

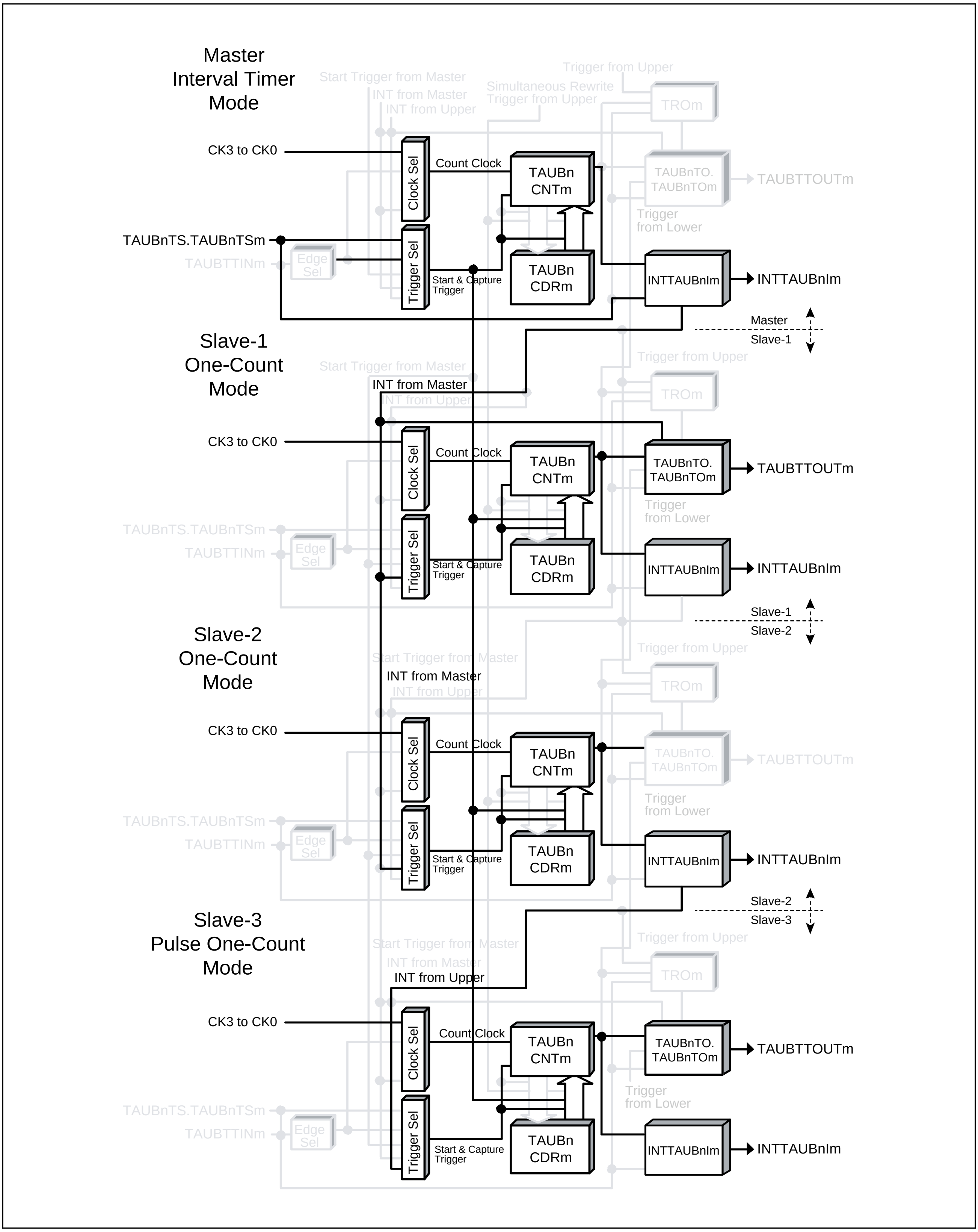


Figure 31.93 Block Diagram for Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

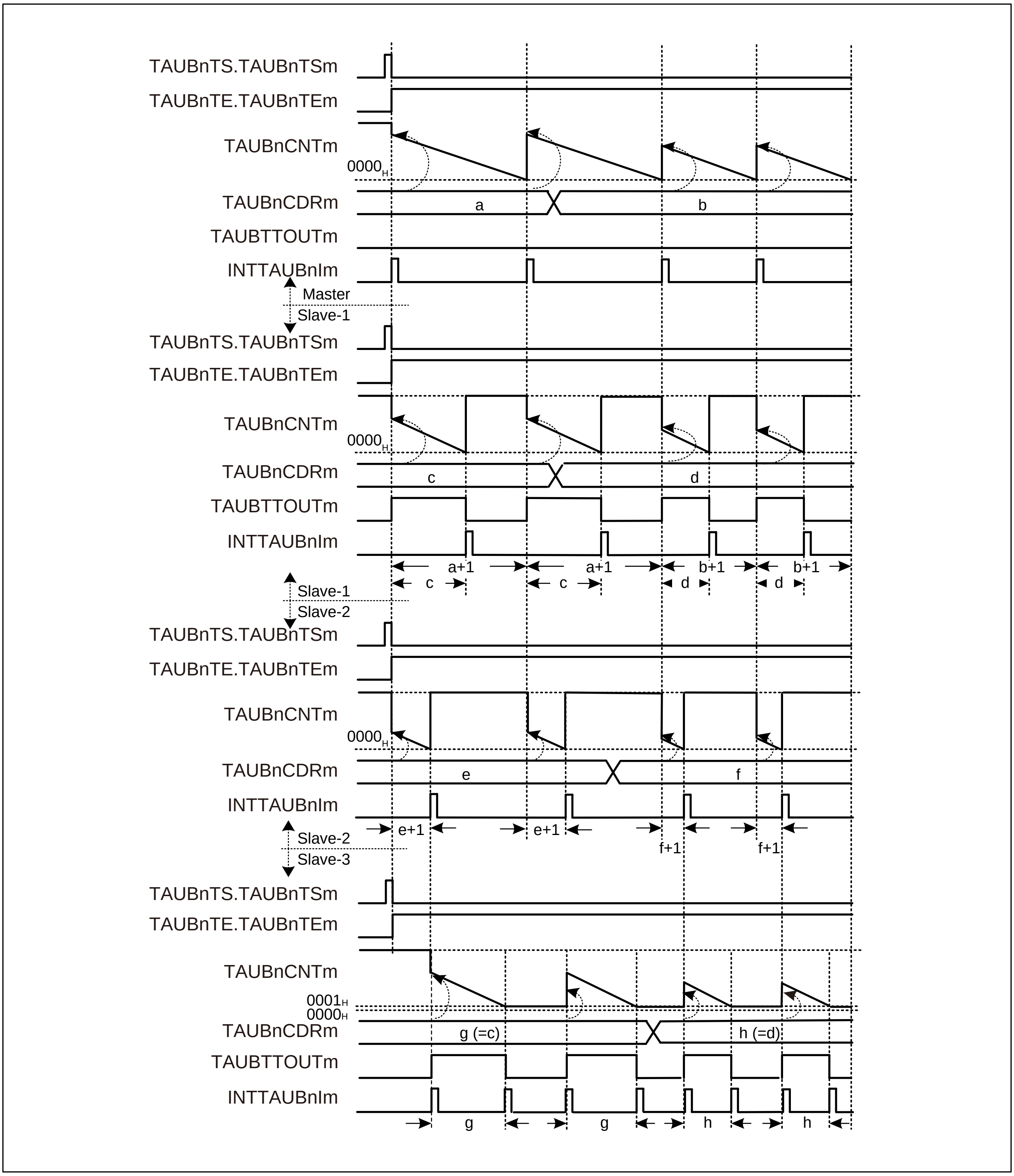


Figure 31.94 General Timing Diagram for Delay Pulse Output Function

31.14.3.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 31.131 Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.132 Contents of the TAUBnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by the master channel of this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.133 Simultaneous Rewrite Settings for the Master Channel of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.3.5 Register Settings for Slave Channel 1

(1) TAUBnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.134 Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.135 Contents of the TAUBnCMURm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 1

Table 31.136 Control Bit Settings for Slave Channel 1 of the Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.137 Simultaneous Rewrite Settings for Slave Channel 1 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.3.6 Register Settings For Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.138 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_{B}.
11	TAUBnMAS	Write 0_{B}.
10 to 8	TAUBnSTS[2:0]	Write 100_{B}.
7, 6	TAUBnCOS[1:0]	Write 00_{B}.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_{B}.
0	TAUBnMD0	Write 1_{B}.

(2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.139 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.140 Simultaneous Rewrite Settings for Slave Channel 2 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.3.7 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.141 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 101_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.142 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 31.143 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simulta00neous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.144 Simultaneous Rewrite Settings for Slave Channel 3 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.3.8 Operating Procedure for Delay Pulse Output Function

Table 31.145 Operating Procedure for Delay Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.3.4, Register Settings for the Master Channel**. Slave channel 1: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.3.5, Register Settings for Slave Channel 1**. Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.3.6, Register Settings For Slave Channel 2**. Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.3.7, Register Settings for Slave Channel 3**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.

Table 31.145 Operating Procedure for Delay Pulse Output Function

Restart operation

	Operation	Status of TAUBn
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master channel and slave channels 1 and 2 start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave 1) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channels 1 and 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000_H: ● INTTAUBnIm (master) is generated ● TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation ● TAUBnCNTm (slave 1 and slave 2) reload the TAUBnCDRm value and start counting down ● TAUBTTOUTm (slave 1) is set When TAUBnCNTm (slave 1) reaches 0000_H: ● INTTAUBnIm (slave 1) is generated ● TAUBTTOUTm (slave 1) is reset When TAUBnCNTm (slave 2) reaches 0000_H: ● INTTAUBnIm (slave 2) is generated ● INTTAUBnIm (slave 3) is generated ● TAUBTTOUTm (slave 3) is set ● TAUBnCNTm (slave 3) reloads the TAUBnCDRm value and starts counting down When TAUBnCNTm (slave 3) reaches 0001_H: ● INTTAUBnIm (slave 3) is generated ● TAUBTTOUTm (slave 3) is reset
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

31.14.3.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = $000B_H$
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = $000B_H$

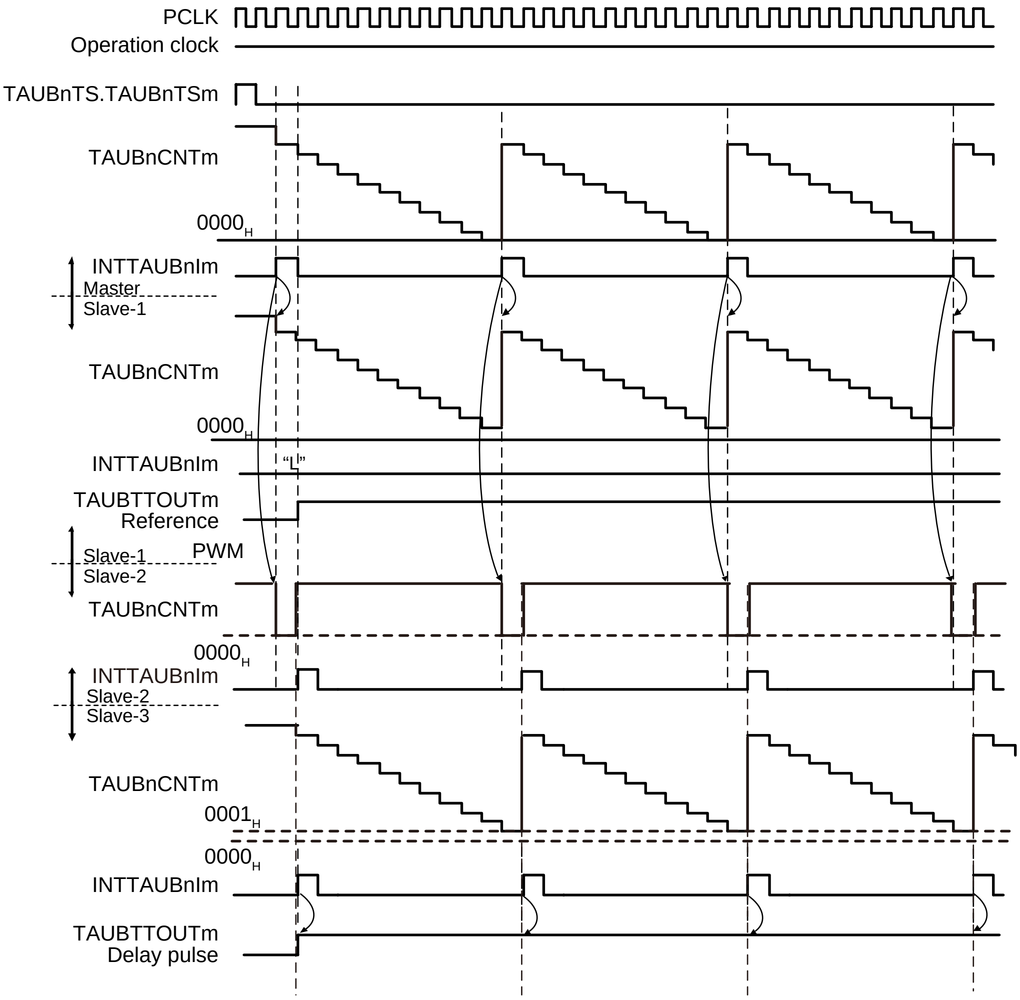


Figure 31.95 Duty Cycle (Slave 3) = 100%

- If the value of TAUBnCDRm (slave 1 and 3) is higher than the value of TAUBnCDRm (master), the counter of slave channel 1 cannot reach 0000_H and cannot generate interrupt request signals. TAUBTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = 0005_H
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = 0005_H

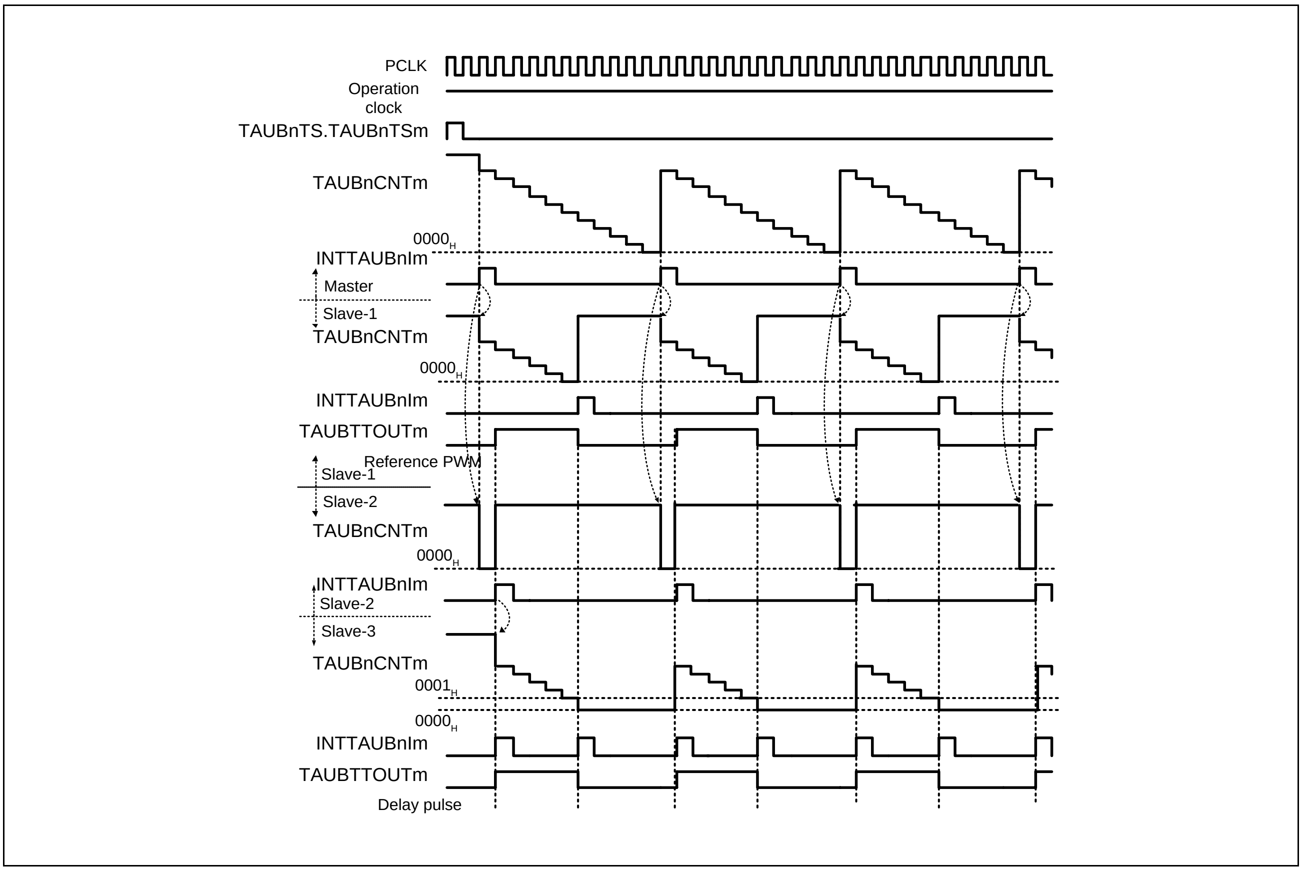


Figure 31.96 TAUBTTOUTm (Slave 1) = TAUBTTOUTm (Slave 3)

- If TAUBnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

31.14.4 AD Conversion Trigger Output Function Type 1

31.14.4.1 Overview

Summary

This function is identical to **Section 31.14.1, PWM Output Function** except that TAUBTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

31.14.4.2 Block Diagram and General Timing Diagram

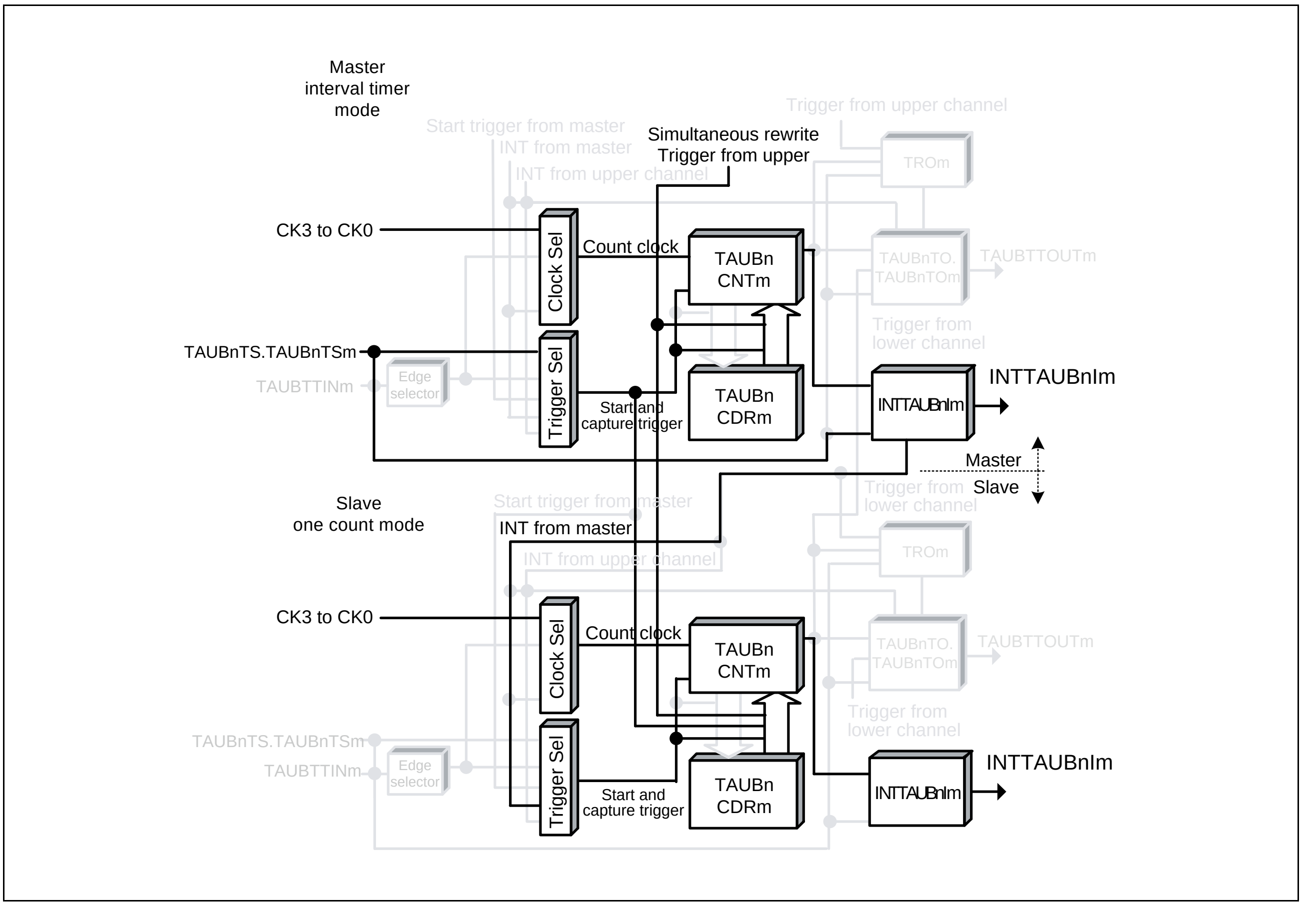


Figure 31.97 Block Diagram for AD Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

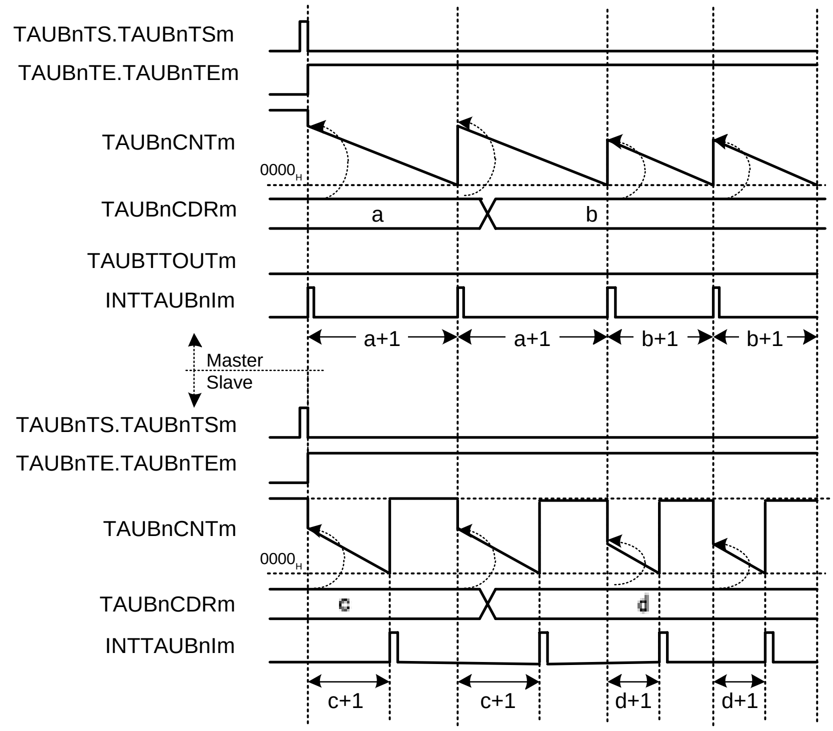


Figure 31.98 General Timing Diagram for AD Conversion Trigger Output Function Type 1

31.14.5 Triangle PWM Output Function

31.14.5.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master channel generates a carrier cycle from two pulse cycles. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slaves counter.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 31.146, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function**.
- The operation mode of the slave channel(s) must be set to up down count mode, see **Table 31.150, Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function**.
- The channel output mode of the master channel must be set to independent channel output mode 1.
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 0, TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnOEm is 0.

Functional description

The counters are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1 for every channel. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation. The current values of TAUBnCDRm (master and slave) are written to TAUBnCNTm (master and slave) and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:
  When the counter of the master channel reaches 0000_H (pulse cycle time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then reloads the TAUBnCDRm value and counts down.
- Slave channel:
  INTTAUBnIm generated on the master channel triggers the counter of the slave channel:
  - If the slave counter currently counts down, it changes count direction.
  - If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUBnIm is generated and the TAUBTTOUTm (slave) signal is set or reset.

The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.

TAUBTTOUTm can be switched between positive and negative phase setting TAUBnTOL.TAUBnTOLm during operation.

The counters can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 31.6, Simultaneous Rewrite**.

31.14.5.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle =

[(TAUBnCDRm (master) + 1 - TAUBnCDRm (slave))/ (TAUBnCDRm (master) + 1)] × 100

- Duty cycle = 100%
  TAUBnCDRm (slave) = 0000_H
- Duty cycle = 0%
  TAUBnCDRm (slave) $\geq$ TAUBnCDRm (master) + 1

31.14.5.3 Block Diagram and General Timing Diagram

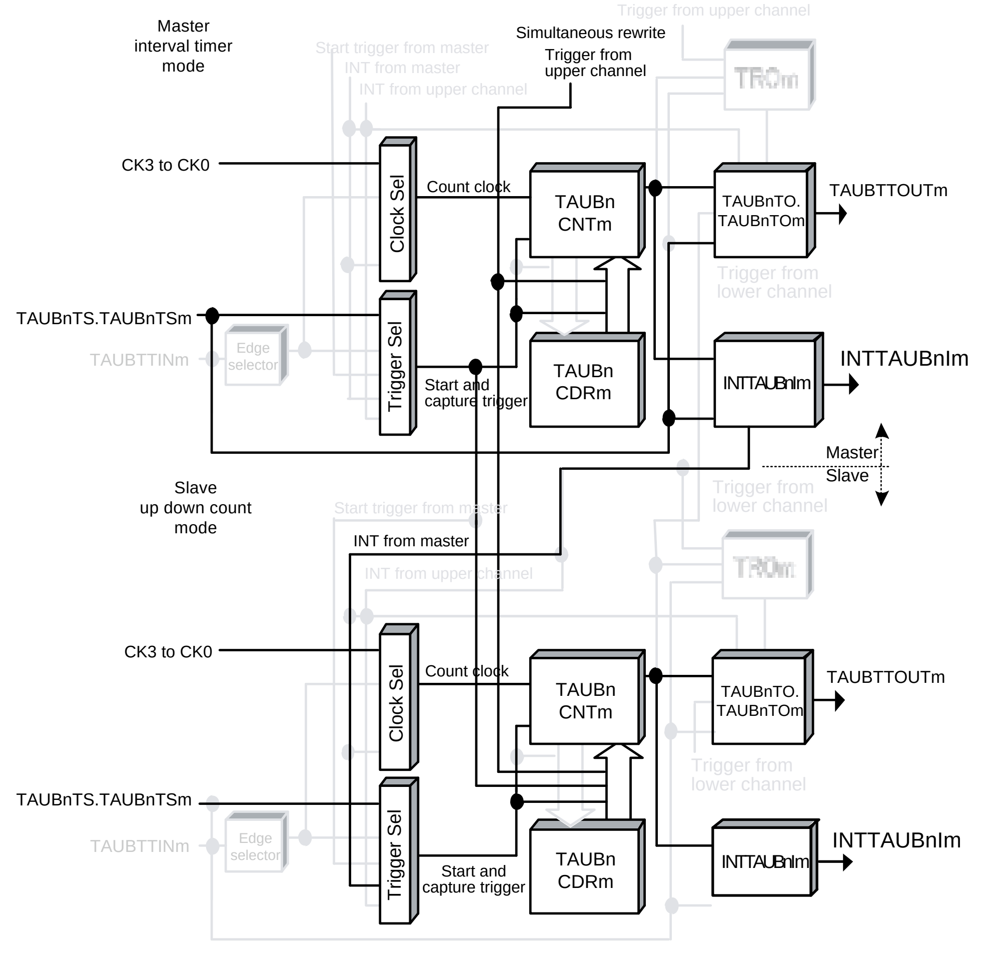


Figure 31.99 Block Diagram for Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

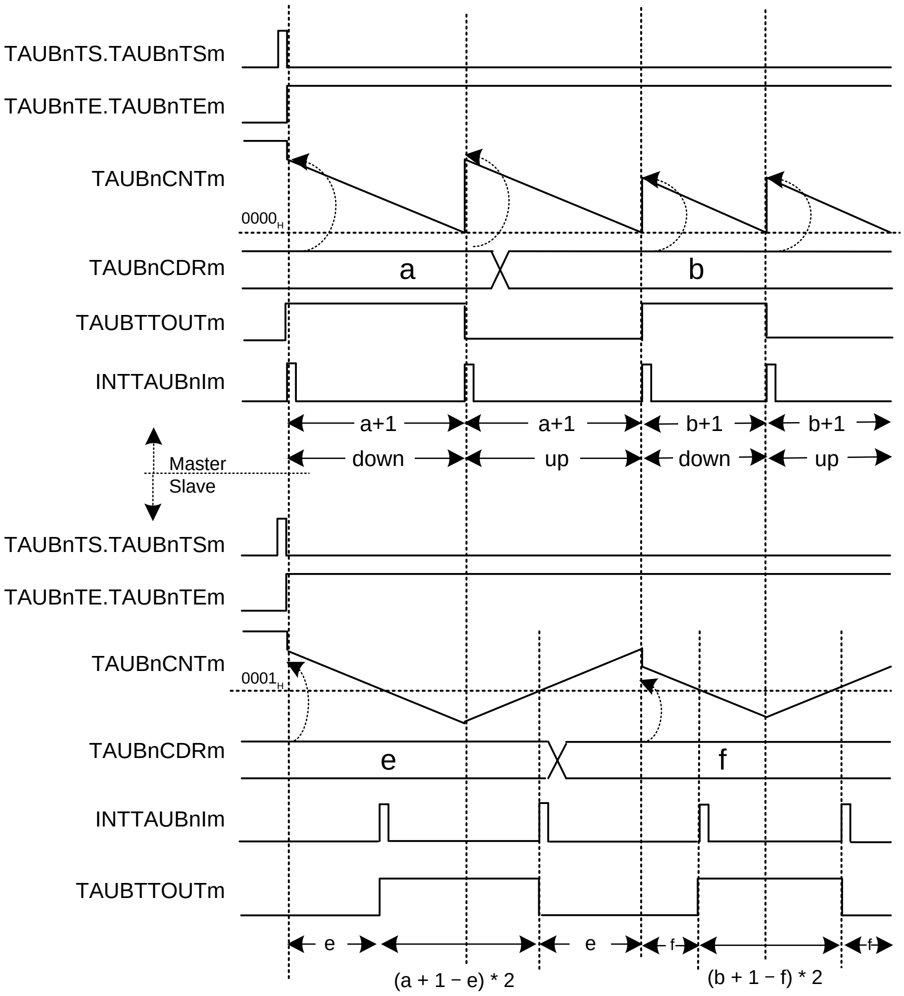


Figure 31.100 General Timing Diagram for Triangle PWM Output Function

31.14.5.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.146 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.147 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 31.148 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.149 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

31.14.5.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.150 Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.151 Contents of the TAUBnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 31.152 Control Bit Settings for Synchronous Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.153 Simultaneous Rewrite Settings for the Slave Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.5.6 Operating Procedure for Triangle PWM Output Function

Table 31.154 Operating Procedure for Triangle PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.5.4, Register Settings for the Master Channel** Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.5.5, Register Settings for the Slave Channel(s)** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated on the master channel when TAUBnCMORm.TAUBnMD0 set to 1.
During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master and slave channels loads TAUBnCDRm and counts down. When the counter of the master channel reaches 0000$_H$: ● INTTAUBnIm (master) is generated ● TAUBTTOUTm (master) toggles ● TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation. ● TAUBnCNTm (slave) reloads the TAUBnCDRm value or counts in the reverse direction. When TAUBnCNTm of the slave = 0001$_H$: ● INTTAUBnIm (slave) is generated ● TAUBTTOUTm (slave) is set (in count-down status) or reset (in count-up status)
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

31.14.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 6_H

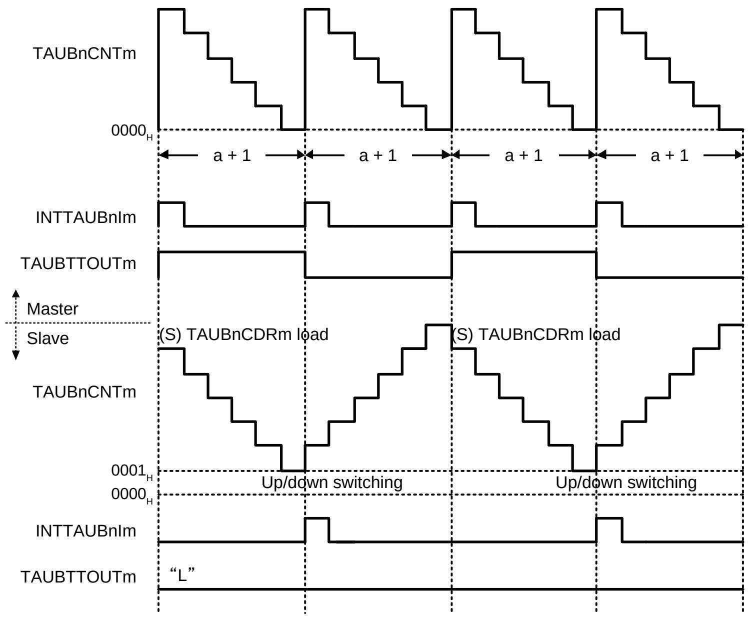


Figure 31.101 TAUBnCDRm (Slave) ≥ TAUBnCDRm (Master) + 1

- If TAUBnCDRm (slave) ≥ TAUBnCDRm (master) +1, INTTAUBnIm of slave channel is not generated during counting down. The set signal is never detected, so TAUBTTOUTm remains at low state.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 0_B

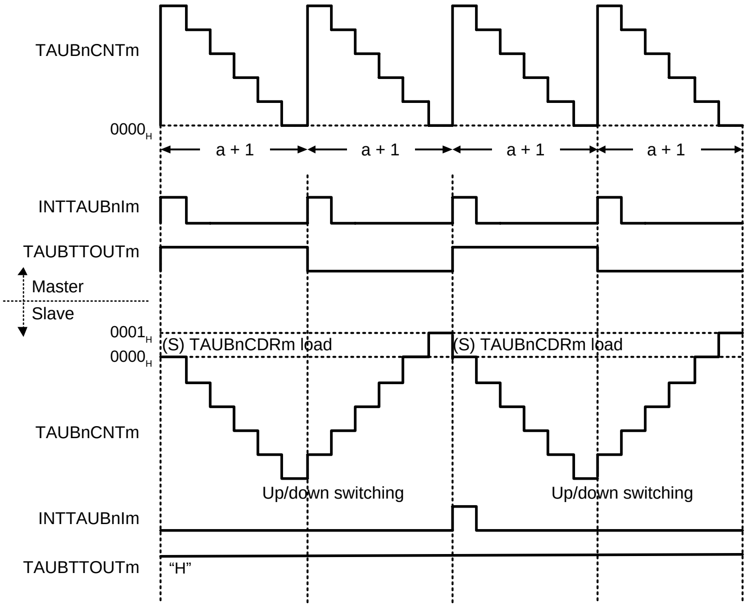


Figure 31.102 TAUBnCDRm (Slave) = 0000_H

- If TAUBnCDRm (slave) = 0000_H, INTTAUBnIm of slave channel is not generated during counting up. The reset signal is never detected, so TAUBTTOUTm remains at high state.

31.14.6 Triangle PWM Output Function with Dead Time

31.14.6.1 Overview

Summary

This function generates multiple triangle PWM outputs with a defined dead time by using a master and two or more slave channels. The resulting PWM signals with the dead time are output via TAUBTTOUTm of the slave channels 2 and 3. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master generates a carrier cycle. The first pulse controls the down status and the second pulse controls the up status of the slaves counter.

An interrupt on slave 2 causes TAUBTTOUTm of the slave channels to be set or reset.
Depending on the settings of TAUBnTDL.TAUBnTDLm, delay time is added to positive or negative logic side of the signal (i.e. whether TAUBTTOUTm is set or reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. Select an even channel CH (a) and an odd channel CH (a + 1) for slave channel 2 and 3 respectively.
- The operation mode of the master channel must be set to interval timer mode, see **Table 31.156, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even-numbered channel (a), and slave channel 3 is an odd-numbered channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operation mode of slave channel 2 must be set to up down mode, see **Table 31.160, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time**
  Furthermore, slave channel 2 must be an even channel
- The operation mode of slave channel 3 must be set to one-count mode, see **Table 31.164, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time**
  Furthermore, slave channel 3 must be an odd channel
- The channel output mode of the master channel must be set to independent channel output mode 1
- The channel output mode of the slave channels 2 and 3 must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.MD0 (master) bit is set to 0, TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.MD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnTOEm is 0.

NOTE

Slave channel 1 is not used for Triangle PWM Output Function with Dead Time. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current values of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:

  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. The counter reloads the TAUBnCDRm value and counts down.

- Slave channel 2:

  INTTAUBnIm generated on the master channel triggers the counter of the slave channel 2:

  - If the slave counter currently counts down, it changes count direction.
  - If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.
When the counter value of slave channel 2 reaches 0001_H, INTTAUBnIm is generated.

- Slave channel 3:

  INTTAUBnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUBnCDRm (slave 3) is written to TAUBnCNTm (slave 3) and the counter starts to count down from this TAUBnCDRm value.
  When the counter reaches 0000_H, INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of slave channel 2.

The TAUBnTDL.TAUBnTDLm settings of the corresponding channel specify whether it is set/reset immediately, or after dead time has elapsed, as shown in **Table 31.155, Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2**.

The TAUBnTOL.TOLm settings specify whether set corresponds to a high signal (TAUBnTOL.TAUBnTOLm = 0) or a low signal (TAUBnTOL.TAUBnTOLm = 1).

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

TAUBnCDRm value of slave channel 2 can be set to 0000_H to output 100% TAUBTTOUTm.

NOTE

If a forced restart is executed during operation, TAUBTTOUTm is not output as a triangle PWM signal.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 31.6, Simultaneous Rewrite**.

TAUBnTOL.TAUBnTOLm and TAUBnTDL.TAUBnTDLm bits should be set before the counter starts, and slave channels 2 and 3 should have opposite TAUBnTOL.TAUBnTOLm settings or opposite TAUBnTDL.TAUBnTDLm settings.

Table 31.155 Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2

TAUBnTDL.TAUBnTDLm	Count Direction of Slave Channel 2 when Interrupt is Generated	TAUBTTOUTm Set/Reset Timing
0	Down	Set after dead time has elapsed
	Up	Reset immediately
1	Down	Set immediately
	Up	Reset after dead time has elapsed

31.14.6.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (positive phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 – (TAUBnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 + (TAUBnCDRm (slave 3) + 1)] × count clock cycle

31.14.6.3 Block Diagram and General Timing Diagram

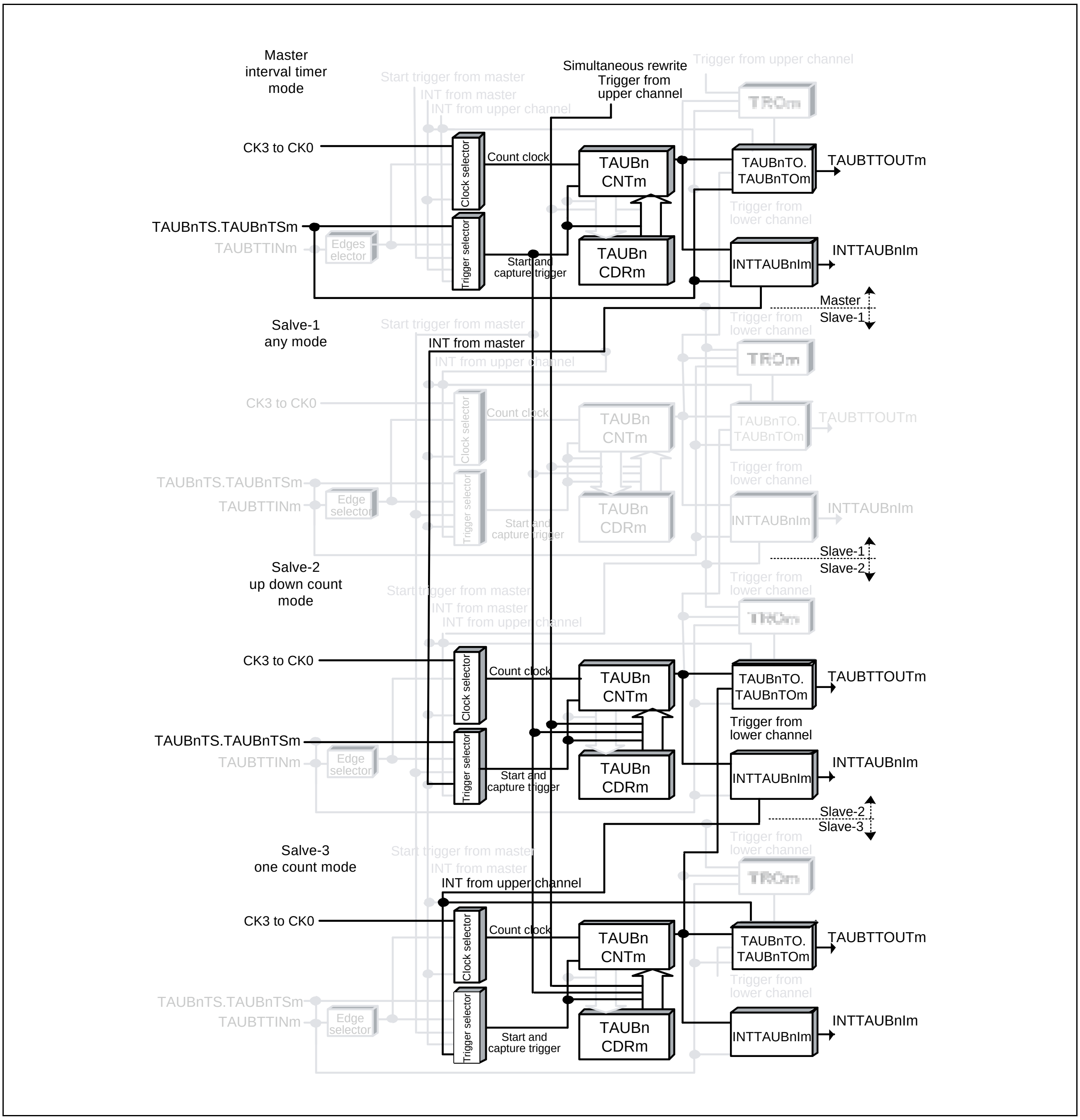


Figure 31.103 Block Diagram for Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Slave channel 2:
  - INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
  - TAUBnTDL.TAUBnTDLm = 0
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Enables start trigger detection during counting (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnTDL.TAUBnTDLm = 1
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)

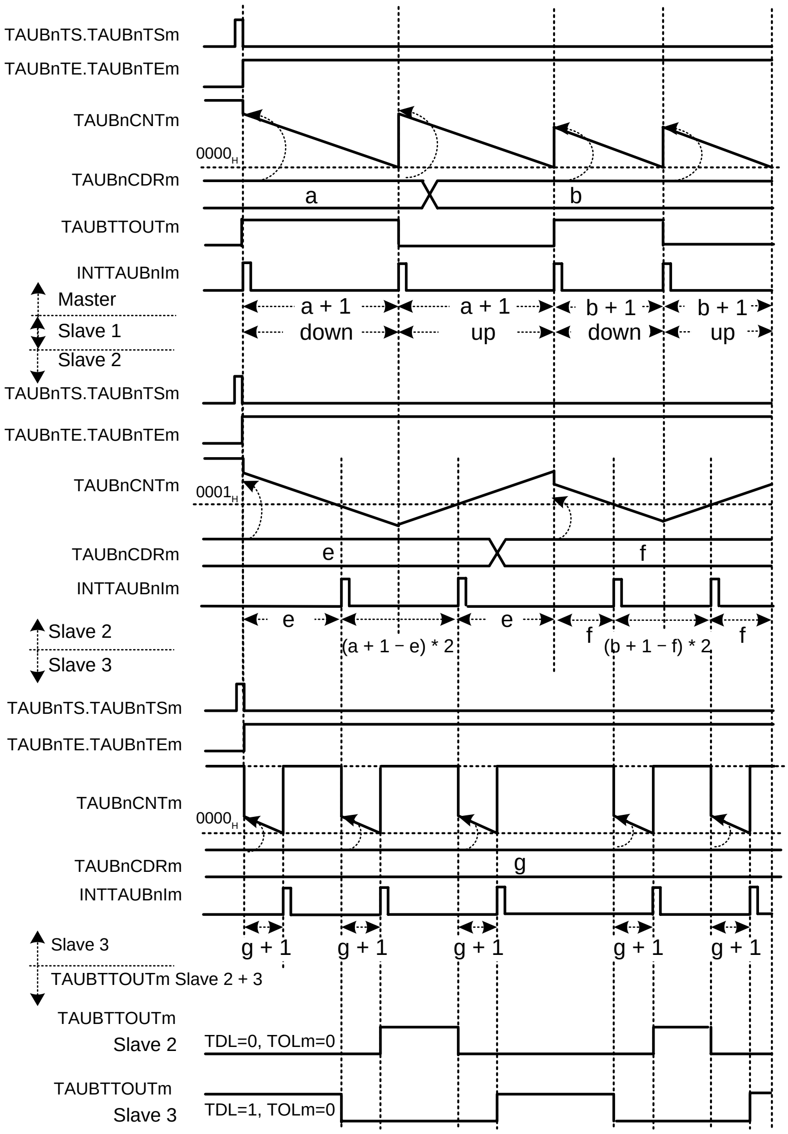


Figure 31.104 General Timing Diagram for Triangle PWM Output Function with Dead Time

31.14.6.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.156 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.157 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 31.158 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.159 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

31.14.6.5 Register Settings for Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.160 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.161 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Table 31.162 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the odd channel.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.163 Simultaneous Rewrite Settings for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.6.6 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 31.164 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 110_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 31.165 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 31.166 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the even channel.

(4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 31.167 Simultaneous Rewrite Settings for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

31.14.6.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 31.168 Operating Procedure for Triangle PWM Output with Dead Time

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.6.4Register Settings for the Master Channel** Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.6.5, Register Settings for Slave Channel 2** Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 31.14.6.6, Register Settings for Slave Channel 3** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated when TAUBnCMORm.TAUBnMD0 is set to 1 on the master channel.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channel 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000$_H$: ● INTTAUBnIm (master) is generated ● TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation ● TAUBnCNTm (slave 2) reloads the TAUBnCDRm value or counts in the reverse direction When TAUBnCNTm (slave 2) reaches 0001$_H$: INTTAUBnIm (slave 2) is generated TAUBnCNTm of slave channel 3 loads the TAUBnCDRm value and counts down When TAUBnCNTm of slave channel 3 = 0000$_H$: INTTAUBnIm is generated
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

31.14.6.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

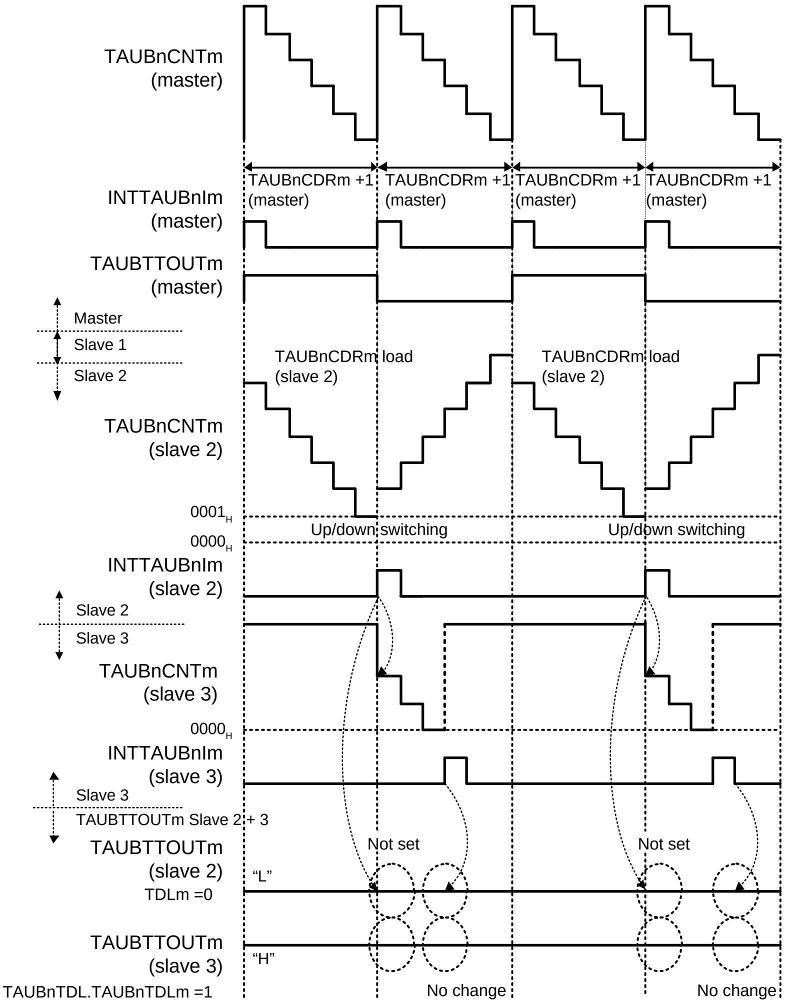


Figure 31.105 TAUBnCDRm (Slave 2) ≥ TAUBnCDRm (Master) + 1

- If TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master), the counter of slave channel cannot reach 0000_H during counting down. Therefore TAUBTTOUTm cannot toggle, i.e. it remains at its initial state. The interrupt from slave channel 2 occurs during count up, therefore it is a reset signal.

(2) Duty cycle = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

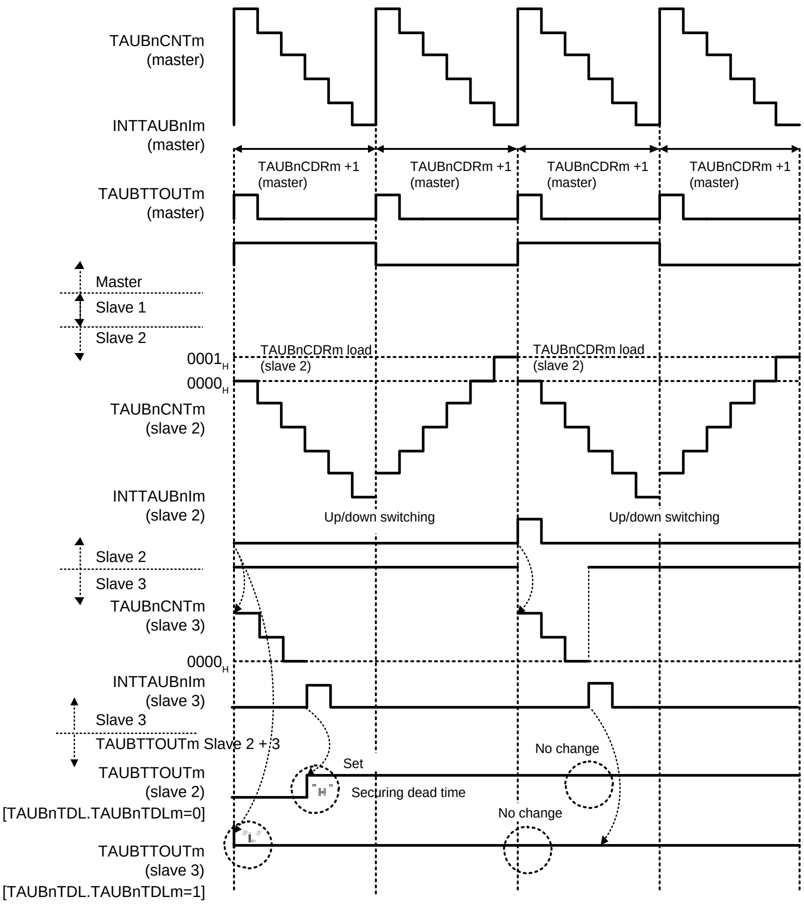


Figure 31.106 TAUBnCDRm (Slave 2) = 0000_H

- If TAUBnCDRm (slave 2) = 0000_H the counter of slave channel cannot reach 0001_H while counting up and therefore cannot generate an INTTAUBnIm while counting up.
  - The set conditions for a channel in which TAUBnTDL.TAUBnTDLm = 0 are met after dead time has elapsed. TAUBTTOUTm toggles but remains in the new state because the reset conditions are never satisfied for such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts.
    However, the reset conditions for a channel in which TAUBnTDL.TAUBnTDLm = 1 are never satisfied so TAUBTTOUTm remains in its initial state for such a slave channel.

(3) TAUBTTOUTm (slave 2) = 0% and TAUBTTOUTm (slave 3) ≥ 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

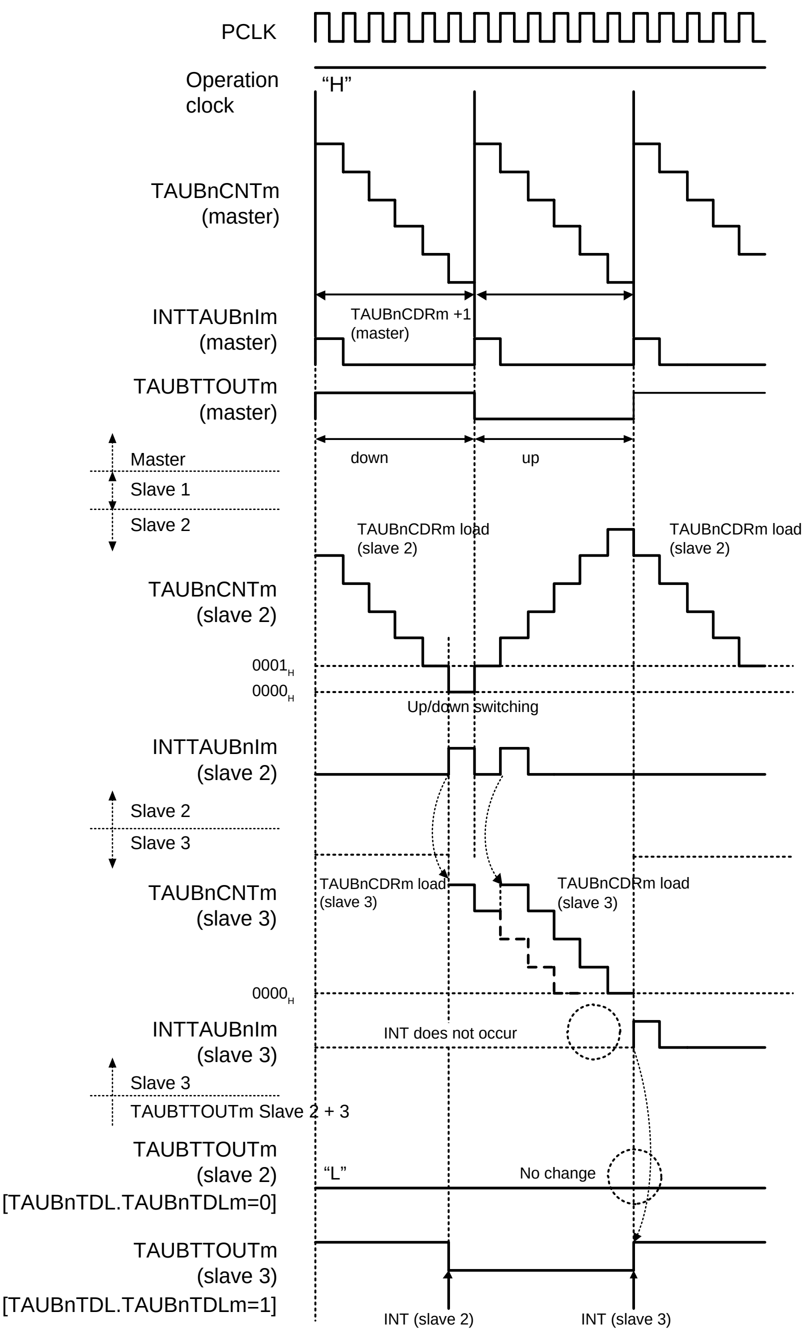


Figure 31.107 TAUBnCDRm (Master) = 0005_H, TAUBnCDRm (Slave 2) = 0005_H TAUBnCDRm (Slave 3) = 0004_H

- When the counter of slave channel 2 reaches 0000_H after detecting that the counter reached 0001_H, INTTAUBnIm (slave 2) is generated. The counter of slave channel 3 starts to count down.
- If another INTTAUBnIm (slave 2) is generated while the counter of slave channel 3 is still counting down, the value of TAUBnCDRm (slave 3) is reloaded and the counter restarts counting down from this value.
- In the diagram above, the first interrupt on channel 2 occurs while the counter is counting down, and the second while it is counting up.
- After the first interrupt, a slave for which TAUBnTDL.TAUBnTDLm = 0 waits for dead time to elapse before setting. However, if another interrupt occurs on slave 2, before the dead time has elapsed, the counter is counting up, so the signal acts as a reset signal, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 0 always remains inactive.
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 is set and reset as normal when the corresponding INTTAUBnIm is generated.

(4) TAUBTTOUTm (slave 2) > 0% and TAUBTTOUTm (slave 3) = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

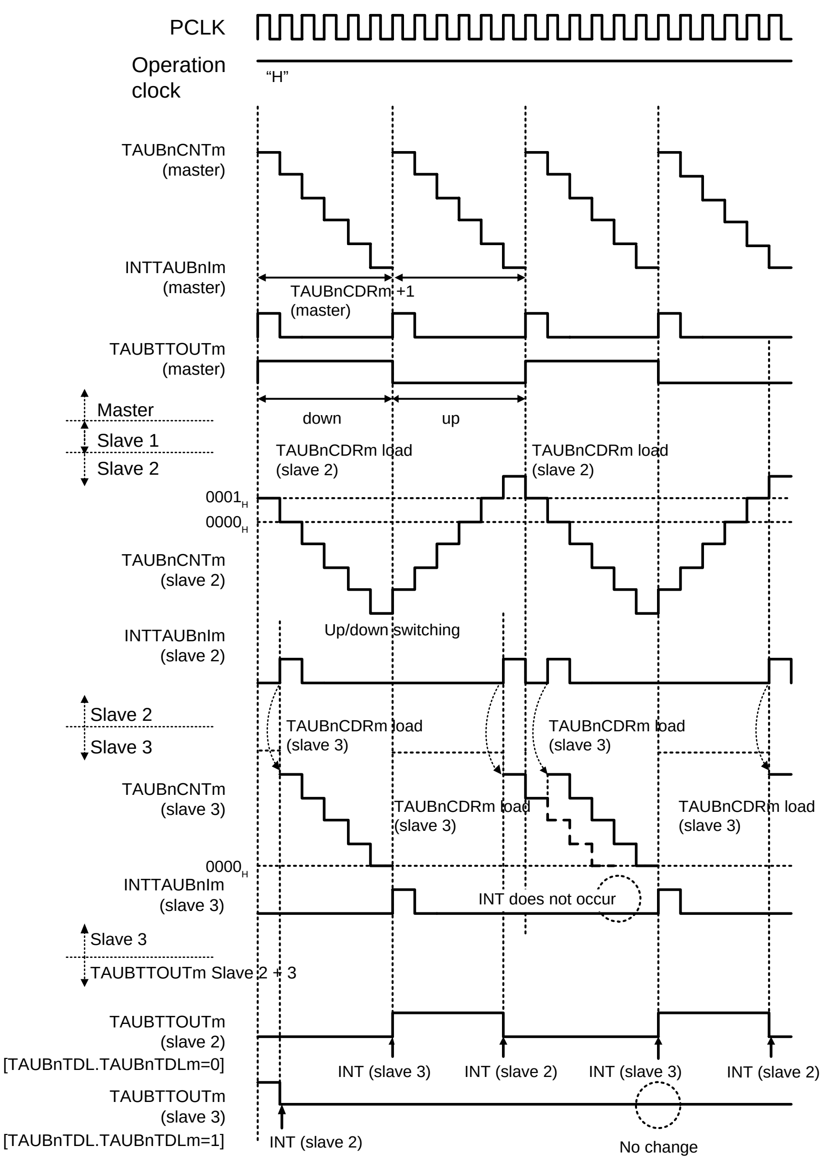


Figure 31.108 TAUBnCDRm (Master) = 0005H, TAUBnCDRm (Slave 2) = 0001H TAUBnCDRm (Slave 3) = 0004H PWM Signal Width (Negative Phase) ≥ Carrier Cycle

- After the second interrupt on slave channel 2, a slave for which TAUBnTDL.TAUBnTDLm = 1 is reset after the dead time has elapsed. However if another interrupt occurs on slave 2 before the dead time has elapsed, slave channel 3 is restarted, and then if an interrupt on slave channel 3 is generated, the counter is counting up, so the signal acts as a setting signal, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 1 always remains active.
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 is set and reset as normal when the corresponding INTTAUBnIm is generated.

(5) Inhibited INTTAUBnIm to set TAUBTTOUTm positive phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

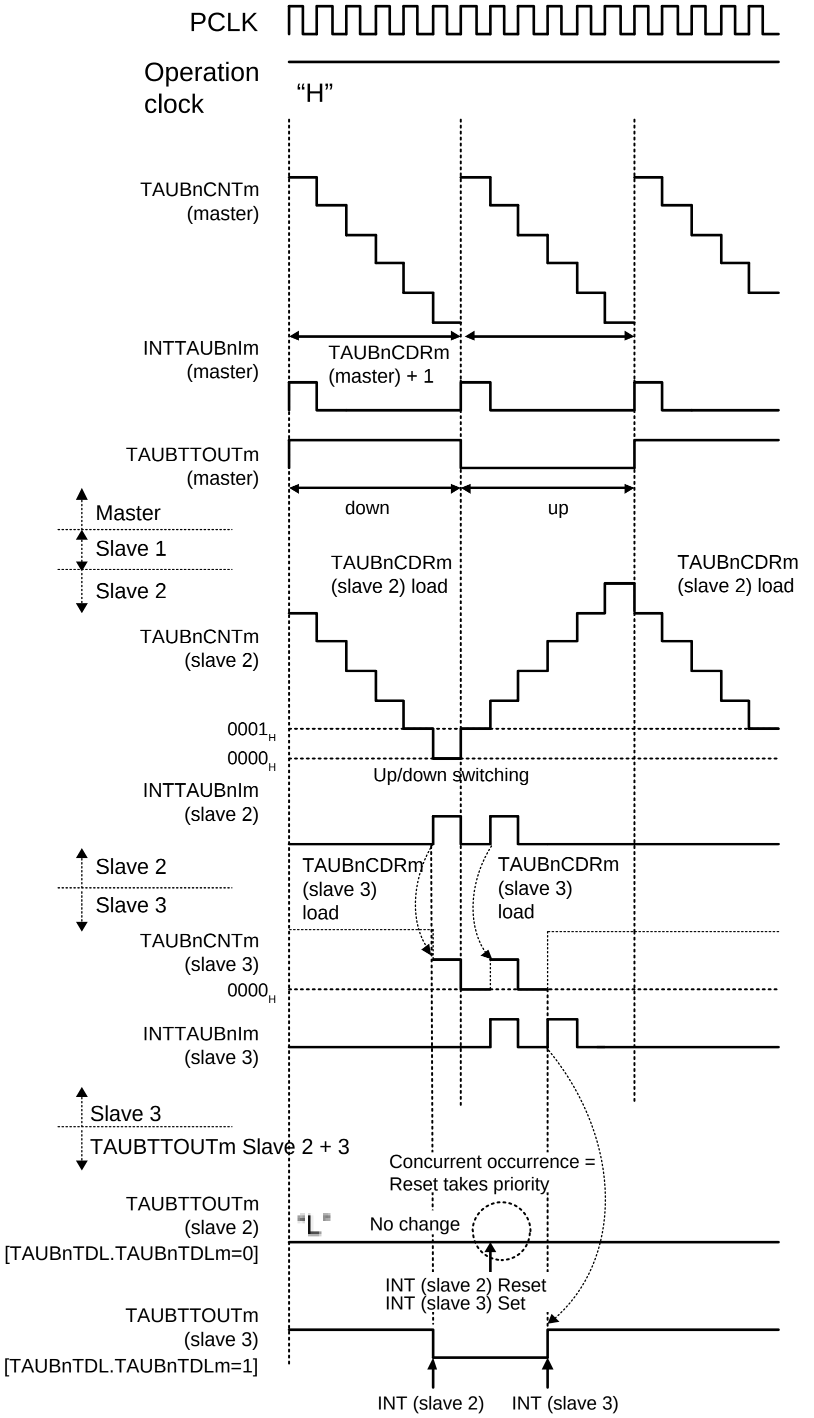


Figure 31.109 TAUBnCDRm (Master) = 0005$_H$, TAUBnCDRm (Slave 2) = 0005$_H$, TAUBnCDRm (Slave 3) = 0001$_H$
PWM Signal Width (Positive Phase) = 0

- The counter of slave channel 3 reaches 0000_H and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 0 (slave channel 2 in this example).
- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, this reset signal is given priority if TAUBnTOL.TAUBnTOLm = 0 (if TAUBnTOL.TAUBnTOLm = 1, the set signal is given priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 remains in the value after reset.

(6) Inhibited INTTAUBnIm to set TAUBTTOUTm negative phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

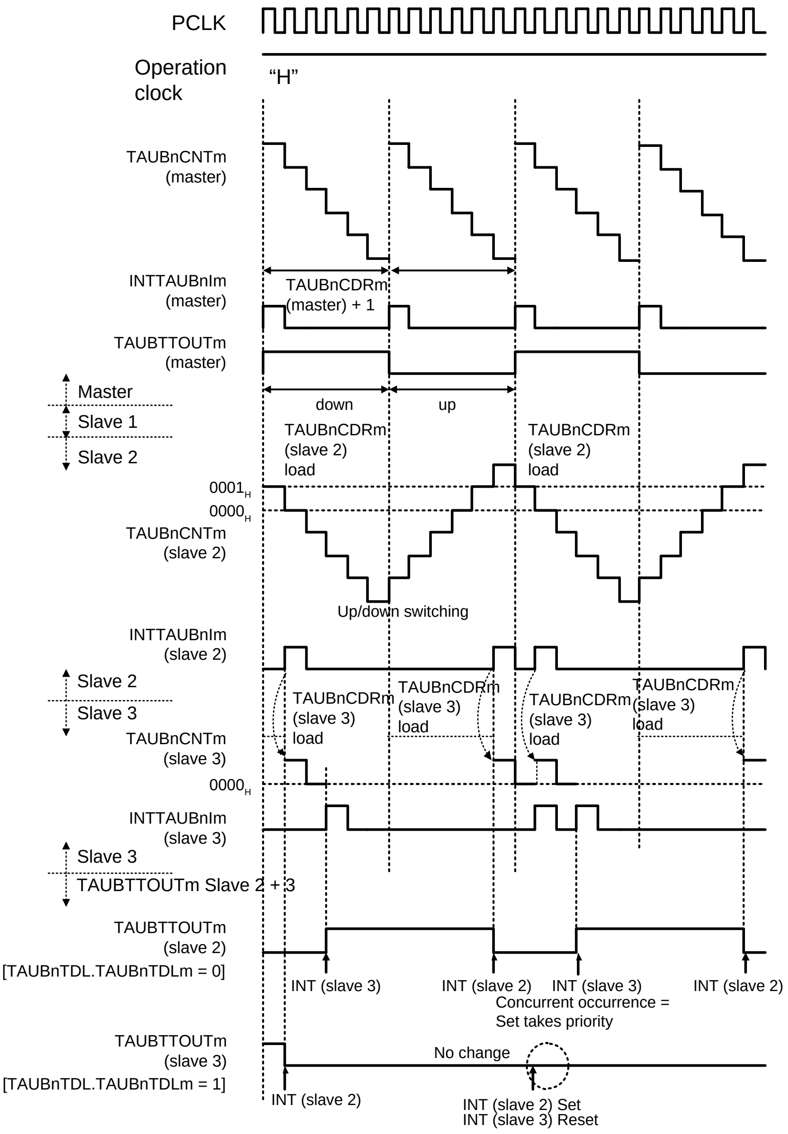


Figure 31.110 TAUBnCDRm (Master) = 0005H, TAUBnCDRm (Slave 2) = 0001H, TAUBnCDRm (Slave 3) = 0001H
PWM Signal Width (Negative Phase) = Carrier Cycle

- The counter of slave channel 3 reaches 0000$_H$ and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 1 (slave 3 in this example).
- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, the set signal is given priority if TAUBnTOL.TAUBnTOLm = 1 (if TAUBnTOL.TAUBnTOLm = 0, the reset signal is given priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 remains in the value after reset.

(7) Slave 2 TAUBnCDRm = 0000$_H$ (Duty cycle = 100%)

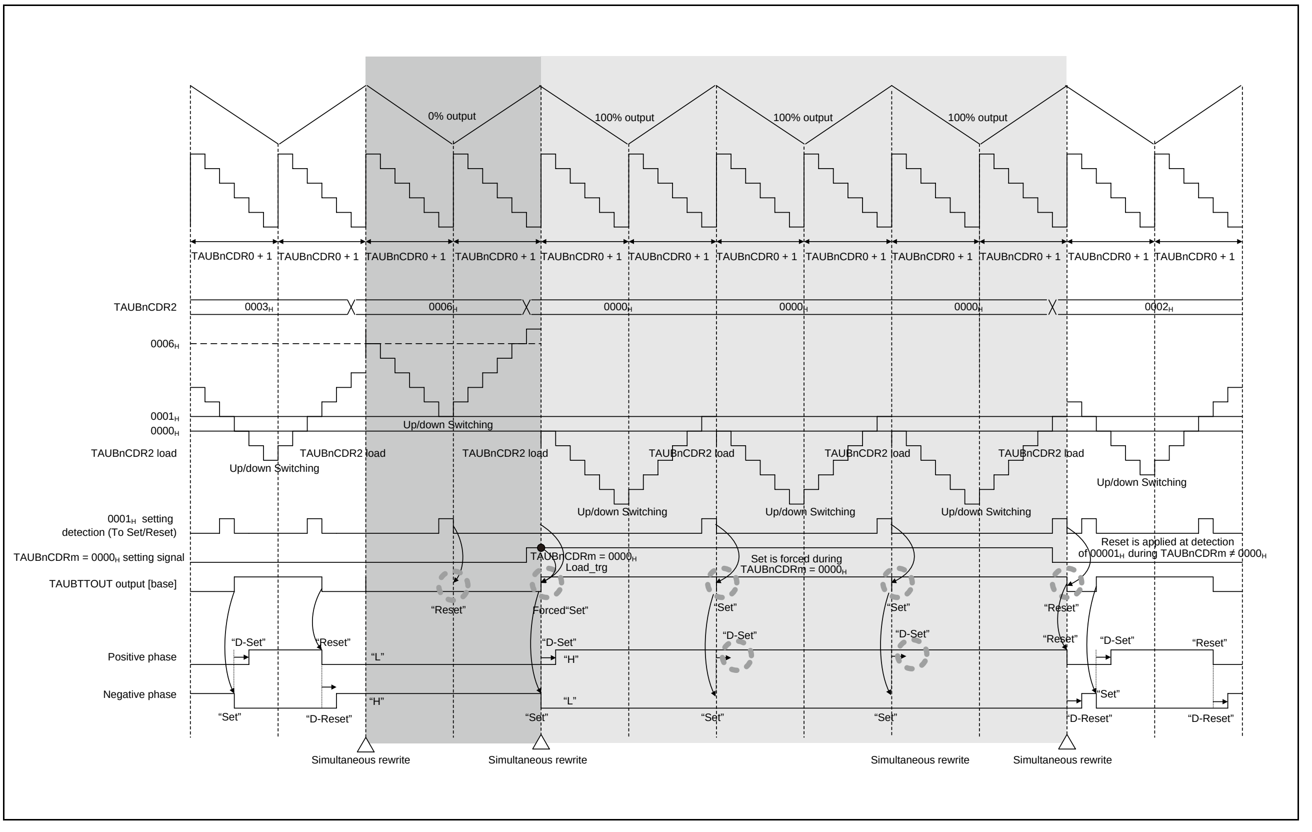


Figure 31.111 Slave 2 TAUBnCDRm = 0000$_H$ (Duty cycle = 100%)

When rewriting (slave channel 2) TAUBnCDRm ≠ 0000$_H$ to (slave channel 2) TAUBnCDRm = 0000$_H$ (100% output), set the negative phase side at the start of the carrier cycle, and set the positive phase side after dead time is secured.

When rewriting (slave channel 2) TAUBnCDRm = 0000$_H$ (100% output) to (slave channel 2) TAUBnCDRm ≠ 0000$_H$, reset the positive phase side at the end of the carrier cycle, and reset the negative phase side after dead time is secured.

31.14.7 A/D Conversion Trigger Output Function Type 2

31.14.7.1 Overview

Summary

This function is identical to **Section 31.14.5, Triangle PWM Output Function**, except that TAUBTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

31.14.7.2 Block Diagram and General Timing Diagram

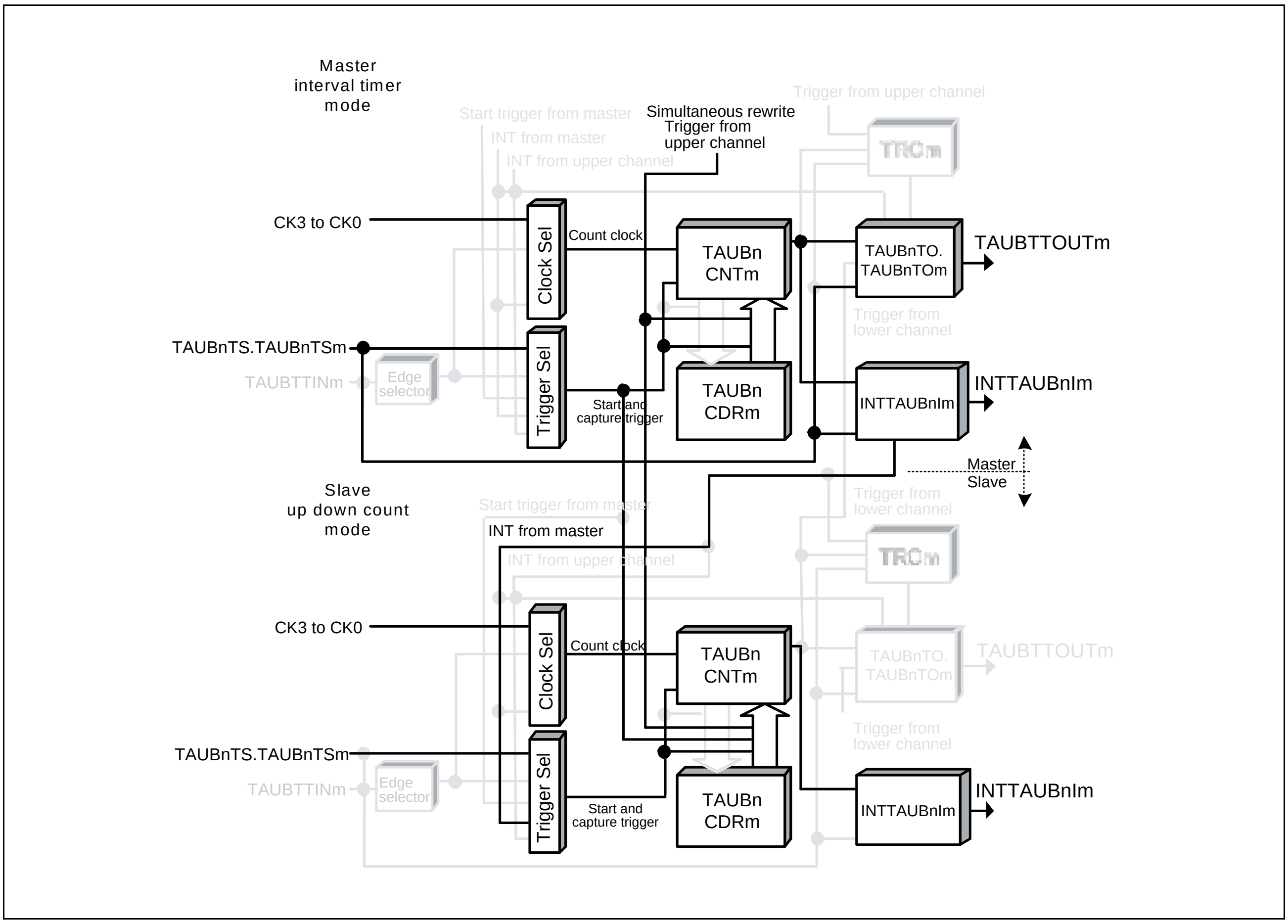


Figure 31.112 Block Diagram for A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

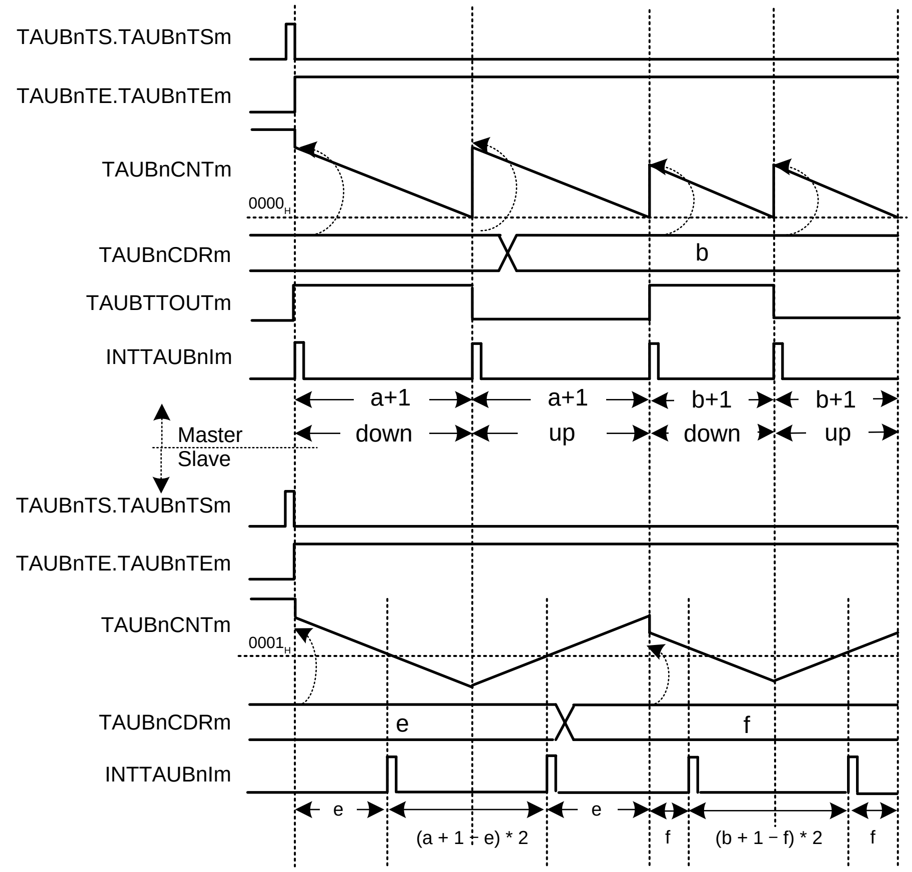


Figure 31.113 General Timing Diagram for A/D Conversion Trigger Output Function Type 2

Section 32 Timer Array Unit D (TAUD)

This section contains a generic description of the timer array unit D (TAUD).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of TAUD.

32.1 Features of RH850/F1KH, RH850/F1KM TAUD

32.1.1 Number of Units and Channels

This microcontroller has the following number of TAUD units and channels.

Table 32.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)

Table 32.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)

Table 32.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)	TAUDn (n = 0)

Table 32.4 TAUDn Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name TAUDn	Number of Channels per Unit	RH850/F1KH-D8 176 Pins (16 ch)	RH850/F1KH-D8 233 Pins (16 ch)	RH850/F1KH-D8 324 Pins (16 ch)
TAUD0	16	✓	✓	✓

Table 32.5 TAUDn Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name TAUDn	Number of Channels per Unit	RH850/F1KM-S4 100 Pins (16 ch)	RH850/F1KM-S4 144 Pins (16 ch)	RH850/F1KM-S4 176 Pins (16 ch)	RH850/F1KM-S4 233 Pins (16 ch)	RH850/F1KM-S4 272 Pins (16 ch)
TAUD0	16	✓	✓	✓	✓	✓

Table 32.6 TAUDn Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name TAUDn	Number of Channels per Unit	RH850/F1KM-S1 48 Pins (16 ch)	RH850/F1KM-S1 64 Pins (16 ch)	RH850/F1KM-S1 80 Pins (16 ch)	RH850/F1KM-S1 100 Pins (16 ch)
TAUD0	16	✓	✓	✓	✓

Table 32.7 Indices (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual TAUD units are identified by the index “n”; for example, TAUDnTOM is the TAUDn channel output mode register.
m	The TAUD has 16 channels. Throughout this section, the individual channels are identified by the index “m”, thus a certain channel is denoted as CHm (m = 0 to 15). The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

32.1.2 Register Base Address

TAUDn base address is listed in the following table.

TAUDn register addresses are given as an offset from the base address.

Table 32.8 Register Base Address (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<TAUD0_base>	FFE2 0000_H

32.1.3 Clock Supply

The TAUDn clock supply is shown in the following table.

Table 32.9 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
TAUDn	PCLK	CKSCLK_IPERI1	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock

32.1.4 Interrupt Requests

TAUDn interrupt requests are listed in the following table.

Table 32.10 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAUD0			
INTTAUD0I0	Channel 0 interrupt	8, 132	0 to 3, 17, 18
INTTAUD0I1	Channel 1 interrupt	48	0 to 3, 17, 18
INTTAUD0I2	Channel 2 interrupt	9, 158	0 to 3, 17, 18
INTTAUD0I3	Channel 3 interrupt	49	0 to 3, 17, 18
INTTAUD0I4	Channel 4 interrupt	10, 133	0 to 3, 17, 18
INTTAUD0I5	Channel 5 interrupt	50	0 to 3, 17, 18
INTTAUD0I6	Channel 6 interrupt	11, 134	0 to 3, 17, 18
INTTAUD0I7	Channel 7 interrupt	51	0 to 3, 17, 18
INTTAUD0I8	Channel 8 interrupt	12, 135	0 to 3, 17, 18
INTTAUD0I9	Channel 9 interrupt	52	0 to 3, 17, 18
INTTAUD0I10	Channel 10 interrupt	13, 159	0 to 3, 17, 18
INTTAUD0I11	Channel 11 interrupt	53	0 to 3, 17, 18
INTTAUD0I12	Channel 12 interrupt	14, 160	0 to 3, 17, 18
INTTAUD0I13	Channel 13 interrupt	54	0 to 3, 17, 18
INTTAUD0I14	Channel 14 interrupt	15, 161	0 to 3, 17, 18
INTTAUD0I15	Channel 15 interrupt	55	0 to 3, 17, 18

32.1.5 Reset Sources

TAUDn reset sources are listed in the following table. TAUDn is initialized by these reset sources.

Table 32.11 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
TAUDn	All reset sources (ISORES)

32.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below.

Table 32.12 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUD0		
TAUDTTIN0	Channel 0 input*[1]	TAUD0I0
TAUDTTIN1	Channel 1 input*[1]	TAUD0I1
TAUDTTIN2	Channel 2 input*[1]	TAUD0I2
TAUDTTIN3	Channel 3 input*[1]	TAUD0I3
TAUDTTIN4	Channel 4 input*[1]	TAUD0I4
TAUDTTIN5	Channel 5 input*[1]	TAUD0I5
TAUDTTIN6	Channel 6 input*[1]	TAUD0I6
TAUDTTIN7	Channel 7 input*[1]	TAUD0I7
TAUDTTIN8	Channel 8 input*[1]	TAUD0I8
TAUDTTIN9	Channel 9 input*[1]	TAUD0I9
TAUDTTIN10	Channel 10 input*[1]	TAUD0I10
TAUDTTIN11	Channel 11 input*[1]	TAUD0I11
TAUDTTIN12	Channel 12 input*[1]	TAUD0I12
TAUDTTIN13	Channel 13 input*[1]	TAUD0I13
TAUDTTIN14	Channel 14 input*[1]	TAUD0I14
TAUDTTIN15	Channel 15 input*[1]	TAUD0I15
TAUDTTOUT0	Channel 0 output	TAUD0O0
TAUDTTOUT1	Channel 1 output	TAUD0O1
TAUDTTOUT2	Channel 2 output	TAUD0O2
TAUDTTOUT3	Channel 3 output	TAUD0O3
TAUDTTOUT4	Channel 4 output	TAUD0O4
TAUDTTOUT5	Channel 5 output	TAUD0O5
TAUDTTOUT6	Channel 6 output	TAUD0O6
TAUDTTOUT7	Channel 7 output	TAUD0O7
TAUDTTOUT8	Channel 8 output	TAUD0O8
TAUDTTOUT9	Channel 9 output	TAUD0O9
TAUDTTOUT10	Channel 10 output	TAUD0O10
TAUDTTOUT11	Channel 11 output	TAUD0O11
TAUDTTOUT12	Channel 12 output	TAUD0O12
TAUDTTOUT13	Channel 13 output	TAUD0O13
TAUDTTOUT14	Channel 14 output	TAUD0O14
TAUDTTOUT15	Channel 15 output	TAUD0O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector** and **Section 2C.12, Noise Filter & Edge/Level Detector**.

32.1.7 Internal Input/Output Signals

The internal input/output signals of TAUDn are listed below.

Table 32.13 Internal Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
TAUDnTSSTm	Simultaneous channel start trigger input	PIC
TAUDnTUDCm (m = 0, 2, 8)	TAUD master up/down signal output	PIC

32.1.8 TAUD0 Input Selection

The output from port TAUD0Im (m = 0 to 15) can be input to TAUDTTINm (m = 0 to 15) as shown in the following figure.

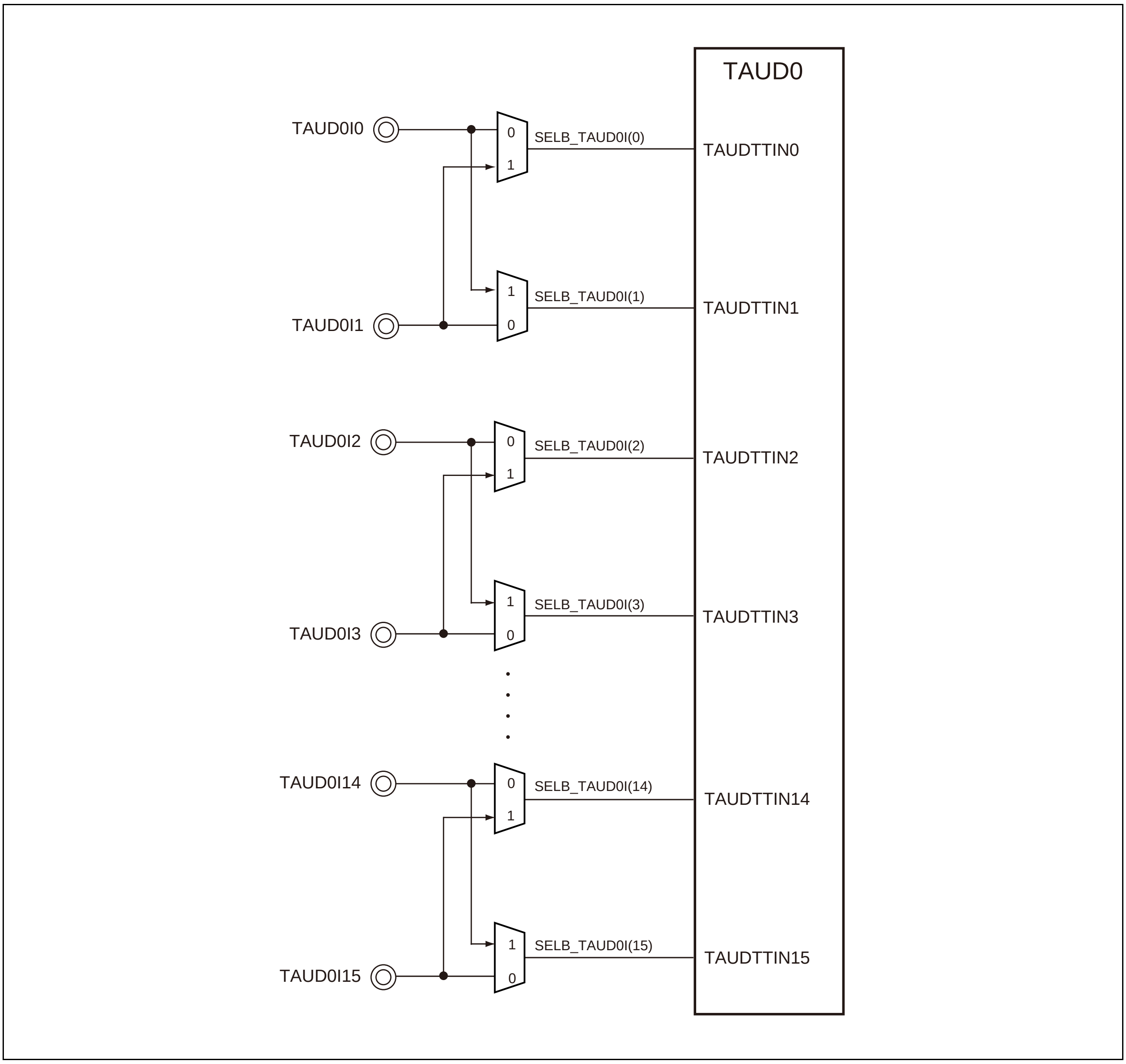


Figure 32.1 Selection of Signals Input to TAUD0

The following table shows the method of selecting input signals to several TAUD0 inputs.

Table 32.14 TAUD0 Input Selection (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Input Signal	Function	Settings
TAUDTTIN [m]	Port TAUD0I[m]	SELB_TAUD0I [m] = 0
	Port TAUD0I[m + 1]	SELB_TAUD0I [m] = 1
TAUDTTIN [m + 1]	Port TAUD0I[m + 1]	SELB_TAUD0I [m + 1] = 0
	Port TAUD0I[m]	SELB_TAUD0I [m + 1] = 1

32.1.8.1 List of Registers

Input signal selection register is listed in the following table.

Table 32.15 List of Register

Module Name	Register Name	Symbol	Address
TAUDn input selection register			
SL_TAUDn	TAUDTTINm input signal selection register	SELB_TAUD0I	FFE2 4000_H

32.1.8.2 SELB_TAUD0I — TAUDTTINm Input Signal Selection Register

This register selects the TAUDTTINm input signals.

Access: This register can be read or written in 16-bit units.

Address: FFE2 4000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SELB_TAUD0I15	SELB_TAUD0I14	SELB_TAUD0I13	SELB_TAUD0I12	SELB_TAUD0I11	SELB_TAUD0I10	SELB_TAUD0I9	SELB_TAUD0I8	SELB_TAUD0I7	SELB_TAUD0I6	SELB_TAUD0I5	SELB_TAUD0I4	SELB_TAUD0I3	SELB_TAUD0I2	SELB_TAUD0I1	SELB_TAUD0I0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.16 SELB_TAUD0I Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>SELB_TAUD0Im</td><td>Selection of TAUDTTINm input signal
<table>
<tr><th>TAUD Input</th><th>Bit [m+1]</th><th>Bit [m]</th><th>Input signal</th></tr>
<tr><td>TAUDTTIN[m]</td><td>x</td><td>0</td><td>Selection of port TAUD0I[m]</td></tr>
<tr><td></td><td>x</td><td>1</td><td>Selection of port TAUD0I[m + 1]</td></tr>
<tr><td>TAUDTTIN[m+1]</td><td>0</td><td>x</td><td>Selection of port TAUD0I[m + 1]</td></tr>
<tr><td></td><td>1</td><td>x</td><td>Selection of port TAUD0I[m]</td></tr>
</table>
(m = 0, 2, 4, 6, 8, 10, 12, 14)</td></tr>
</table>

CAUTION

Do not change the input signal of each channel during the timer counting.

32.2 Overview

32.2.1 Functional Overview

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

32.2.2 Terms

In this section, the following terms are used.

Independent / synchronous channel operation

Independent or synchronous channel operation describes the dependency of channels on each other:

- If a channel operates independently of all other channels, this is called independent channel operation.
- If a channel operates depending on other channels, this is called synchronous channel operation.

Channel group

In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Operation mode

An operation mode can be selected for every channel m. The operation mode defines the basic operation and features of a channel.

In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

Channel output mode

The channel output mode defines the operation of TAUDTTOUTm

- of a single channel (independent output operation) or
- of all channels in a channel group (synchronous output operation).

Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

Channel operation function

The channel operation function defines the complete function and all features

- of a single channel (independent channel operation) or
- of all channels in a channel group (synchronous channel operation).

Upper / lower channel

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

32.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 32.17 Functional List of TAUD Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 32.12**
Interval Timer Function	**Section 32.12.1**
TAUDTTINm Input Interval Timer Function	**Section 32.12.2**
Clock Divide Function	**Section 32.12.3**
External Event Count Function	**Section 32.12.4**
Delay Count Function	**Section 32.12.5**
One-Pulse Output Function	**Section 32.12.6**
TAUDTTINm Input Pulse Interval Measurement Function	**Section 32.12.7**
TAUDTTINm Input Signal Width Measurement Function	**Section 32.12.8**
TAUDTTINm Input Position Detection Function	**Section 32.12.9**
TAUDTTINm Input Period Count Detection Function	**Section 32.12.10**
TAUDTTINm Input Pulse Interval Judgment Function	**Section 32.12.11**
TAUDTTINm Input Signal Width Judgment Function	**Section 32.12.12**
Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)	**Section 32.12.13**
Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)	**Section 32.12.14**
One-Phase PWM Output Function	**Section 32.12.15**
Independent Channel Real-Time Functions	**Section 32.13**
Real-Time Output Function Type 1	**Section 32.13.1**
Real-Time Output Function Type 2	**Section 32.13.2**
Independent Channel Simultaneous Rewrite Functions	**Section 32.14**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 32.14.1**
Simultaneous Rewrite Trigger Generation Function Type 2	**Section 32.14.2**
Synchronous Channel Operation Functions	**Section 32.15**
PWM Output Function	**Section 32.15.1**
One-Shot Pulse Output Function	**Section 32.15.2**
Trigger Start PWM Output Function	**Section 32.15.3**
Delay Pulse Output Function	**Section 32.15.4**
Offset Trigger Output Function	**Section 32.15.5**
A/D Conversion Trigger Output Function Type 1	**Section 32.15.6**
Triangle PWM Output Function	**Section 32.15.7**
Triangle PWM Output Function with Dead Time	**Section 32.15.8**
A/D Conversion Trigger Output Function Type 2	**Section 32.15.9**
Interrupt Request Signals Culling Function	**Section 32.15.10**
Synchronous Non-Complementary and Complementary Modulation Output Functions	**Section 32.16**
Non-Complementary Modulation Output Function Type 1	**Section 32.16.1**
Non-Complementary Modulation Output Function Type 2	**Section 32.16.2**
Complementary Modulation Output Function	**Section 32.16.3**

32.2.4 TAUD I/O and Interrupt Request Signals

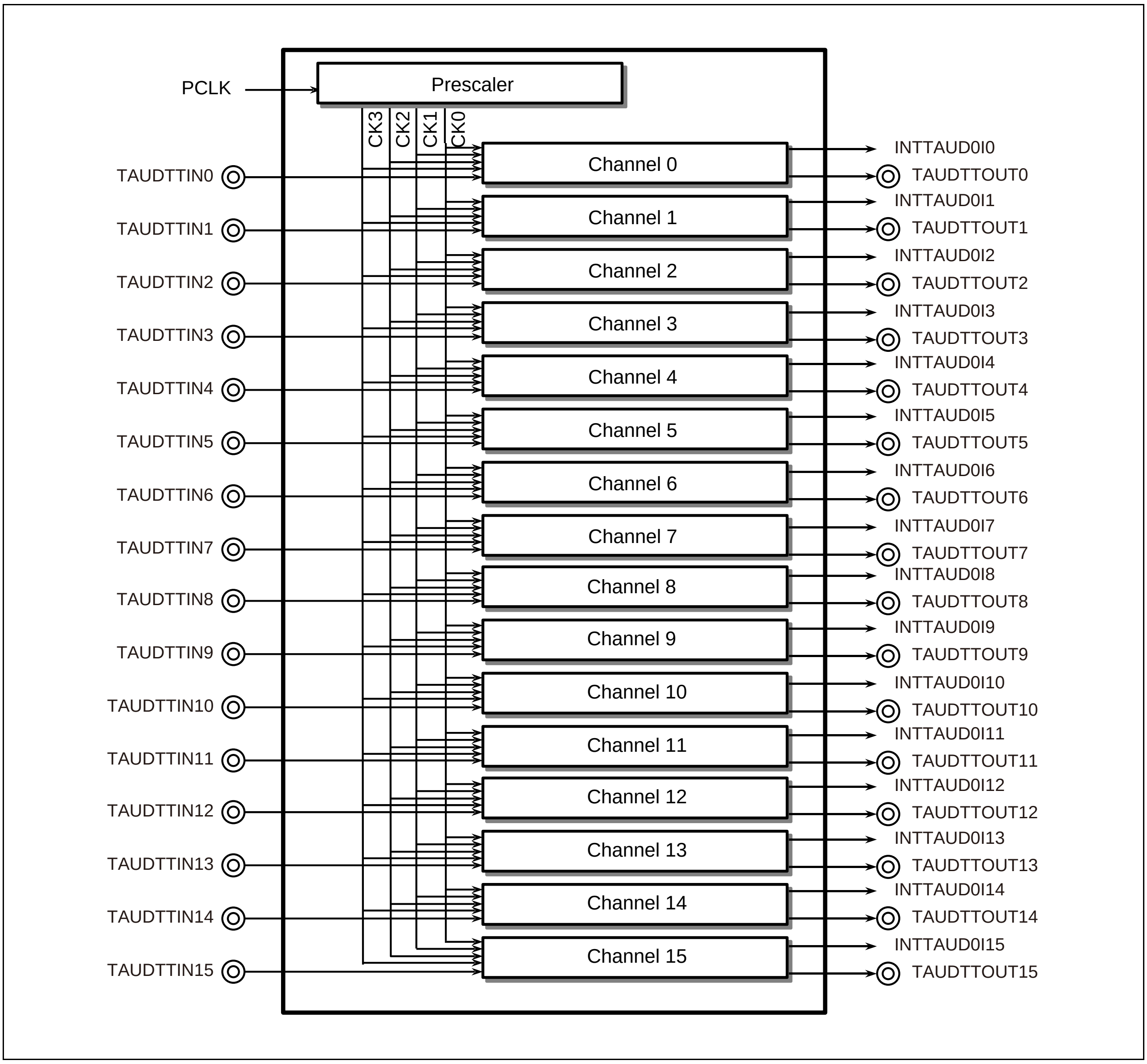


Figure 32.2 TAUD I/O and Interrupt Request Signals

32.2.5 Block Diagram

Figure 32.3, Block Diagram of the TAUD shows the main components of the TAUD.

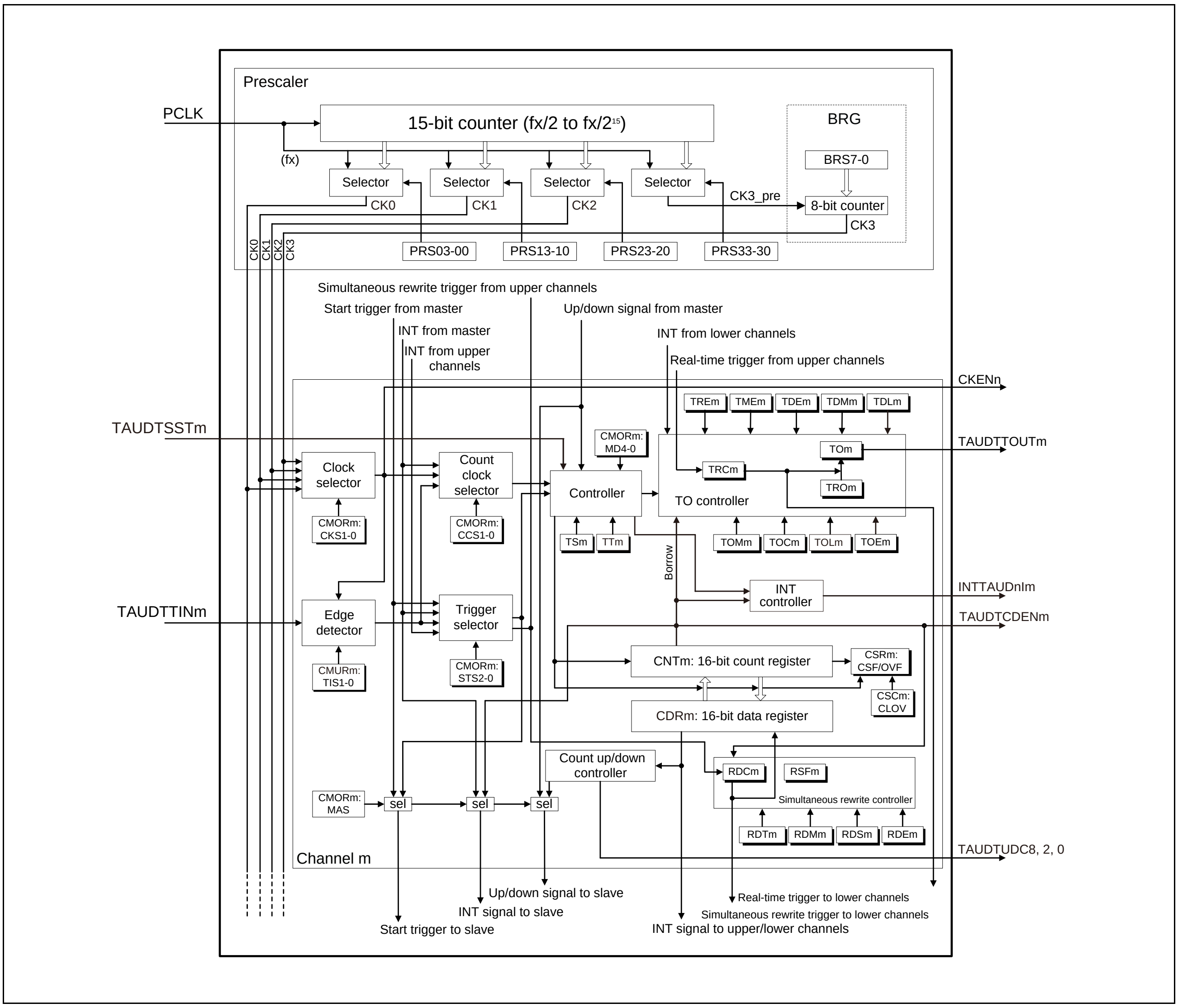


Figure 32.3 Block Diagram of the TAUD

The module name “TAUDn” has been omitted from the register names for the sake of clarity in the above figure.

32.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to four clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth count clock CK3 can be adjusted more precisely by using BRG to set an additional division factor that is not a power of 2.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUDnIm from master channel
- TAUDTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDnCMORm.TAUDnMD[4:0])
- Counter start enable (TAUDnTS.TAUDnTSm) and counter stop (TAUDnTT.TAUDnTTm)
  When counter start is enabled, status flag TAUDnTE.TAUDnTEm is set.
- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUDnTE.TAUDnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUDnTSSTm
- TAUDTTINm input signal valid edge
- INTTAUDnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUDTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a function that can be used in synchronous operating modes. The data registers (TAUDnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUDnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

32.3 Registers

32.3.1 List of Registers

TAUD registers are listed in the following table.

For details about <TAUDn_base>, see **Section 32.1.2, Register Base Address**.

Table 32.18 List of Registers

Module Name	Register Name	Symbol	Address
TAUDn prescaler registers			
TAUDn	TAUDn prescaler clock select register	TAUDnTPS	<TAUDn_base> + 240_H
	TAUDn prescaler baud rate setting register	TAUDnBRS	<TAUDn_base> + 244_H
TAUDn control registers			
TAUDn	TAUDn channel data register	TAUDnCDRm	<TAUDn_base> + m × 4_H
	TAUDn channel counter register	TAUDnCNTm	<TAUDn_base> + 80_H + m × 4_H
	TAUDn channel mode OS register	TAUDnCMORm	<TAUDn_base> + 200_H + m × 4_H
	TAUDn channel mode user register	TAUDnCMURm	<TAUDn_base> + $C0_H$ + m × 4_H
	TAUDn channel status register	TAUDnCSRm	<TAUDn_base> + 140_H + m × 4_H
	TAUDn channel status clear trigger register	TAUDnCSCm	<TAUDn_base> + 180_H + m × 4_H
	TAUDn channel start trigger register	TAUDnTS	<TAUDn_base> + $1C4_H$
	TAUDn channel enable status register	TAUDnTE	<TAUDn_base> + $1C0_H$
	TAUDn channel stop trigger register	TAUDnTT	<TAUDn_base> + $1C8_H$
TAUDn output registers			
TAUDn	TAUDn channel output enable register	TAUDnTOE	<TAUDn_base> + $5C_H$
	TAUDn channel output register	TAUDnTO	<TAUDn_base> + 58_H
	TAUDn channel output mode register	TAUDnTOM	<TAUDn_base> + 248_H
	TAUDn channel output configuration register	TAUDnTOC	<TAUDn_base> + $24C_H$
	TAUDn channel output active level register	TAUDnTOL	<TAUDn_base> + 040_H
	TAUDn channel dead time output enable register	TAUDnTDE	<TAUDn_base> + 250_H
	TAUDn channel dead time output mode register	TAUDnTDM	<TAUDn_base> + 254_H
	TAUDn channel dead time output level register	TAUDnTDL	<TAUDn_base> + 54_H
	TAUDn channel real-time output register	TAUDnTRO	<TAUDn_base> + $4C_H$
	TAUDn channel real-time output enable register	TAUDnTRE	<TAUDn_base> + 258_H
	TAUDn channel real-time output control register	TAUDnTRC	<TAUDn_base> + $25C_H$
	TAUDn channel modulation output enable register	TAUDnTME	<TAUDn_base> + 50_H
TAUDn reload data registers			
TAUDn	TAUDn channel reload data enable register	TAUDnRDE	<TAUDn_base> + 260_H
	TAUDn channel reload data mode register	TAUDnRDM	<TAUDn_base> + 264_H
	TAUDn channel reload data control channel select register	TAUDnRDS	<TAUDn_base> + 268_H
	TAUDn channel reload data control register	TAUDnRDC	<TAUDn_base> + $26C_H$
	TAUDn channel reload data trigger register	TAUDnRDT	<TAUDn_base> + 44_H
	TAUDn channel reload status register	TAUDnRSF	<TAUDn_base> + 48_H
TAUDn emulation register			
TAUDn	TAUDn emulation register	TAUDnEMU	<TAUDn_base> + 290_H

32.3.2 Details of TAUDn Prescaler Registers

32.3.2.1 TAUDnTPS — TAUDn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3_PRE by the factor specified in TAUDnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 240_H

Value after reset: $FFFF_H$

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="4">TAUDnPRS3
[3:0]</td><td colspan="4">TAUDnPRS2
[3:0]</td><td colspan="4">TAUDnPRS1
[3:0]</td><td colspan="4">TAUDnPRS0
[3:0]</td></tr>
<tr><td>Value after reset</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>

Table 32.19 TAUDnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 12</td><td>TAUDnPRS3 [3:0]</td><td>Specifies CK3_PRE clock.
CK3_PRE clock is an input clock to BRG unit which supplies the CK3 operation clock to all channels.
<table>
<tr><th>TAUDnPRS3[3:0]</th><th>CK3_PRE Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK3 are stopped (TAUDnTE.TAUDnTEm = 0).</td></tr>
</table>

Table 32.19 TAUDnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUDnPRS2 [3:0]</td><td>Specifies the CK2 clock.
<table>
<tr><th>TAUDnPRS2[3:0]</th><th>CK2 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUDnTE.TAUDnTEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUDnPRS1 [3:0]</td><td>Specifies the CK1 clock.
<table>
<tr><th>TAUDnPRS1[3:0]</th><th>CK1 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUDnTE.TAUDnTEm = 0).</td></tr>
</table>

Table 32.19 TAUDnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th colspan="2">Function</th></tr>
<tr><td rowspan="19">3 to 0</td><td rowspan="19">TAUDnPRS0 [3:0]</td><td colspan="2">Specifies the CK0 clock.</td></tr>
<tr><td>TAUDnPRS0[3:0]</td><td>CK0 Clock</td></tr>
<tr><td>0000_B</td><td>$PCLK/2^0$</td></tr>
<tr><td>0001_B</td><td>$PCLK/2^1$</td></tr>
<tr><td>0010_B</td><td>$PCLK/2^2$</td></tr>
<tr><td>0011_B</td><td>$PCLK/2^3$</td></tr>
<tr><td>0100_B</td><td>$PCLK/2^4$</td></tr>
<tr><td>0101_B</td><td>$PCLK/2^5$</td></tr>
<tr><td>0110_B</td><td>$PCLK/2^6$</td></tr>
<tr><td>0111_B</td><td>$PCLK/2^7$</td></tr>
<tr><td>1000_B</td><td>$PCLK/2^8$</td></tr>
<tr><td>1001_B</td><td>$PCLK/2^9$</td></tr>
<tr><td>1010_B</td><td>$PCLK/2^{10}$</td></tr>
<tr><td>1011_B</td><td>$PCLK/2^{11}$</td></tr>
<tr><td>1100_B</td><td>$PCLK/2^{12}$</td></tr>
<tr><td>1101_B</td><td>$PCLK/2^{13}$</td></tr>
<tr><td>1110_B</td><td>$PCLK/2^{14}$</td></tr>
<tr><td>1111_B</td><td>$PCLK/2^{15}$</td></tr>
<tr><td colspan="2">The above bits are rewritable only when all the counters using CK0 are stopped (TAUDnTE.TAUDnTEm = 0).</td></tr>
</table>

NOTE

The TAUDn clock input PCLK is specified in the first part of this section, **Section 32.1.3, Clock Supply**.

32.3.2.2 TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUDnTPS.TAUDnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + 244_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUDnBRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.20 TAUDnBRS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 0</td><td>TAUDnBRS[7:0]</td><td>Specifies a CK3_PRE clock division factor for generating CK3.
<table>
<tr><th>TAUDnBRS[7:0]</th><th>CK3 Clock</th></tr>
<tr><td>0000 0000$_B$</td><td>CK3_PRE / 1</td></tr>
<tr><td>0000 0001$_B$</td><td>CK3_PRE / 2</td></tr>
<tr><td>0000 0010$_B$</td><td>CK3_PRE / 3</td></tr>
<tr><td>0000 0011$_B$</td><td>CK3_PRE / 4</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111 1110$_B$</td><td>CK3_PRE / 255</td></tr>
<tr><td>1111 1111$_B$</td><td>CK3_PRE / 256</td></tr>
</table></td></tr>
</table>

32.3.3 Details of TAUDn Control Registers

32.3.3.1 TAUDnCDRm — TAUDn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDnCMORm.TAUDnMD[4:1].

Access: This register can be read or written in 16-bit units.

- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUDn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.21 TAUDnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCDR [15:0]	Data register for capture/compare values

32.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register

This is a channel m counter register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 80$_H$ + m × 4$_H$

Value after reset: FFFF$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.22 TAUDnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCNT [15:0]	16-bit counter value

A read value depends on a counter value, a changed operating mode, TAUDnTS.TAUDnTSm or TAUDnTT.TAUDnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUDnTT.TAUDnTTm = 1)

Table 32.23, TAUDnCNTm Read Values after Re-Enabling Counter lists the initial counter read values after the counter is stopped (TAUDnTE.TAUDnTEm = 0) and re-enabled (TAUDnTS.TAUDnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUDnTS.TAUDnTSm = 1) with the counter waiting for a start trigger.

Table 32.23 TAUDnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUDnCNTm Value		
		Start Value*[1]	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF$_H$	Stop value	—
Judge mode	Count down	FFFF$_H$	Stop value	—
Capture mode	Count up	0000$_H$	Stop value	—
Event count mode	Count down	FFFF$_H$	Stop value	—
One-count mode	Count down	FFFF$_H$	Stop value	FFFF$_H$
Capture and one-count mode	Count up	0000$_H$	Stop value	Capture value + 1 (TAUDnCDRm)
Judge and one-count mode	Count down	FFFF$_H$	Stop value	TAUDnCNTm value – 1
Count-up/-down mode	Count down/up	FFFF$_H$	Stop value	—
Pulse one-count mode	Count down	FFFF$_H$	Stop value	0000$_H$
Count capture mode	Count up	0000$_H$	Stop value	—
Gate count mode	Count down	FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000$_H$	Stop value	Stop value

Note 1. The value set for TAUDnCNTm when the operating mode is changed after a reset is deasserted.

32.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. Writable only when the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.24 TAUDnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>TAUDnCKS[1:0]</td><td>Selects an operation clock.
An operation clock is used for the TAUDTTINm input edge detection circuit.
Setting of TAUDnCMORm.TAUDnCCS[1:0] bits also allows the operation clock to serve as the TAUDnCNTm counter clock.
<table>
<tr><th>TAUDnCKS1</th><th>TAUDnCKS0</th><th>Selection of Operation Clock</th></tr>
<tr><td>0</td><td>0</td><td>CK0</td></tr>
<tr><td>0</td><td>1</td><td>CK1</td></tr>
<tr><td>1</td><td>0</td><td>CK2</td></tr>
<tr><td>1</td><td>1</td><td>CK3</td></tr>
</table></td></tr>
<tr><td>13, 12</td><td>TAUDnCCS[1:0]</td><td>Selects a count clock for TAUDnCNTm counter.
<table>
<tr><th>TAUDnCCS1</th><th>TAUDnCCS0</th><th>Selection of Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]</td></tr>
<tr><td>0</td><td>1</td><td>Valid edge of TAUDTTINm input signal</td></tr>
<tr><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>INTTAUDnIm signal of master channel</td></tr>
</table></td></tr>
<tr><td>11</td><td>TAUDnMAS</td><td>Specifies whether the channel is a master channel or slave channel during synchronous channel operation.
0: Slave
1: Master
This bit setting is valid only for even channels (CHm_even). Odd channels (CHm_odd) are fixed to 0.</td></tr>
</table>

Table 32.24 TAUDnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUDnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUDnSTS2</th><th>TAUDnSTS1</th><th>TAUDnSTS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUDTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUDTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Triggers simultaneous rewrite.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTT AUDnIm is the start trigger of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTT AUDnIm of upper channel (m − 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead time output signal of TAUDTTOUTm generating unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of master channel</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUDnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUDnCDRm and overflow flag TAUDnCSRm.TAUDnOVF of channel m.
These bits are valid only when channel m is in capture function.
The bits must be fixed to 01_B in Capture mode and Capture and gate count mode.
<table>
<tr><th>TAUDn COS1</th><th>TAUDn COS0</th><th>TAUDnCDRm</th><th>TAUDnCSRm.TAUDnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a valid edge of TAUDTTINm input.</td><td>Updated (cleared or set) upon detection of a TAUDTTINm input valid edge:
● If a counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is set.
● If no counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUDTTINm input and at the occurrence of counter overflow:
● Detection of valid edge of TAUDTTINm input: Counter value is written into TAUDnCDRm.
● Occurrence of overflow: $FFFF_H$ is loaded into TAUDnCDRm. The next detection of valid edge of TAUDTTINm input is ignored.</td><td>Not set</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 32.24 TAUDnCMORm Register Contents

Bit Position	Bit Name	Function
4 to 0	TAUDnMD[4:0]	Specifies an operating mode.

TAUDn MD4	TAUDn D3	TAUDn MD2	TAUDn MD1	TAUDn MD0	Functional Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Judge mode
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Event count mode
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Judge and one-count mode
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Count-up/-down mode
1	0	1	0	1/0	Pulse one-count mode
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUDnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.
Event count mode Count-up/-down mode	This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).
One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION:** ● INTTAUDnIm signal is not output at the beginning of count operation in one-count mode. ● INTTAUDnIm signal is output at the beginning of count operation in pulse onecount mode.
Gate count mode	This bit should be set to 0 (disables start trigger detection during counting).
Capture and one-count mode Capture and gate count mode	This bit should be set to 0. **CAUTION:** INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.
Judge mode Judge and one-count mode	Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm

32.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUDTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + C0$_{H}$ + m × 4$_{H}$

Value after reset: 00$_{H}$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.25 TAUDnCMURm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 2</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>1, 0</td><td>TAUDnTIS[1:0]</td><td>Specifies a valid edge of TAUDTTINm input signal.
<table>
<tr><th>TAUDnTIS1</th><th>TAUDnTIS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>Detection of rising and falling edges (selects low width measurement)
Start trigger: Falling edge
Stop trigger (capture): Rising edge</td></tr>
<tr><td>1</td><td>1</td><td>Detection of rising and falling edges (selects high width measurement)
Start trigger: Rising edge
Stop trigger (capture): Falling edge</td></tr>
</table>
● Edge detection of TAUDTTINm input signal is based on the operation clock selected by TAUDnCMORm.TAUDnCKS[1:0].</td></tr>
</table>

32.3.3.5 TAUDnCSRm — TAUDn Channel Status Register

This register indicates the count direction and overflow status of channel m counter.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUDn_base> + 140_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnCSF	TAUDnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.26 TAUDnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUDnCSF	Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in the following mode: ● Count-up/-down mode
0	TAUDnOVF	Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the following modes: ● Capture mode ● Capture and one-count mode The function of this bit depends on the setting of control bit TAUDnCMORm.TAUDnCOS[1:0].

32.3.3.6 TAUDnCSCm — TAUDn Channel Status Clear Trigger Register

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUDn_base> + 180_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.27 TAUDnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUDnCLOV	0: No function 1: Clears overflow flag TAUDnCSRm.TAUDnOVF.

32.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + $1C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TS15	TAUDn TS14	TAUDn TS13	TAUDn TS12	TAUDn TS11	TAUDn TS10	TAUDn TS09	TAUDn TS08	TAUDn TS07	TAUDn TS06	TAUDn TS05	TAUDn TS04	TAUDn TS03	TAUDn TS02	TAUDn TS01	TAUDn TS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 32.28 TAUDnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTSm	Enables the counter operation of channel m. 0: No function 1: Enables the counter operation and sets TAUDnTE.TAUDnTEm to 1. The counter operation is only enabled when TAUDnTE.TAUDnTEm is set to 1. Whether counting is started or not depends on a selected operating mode.

32.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register enables/disables a counter operation.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 1C0$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TE15	TAUDn TE14	TAUDn TE13	TAUDn TE12	TAUDn TE11	TAUDn TE10	TAUDn TE09	TAUDn TE08	TAUDn TE07	TAUDn TE06	TAUDn TE05	TAUDn TE04	TAUDn TE03	TAUDn TE02	TAUDn TE01	TAUDn TE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.29 TAUDnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTEm	Enables/disables the counter operation of channel m. 0: Disables counter operation. 1: Enables counter operation. This bit is set to 1 when trigger input of TAUDnTSSTm (synchronous channel start trigger signal) is detected or when TAUDnTS.TAUDnTSm is set to 1. This bit is set to 0 when TAUDnTT.TAUDnTTm is set to 1.

32.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000$_{H}$.

Address: <TAUDn_base> + 1C8$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TT15	TAUDn TT14	TAUDn TT13	TAUDn TT12	TAUDn TT11	TAUDn TT10	TAUDn TT09	TAUDn TT08	TAUDn TT07	TAUDn TT06	TAUDn TT05	TAUDn TT04	TAUDn TT03	TAUDn TT02	TAUDn TT01	TAUDn TT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 32.30 TAUDnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTTm	Stops the counter operation of channel m. 0: No function 1: Stops the counter operation and resets TAUDnTE.TAUDnTEm. TAUDnCNTm, TAUDnTO.TAUDnTOm, and TAUDTTOUTm retain the values provided before the counter is stopped.

32.3.4 Details of TAUDn Simultaneous Rewrite Registers

32.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDE15	TAUDn RDE14	TAUDn RDE13	TAUDn RDE12	TAUDn RDE11	TAUDn RDE10	TAUDn RDE09	TAUDn RDE08	TAUDn RDE07	TAUDn RDE06	TAUDn RDE05	TAUDn RDE04	TAUDn RDE03	TAUDn RDE02	TAUDn RDE01	TAUDn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.31 TAUDnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enables simultaneous rewrite

32.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 268_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDS15	TAUDn RDS14	TAUDn RDS13	TAUDn RDS12	TAUDn RDS11	TAUDn RDS10	TAUDn RDS09	TAUDn RDS08	TAUDn RDS07	TAUDn RDS06	TAUDn RDS05	TAUDn RDS04	TAUDn RDS03	TAUDn RDS02	TAUDn RDS01	TAUDn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.32 TAUDnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDSm	Selects a channel that controls a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

32.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 264$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDM15	TAUDn RDM14	TAUDn RDM13	TAUDn RDM12	TAUDn RDM11	TAUDn RDM10	TAUDn RDM09	TAUDn RDM08	TAUDn RDM07	TAUDn RDM06	TAUDn RDM05	TAUDn RDM04	TAUDn RDM03	TAUDn RDM02	TAUDn RDM01	TAUDn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.33 TAUDnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 0.

32.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnIm signal to trigger simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 26C$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDC15	TAUDn RDC14	TAUDn RDC13	TAUDn RDC12	TAUDn RDC11	TAUDn RDC10	TAUDn RDC09	TAUDn RDC08	TAUDn RDC07	TAUDn RDC06	TAUDn RDC05	TAUDn RDC04	TAUDn RDC03	TAUDn RDC02	TAUDn RDC01	TAUDn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.34 TAUDnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 1.

32.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDT15	TAUDn RDT14	TAUDn RDT13	TAUDn RDT12	TAUDn RDT11	TAUDn RDT10	TAUDn RDT09	TAUDn RDT08	TAUDn RDT07	TAUDn RDT06	TAUDn RDT05	TAUDn RDT04	TAUDn RDT03	TAUDn RDT02	TAUDn RDT01	TAUDn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 32.35 TAUDnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDTm	Triggers a simultaneous rewrite enabling state. 0: No function 1: The simultaneous rewrite enabling flag (TAUDnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger. These bits only apply when: ● TAUDnRDE.TAUDnRDEm = 1

32.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RSF15	TAUDn RSF14	TAUDn RSF13	TAUDn RSF12	TAUDn RSF11	TAUDn RSF10	TAUDn RSF09	TAUDn RSF08	TAUDn RSF07	TAUDn RSF06	TAUDn RSF05	TAUDn RSF04	TAUDn RSF03	TAUDn RSF02	TAUDn RSF01	TAUDn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.36 TAUDnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRSFm	Indicates simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUDnRDTm = 1).

32.3.5 Details of TAUDn Output Registers

32.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 5C_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOE15	TAUDn TOE14	TAUDn TOE13	TAUDn TOE12	TAUDn TOE11	TAUDn TOE10	TAUDn TOE09	TAUDn TOE08	TAUDn TOE07	TAUDn TOE06	TAUDn TOE05	TAUDn TOE04	TAUDn TOE03	TAUDn TOE02	TAUDn TOE01	TAUDn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.37 TAUDnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOEm	Enables/disables the independent channel output function. 0: Disables the independent timer output function (controlled by software). 1: Enables the independent timer output function.

32.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDTTOUTm level.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TO15	TAUDn TO14	TAUDn TO13	TAUDn TO12	TAUDn TO11	TAUDn TO10	TAUDn TO09	TAUDn TO08	TAUDn TO07	TAUDn TO06	TAUDn TO05	TAUDn TO04	TAUDn TO03	TAUDn TO02	TAUDn TO01	TAUDn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.38 TAUDnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOm	Specifies and reads a TAUDTTOUTm level. 0: Low level 1: High level Only TAUDnTOm bits for which Independent Channel Output function is disabled (TAUDnTOEm = 0) can be written.

32.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 248$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOM15	TAUDn TOM14	TAUDn TOM13	TAUDn TOM12	TAUDn TOM11	TAUDn TOM10	TAUDn TOM09	TAUDn TOM08	TAUDn TOM07	TAUDn TOM06	TAUDn TOM05	TAUDn TOM04	TAUDn TOM03	TAUDn TOM02	TAUDn TOM01	TAUDn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.39 TAUDnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

32.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOMm.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 24C$_{H}$

Value after reset: 0000$_{H}$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOC15	TAUDn TOC14	TAUDn TOC13	TAUDn TOC12	TAUDn TOC11	TAUDn TOC10	TAUDn TOC09	TAUDn TOC08	TAUDn TOC07	TAUDn TOC06	TAUDn TOC05	TAUDn TOC04	TAUDn TOC03	TAUDn TOC02	TAUDn TOC01	TAUDn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.40 TAUDnTOC Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>TAUDnTOCm</td><td>Specifies an output mode.
0: Operating mode 1
1: Operating mode 2
As listed below, the output mode depends on the setting of TAUDnTOM.TAUDnTOMm.
<table>
<tr><th>TAUDnTOMm</th><th>TAUDnTOCm</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>Toggle mode: Toggle operation is conducted when INTTAUDnIm occurs.</td></tr>
<tr><td>0</td><td>1</td><td>Set/reset mode: Set when INTTAUDnIm occurs at the beginning of count operation, and reset when INTTAUDnIm is caused by detection of a match between TAUDnCNTm and TAUDnCDRm.</td></tr>
<tr><td>1</td><td>0</td><td>Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels.</td></tr>
<tr><td>1</td><td>1</td><td>Synchronous channel operating mode 2: Set when INTTAUDnIm occurs in count-down status, and reset when INTTAUDnIm occurs in count-up status.</td></tr>
</table>
</td></tr>
</table>

32.3.5.5 TAUDnTOL — TAUDn Channel Output Active Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOm).

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOL15	TAUDn TOL14	TAUDn TOL13	TAUDn TOL12	TAUDn TOL11	TAUDn TOL10	TAUDn TOL09	TAUDn TOL08	TAUDn TOL07	TAUDn TOL06	TAUDn TOL05	TAUDn TOL04	TAUDn TOL03	TAUDn TOL02	TAUDn TOL01	TAUDn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.41 TAUDnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOLm	Specifies the output logic of channel m output bit (TAUDnTO.TAUDnTOm). 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1 and independent channel output mode 1 with real-time output.

32.3.6 Details of TAUDn Dead Time Output Registers

32.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDE15	TAUDn TDE14	TAUDn TDE13	TAUDn TDE12	TAUDn TDE11	TAUDn TDE10	TAUDn TDE09	TAUDn TDE08	TAUDn TDE07	TAUDn TDE06	TAUDn TDE05	TAUDn TDE04	TAUDn TDE03	TAUDn TDE02	TAUDn TDE01	TAUDn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.42 TAUDnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDEm	Enables/disables the dead time control operation of channel m. 0: Disables dead time operation 1: Enables dead time operation. The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm = 1

32.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 254_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDM15	TAUDn TDM14	TAUDn TDM13	TAUDn TDM12	TAUDn TDM11	TAUDn TDM10	TAUDn TDM09	TAUDn TDM08	TAUDn TDM07	TAUDn TDM06	TAUDn TDM05	TAUDn TDM04	TAUDn TDM03	TAUDn TDM02	TAUDn TDM01	TAUDn TDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.43 TAUDnTDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDMm	Specifies the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even channel (duty dead time output). 1: When detecting the TIN input edge of a lower odd channel (one-phase dead time output). The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

32.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access:	This register can be read or written in 16-bit units.
Address:	<TAUDn_base> + 54_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDL15	TAUDn TDL14	TAUDn TDL13	TAUDn TDL12	TAUDn TDL11	TAUDn TDL10	TAUDn TDL09	TAUDn TDL08	TAUDn TDL07	TAUDn TDL06	TAUDn TDL05	TAUDn TDL04	TAUDn TDL03	TAUDn TDL02	TAUDn TDL01	TAUDn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.44 TAUDnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDLm	Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when: ● TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

32.3.7 Details of TAUDn Real-time/Modulation Output Registers

32.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 258_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRE15	TAUDn TRE14	TAUDn TRE13	TAUDn TRE12	TAUDn TRE11	TAUDn TRE10	TAUDn TRE09	TAUDn TRE08	TAUDn TRE07	TAUDn TRE06	TAUDn TRE05	TAUDn TRE04	TAUDn TRE03	TAUDn TRE02	TAUDn TRE01	TAUDn TRE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.45 TAUDnTRE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTREm	Enables or disables real-time output of channel m. 0: Disables real-time output 1: Enables real-time output. These bit settings are applied only when TAUDnTOE.TAUDnTOEm = 1. When TAUDnTRE.TAUDnTREm = 0, TAUDTTOUTm is not affected by real-time output. When TAUDnTRE.TAUDnTREm = 1, TAUDTTOUTm outputs the value of real-time output bit TAUDnTRO.TAUDnTROm in response to a timer operation.

32.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + $25C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRC15	TAUDn TRC14	TAUDn TRC13	TAUDn TRC12	TAUDn TRC11	TAUDn TRC10	TAUDn TRC09	TAUDn TRC08	TAUDn TRC07	TAUDn TRC06	TAUDn TRC05	TAUDn TRC04	TAUDn TRC03	TAUDn TRC02	TAUDn TRC01	TAUDn TRC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.46 TAUDnTRC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTRCm	Specifies a channel on which the real-time output trigger for channel m is generated. 0: Next upper channel with this bit set to 1 1: Channel m These bit settings are applied only when TAUDnTRE.TAUDnTREm = 1.

32.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDTTOUTm.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 04C$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRO15	TAUDn TRO14	TAUDn TRO13	TAUDn TRO12	TAUDn TRO11	TAUDn TRO10	TAUDn TRO09	TAUDn TRO08	TAUDn TRO07	TAUDn TRO06	TAUDn TRO05	TAUDn TRO04	TAUDn TRO03	TAUDn TRO02	TAUDn TRO01	TAUDn TRO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.47 TAUDnTRO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTROm	Sets a value which is output to TAUDTTOUTm. 0: Low 1: High TAUDnTROm value is not output to TAUDTTOUTm when TAUDnTRE.TAUDnTREm = 0, even if a real-time output trigger occurs.

32.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 050$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TME15	TAUDn TME14	TAUDn TME13	TAUDn TME12	TAUDn TME11	TAUDn TME10	TAUDn TME09	TAUDn TME08	TAUDn TME07	TAUDn TME06	TAUDn TME05	TAUDn TME04	TAUDn TME03	TAUDn TME02	TAUDn TME01	TAUDn TME00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.48 TAUDnTME Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTMEm	Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation 1: Enables modulation These bit settings are applied only when TAUDnTOE.TAUDnTOEm and TAUDnTRE.TAUDnTREm = 1.

32.3.8 TAUDn Emulation Register

32.3.8.1 TAUDnEMU — TAUDn Emulation Register

This register controls SVSTOP operations.

Access:	This register can be read or written in 8-bit units. Perform write operations when the counter is stopped (TAUDnTE.TAUDnTEm = 0) and (EPC.SVSTOP = 0).
Address:	<TAUDn_base> + 290_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	TAUDnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 32.49 TAUDnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUDnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

32.4 Operating Procedure

The following lists the general operation procedure for the TAUDn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUDTTOUTm is also initialized and outputs a low level.

(1) Set the TAUDnTPS and TAUDnBRS registers to specify the clock frequency of CK0 to CK3.

(2) Configure the desired TAUDn function:

- Set the operation mode
- Set the channel output mode
- Set any other control bits

(3) Enable the counter by setting the TAUDnTS.TAUDnTSm bit to 1.
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.

(4) If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUDnTT.TAUDnTTm bit to 1. The counter can be forcibly restarted by setting the TAUDnTS.TAUDnTSm bit to 1.

(5) Stop the function by setting the TAUDnTT.TAUDnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 32.12, Independent Channel Operation Functions** and **Section 32.15, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

32.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). Several rules apply to the settings of channels. These rules are detailed in **Section 32.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 32.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 32.6, Simultaneous Rewrite**

32.5.1 Rules of Synchronous Channel Operation

Number of master and slave channels

- Only even channels (CH0, CH2, CH4, …) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the master channel and the synchronized slave channel. This is achieved by setting the TAUDnCMORm.TAUDnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 32.4, Grouping of Channels and Assignment of Operatiion Clocks**.

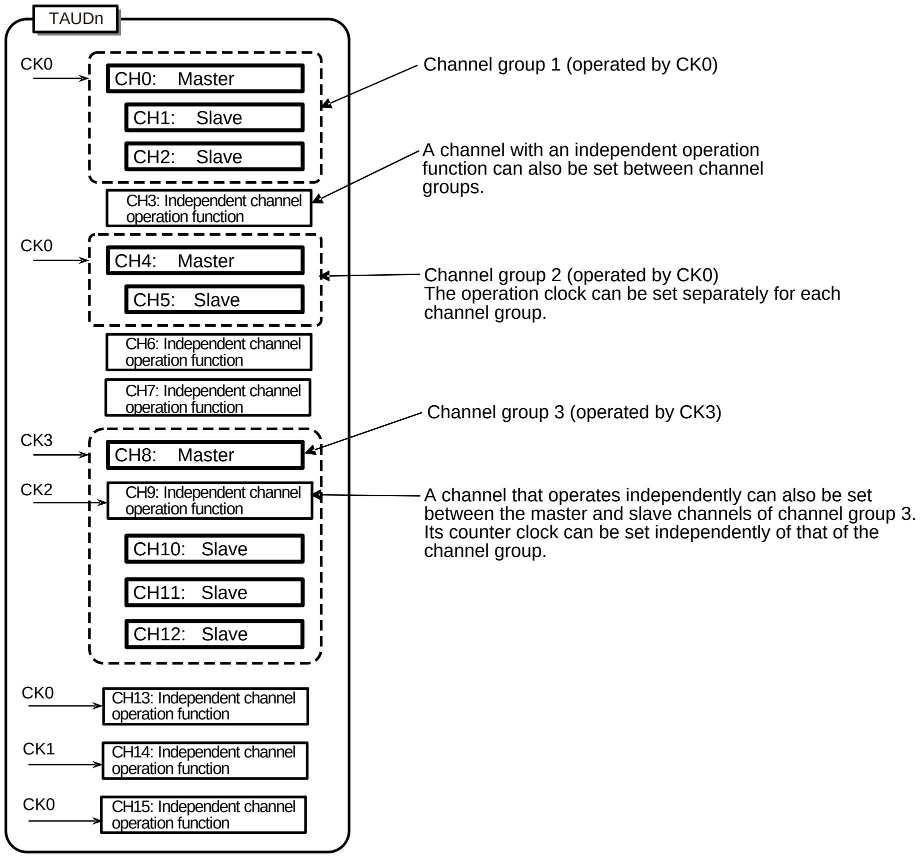


Figure 32.4 Grouping of Channels and Assignment of Operatiion Clocks

Control trigger signal for master/slave channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals for their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

32.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

32.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDnTS.TAUDnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDnTT.TAUDnTTm bits of the channels should be set at the same time.

Setting to the TAUDnTS.TAUDnTSm bits to 1 also sets the corresponding TAUDnTE.TAUDnTEm bits to 1, enabling counting. The count start timing depends on operating mode.

32.5.2.2 Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to perform simultaneous start between the units, see **Section 36.8, Simultaneous Start Trigger Function**.

32.6 Simultaneous Rewrite

32.6.1 Overview of Operations

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUDnCDRm and TAUDnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDnIm being issued on the upper channel specified by TAUDnRDC.TAUDnRDCm

There are four methods for simultaneous rewrite. These are listed in **Table 32.50, Simultaneous Rewrite Methods and when They are Triggered**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 32.50 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUDnRDE. TAUDnRDEm	TAUDnRDS. TAUDnRDSm	TAUDnRDM. TAUDnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular cycle of the corresponding slave channel.	1	0	1
C1	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm	1	1	0/1
C2	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal	1	1	0/1

Table 32.51, Channel Functions and the Methods They Use for Simultaneous Rewrite lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 32.14 Independent Channel Simultaneous Rewrite Functions**, **Section 32.15, Synchronous Channel Operation Functions**, and **Section 32.16, Synchronous Non-Complementary and Complementary Modulation Output Functions**.

Table 32.51 Channel Functions and the Methods They Use for Simultaneous Rewrite

Function	A	B	C1	C2	TAUDnTOL.TAUDnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			✓		
PWM Output Function	✓		✓		✓
One-Shot Pulse Output Function	✓				
Trigger Start PWM Output Function	✓			✓	
Delay Pulse Output Function	✓				
Triangle PWM Output Function		✓	✓		✓
Triangle PWM Output Function with Dead Time		✓	✓		
Interrupt Request Signals Culling Function	✓	✓	✓		
AD Conversion Trigger Output Function Type 1	✓		✓		
AD Conversion Trigger Output Function Type 2		✓	✓		
Non-Complementary Modulation Output Function Type 1	✓		✓		
Non-Complementary Modulation Output Function Type 2		✓	✓		
Complementary Modulation Output Function		✓	✓		

Note: ✓: Available, (Blank): Unavailable

32.6.2 How to Control Simultaneous Rewrite

Figure 32.5, General Procedure for Simultaneous Rewrite shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

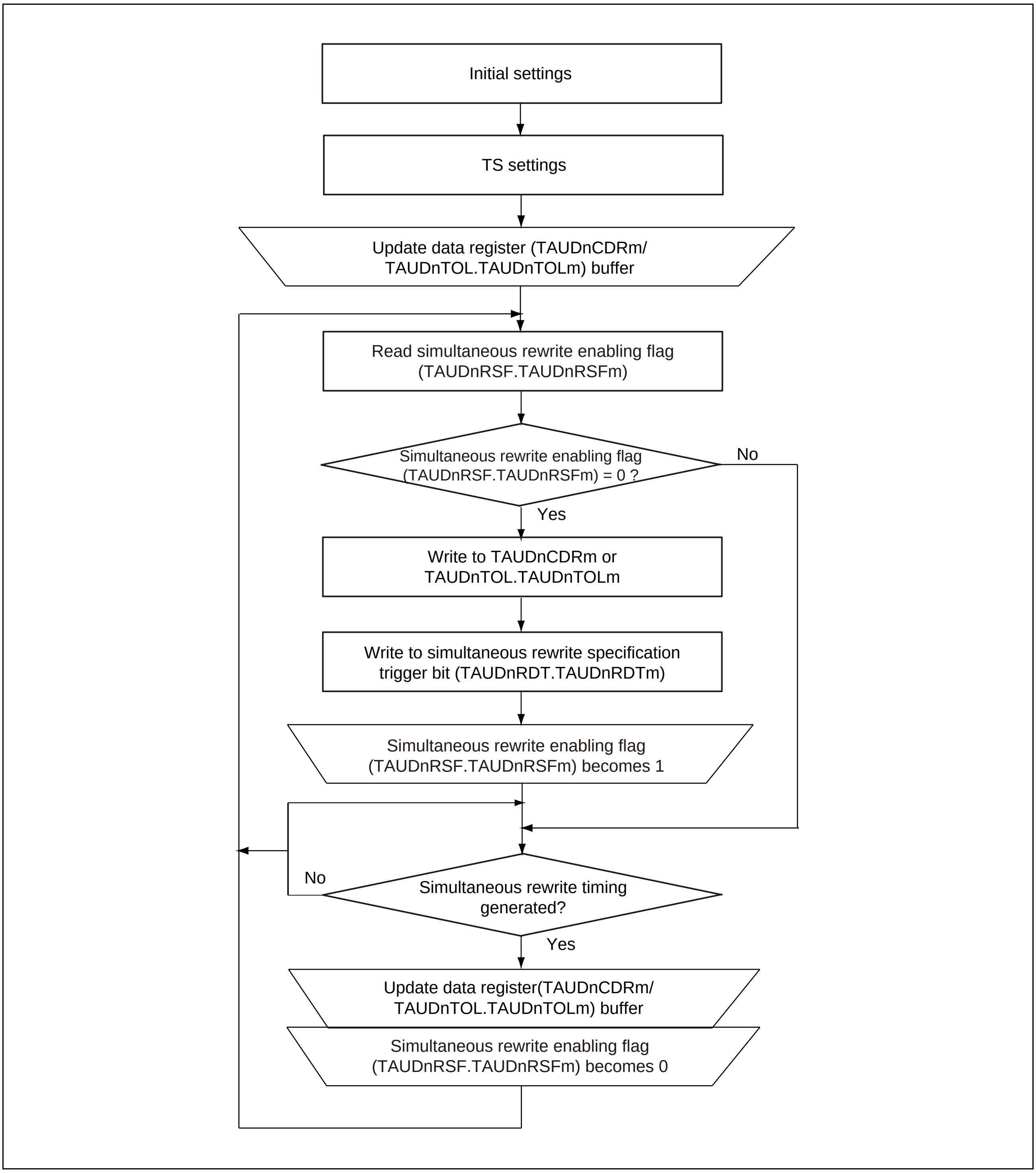


Figure 32.5 General Procedure for Simultaneous Rewrite

32.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUDnRDE.TAUDnRDEm = 1
- To select the type of simultaneous rewrite, set TAUDnRDM.TAUDnRDMm and TAUDnRDS.TAUDnRDSm according to the values listed in **Table 32.50, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using TAUDnRDC.TAUDnRDCm. (Prerequisite: TAUDnRDS.TAUDnRDSm has been set to the upper channel.)

32.6.2.2 Start Counter and Count Operation

- To start all the TAUDnCNTm counters of the channel group, set the corresponding TAUDnTS.TAUDnTSm bits to 1. The values of TAUDnTOL.TAUDnTOLm and the data registers (TAUDnCDRm) are loaded into the corresponding TAUDnTOL.TAUDnTOLm buffer (TAUDnTOL.TAUDnTOLm buf) and data buffer registers (TAUDnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1 sets the reload flag (TAUDnRSF.TAUDnRSFm) to 1, enabling simultaneous rewrite. TAUDnRSF.TAUDnRSFm remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the TAUDnRSF.TAUDnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

32.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- When simultaneous rewrite is complete, the TAUDnRSF.TAUDnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

32.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUDnRDE.TAUDnRDEm, TAUDnRDS.TAUDnRDSm, TAUDnRDM.TAUDnRDMm, and TAUDnRDC.TAUDnRDCm cannot be changed while the counter is in operation (TAUDnTE.TAUDnTEm = 1).
- TAUDnTOL.TAUDnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUDnTOL.TAUDnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUDTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUDnRDS.TAUDnRDSm = 1), the TAUDnRDC.TAUDnRDCm bit controls all the lower channels. This means that if the TAUDnRDC.TAUDnRDCm bits of CH2 and CH7 are set to 1 and the TAUDnRDC.TAUDnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger (TAUDnRDE.TAUDnRDEm and TAUDnRDS.TAUDnRDSm = 1) but no upper channel is set (TAUDnRDC.TAUDnRDC[15:0] = 0), simultaneous rewrite cannot take place.

32.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

32.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

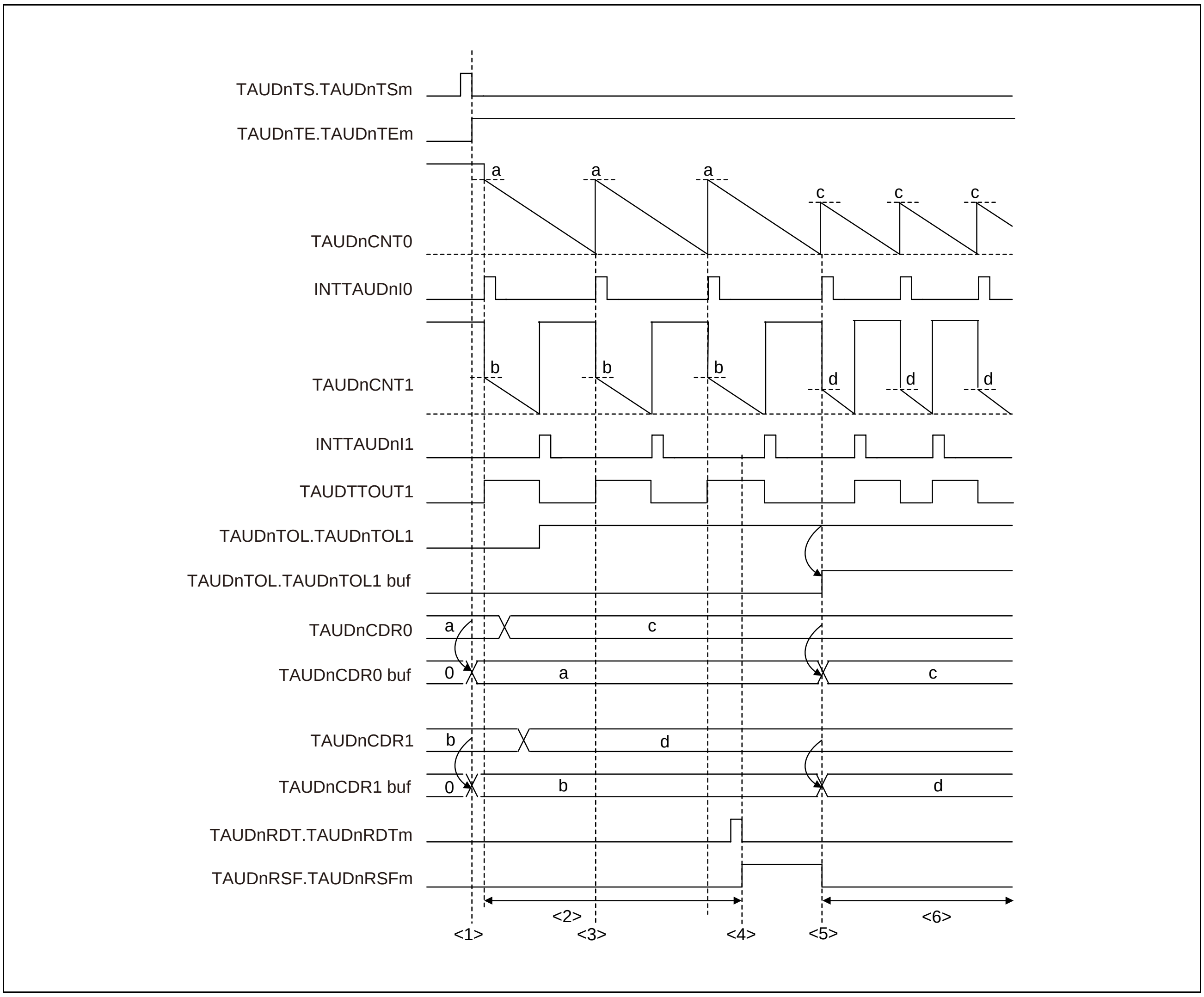


Figure 32.6 Simultaneous Rewrite when the Master Channel (Re)starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer and TAUDnTOL.TAUDnTOLm value is copied to the TAUDnTOL.TAUDnTOLm buffer.

(2) The TAUDnCDRm and TAUDnTOL.TAUDnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0)

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer and the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

32.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

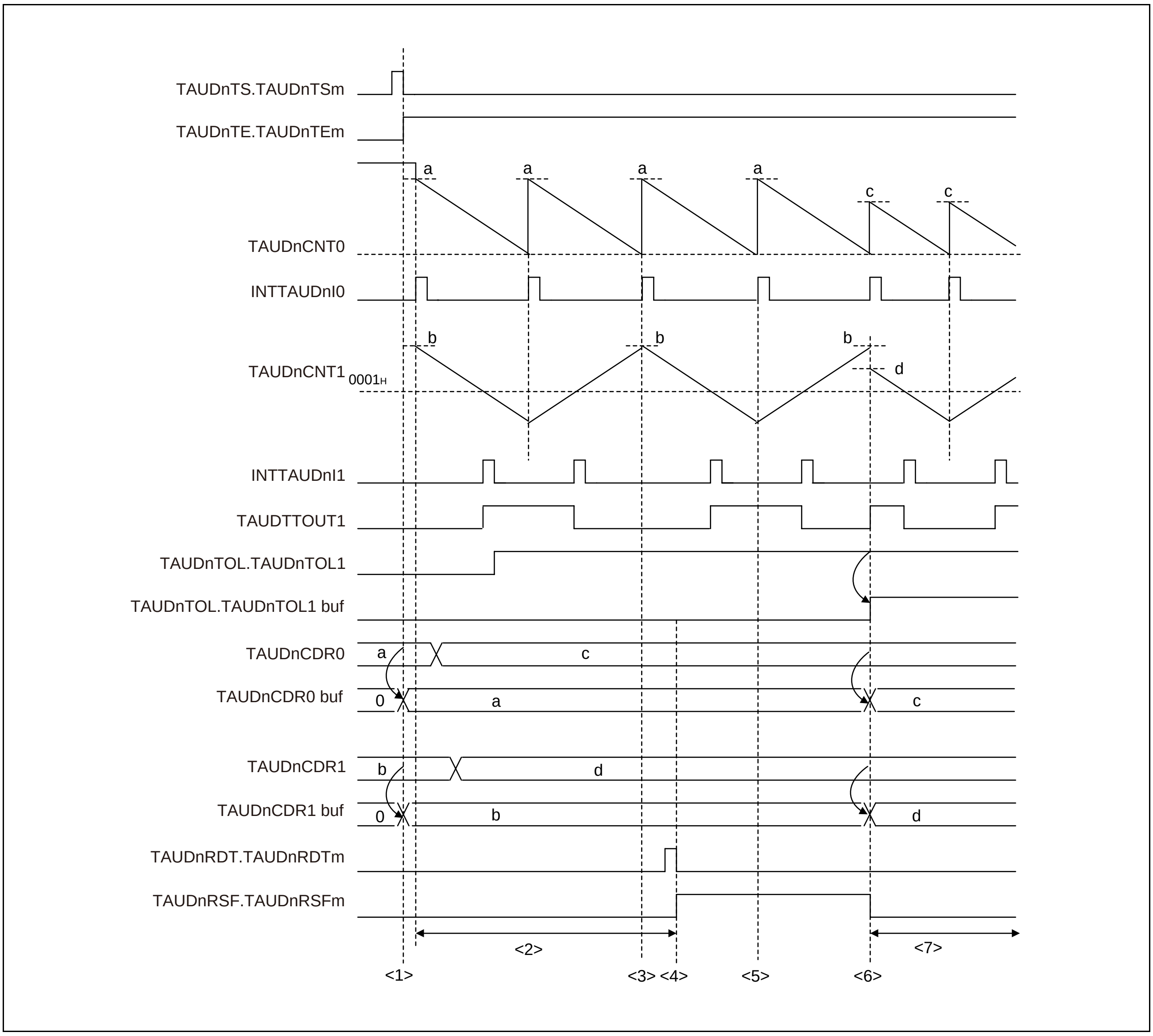


Figure 32.7 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

Setting:

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm and TAUDnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the top of the triangular cycle. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer, the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

32.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

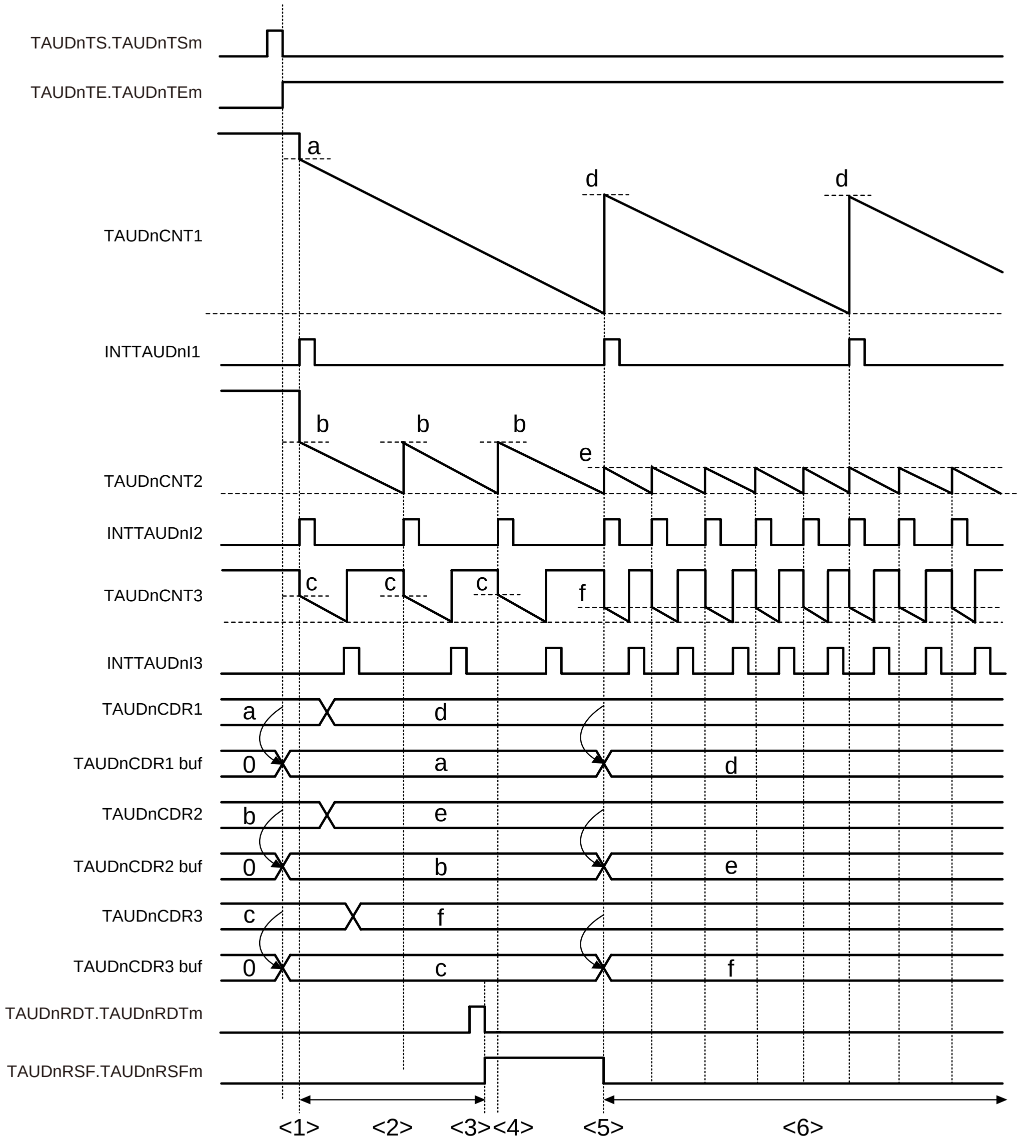


Figure 32.8 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

Setting:

CH1 is an upper channel which performs counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUDnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUDnCDRm values are loaded into the corresponding TAUDnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be rechanged.

32.6.4.4 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2)

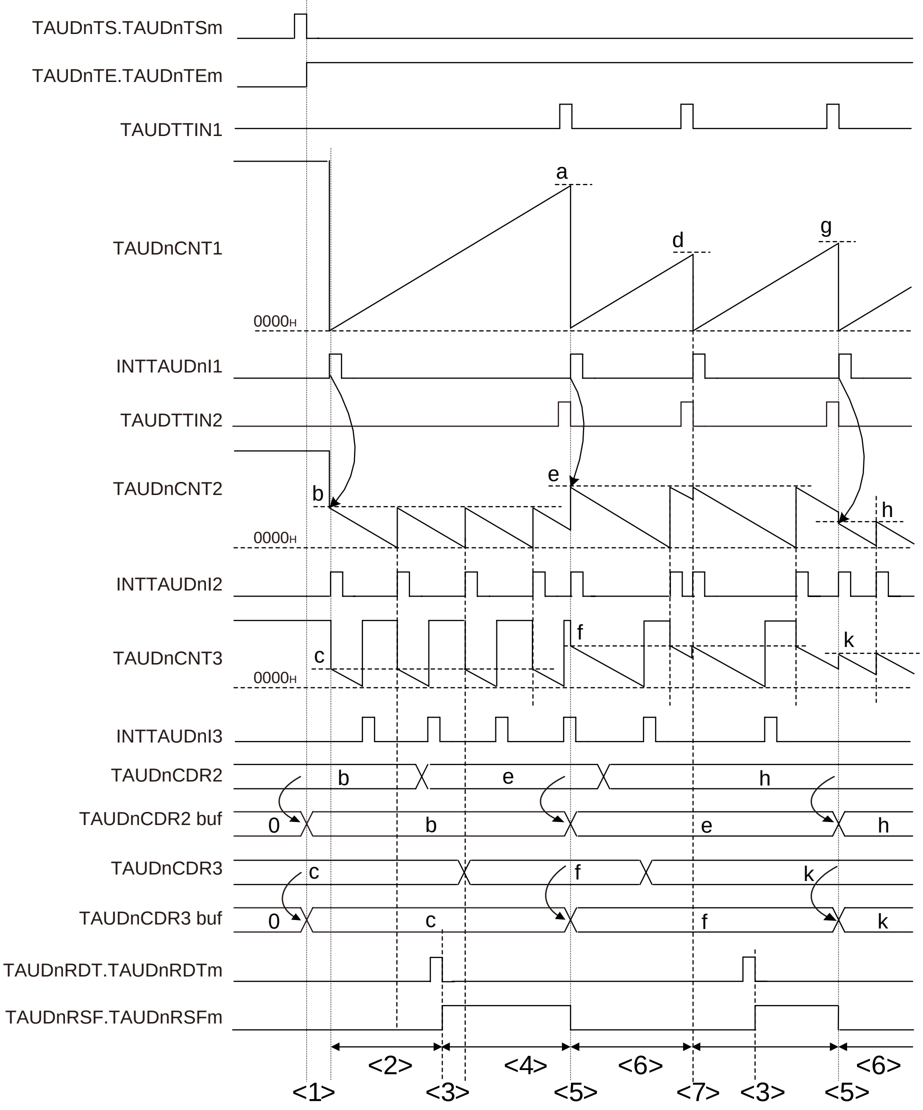


Figure 32.9 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal

Setting:

CH1 is an upper channel which performs counting up, CH2 is a master channel, and CH3 is the slave channel. The synchronous channel operation method C2 is applied. The TAUDnRDC register specifies which upper channel is monitored for an INTTAUDnIm trigger.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer. However, as TAUDnCDR1 operates in capture mode, TAUDnCDR1 value is not copied to the TAUDnCDR1 buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is caused by external signal TIN1. The TAUDnCDRm values are written to the corresponding TAUDnCDRm buffers.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.

(7) An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

32.7 Channel Output Modes

The output of the TAUDTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUDnTOE.TAUDnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUDnTO.TAUDnTOm) is sent to the output pin (TAUDTTOUTm).
- By TAUD signals (TAUDnTOE.TAUDnTOEm = 1)
  When controlled by TAUD signals, the output level of TAUDTTOUTm is set or reset or toggled by internal signals. The value of TAUDnTO.TAUDnTOm is updated accordingly to reflect the value of TAUDTTOUTm.
  - Independently (TAUDnTOM.TAUDnTOMm = 0)
    In case of independent operation, the output of the TAUDTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUDnTOM.TAUDnTOMm = 0).
  - Synchronously (TAUDnTOM.TAUDnTOMm = 1)
    In case of synchronous operation, the output of the TAUDTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUDnTOM.TAUDnTOMm = 1).

The TAUDnTO.TAUDnTOm bit can always be read to determine the current value of TAUDTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 32.52, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 32.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 32.7.3, Channel Output Modes Controlled Synchronously by TAUDn Signals**

Batch operation of TAUDnTOm bit

Whether a set value is reflected to the TAUDnTOm bit or not is controlled by the TAUDnTOE.TAUDnTOEm bit.

The TAUDnTOm setting is written only to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 0 when a write to the TAUDnTO register is attempted. No TAUDnTOm setting is reflected to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 1.

NOTE

TAUDnTO.TAUDnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUDnTOL.TAUDnTOLm.

The value of TAUDnTOL.TAUDnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUDnTOL.TAUDnTOLm result in an undefined TAUDTTOUTm signal output.

See **Section 32.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 32.52, Channel Output Modes**.

Table 32.52 Channel Output Modes

Channel Output Mode	TAUDn TOE. TAUDn TOEm	TAUDn TOM. TAUDn TOMm	TAUDn TOC. TAUDn TOCm	TAUDn TDE. TAUDn TDEm	TAUDn TRE. TAUDn TREm	TAUDn TME. TAUDn TMEm	TAUDn TDM. TAUDn TDMm
By software							
Independent channel output mode controlled by software	0	X					
By TAUD signals, independently							
Independent channel output mode 1	1	0	0	0	0	0	0
with real-time output					1		
Independent channel output mode 2			1		0		
By TAUD signals, synchronously							
Synchronous channel output mode 1	1	1	0	0	0	0	0
with non-complementary modulation output					1	X	
Synchronous channel output mode 2			1	0	0	0	0
with dead time output				1			
with one-phase PWM output							1
with complementary modulation output					1	1	0
with non-complementary modulation output			1	0			

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTES

1. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1):
   - TAUDnTOM.TAUDnTOMm
   - TAUDnTOC.TAUDnTOCm
   - TAUDnTDE.TAUDnTDEm
   - TAUDnTRE.TAUDnTREm
   - TAUDnTDM.TAUDnTDMm
2. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1) except in channel output modes with modulation output:
   - TAUDnTME.TAUDnTMEm
   - TAUDnTDL.TAUDnTDLm

32.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUDTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUDnTOE.TAUDnTOEm = 0).

(1) Set TAUDnTO.TAUDnTOm to specify the initial level of the TAUDTTOUTm output.

(2) Set channel output mode according to **Table 32.52, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.

(3) Start the counter (TAUDnTS.TAUDnTSm = 1).

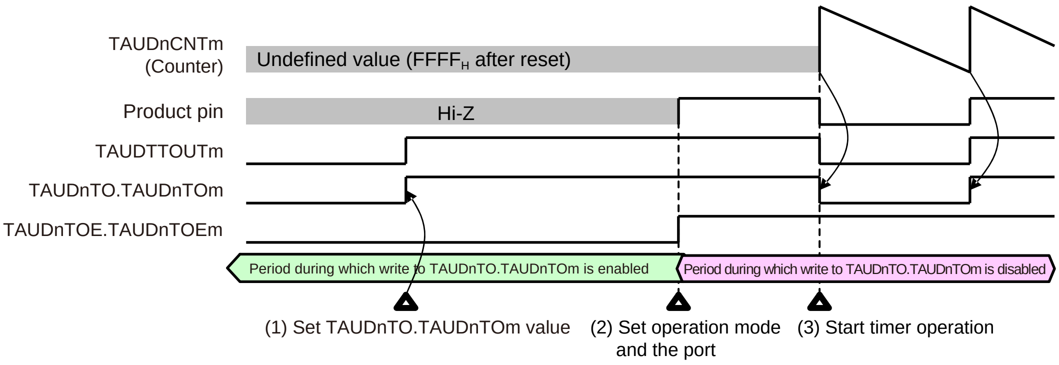


Figure 32.10 General Procedure for Specifying a TAUDTTOUTm Channel Output Mode

32.7.2 Channel Output Modes Controlled Independently by TAUDn Signals

This section lists the channel output modes that are controlled independently by TAUDn signals. The control bits used to specify a mode are listed in **Table 32.52, Channel Output Modes**.

32.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUDTTOUTm toggles when INTTAUDnIm is detected. The value of TAUDnTOL.TAUDnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 32.52, Channel Output Modes**.

32.7.2.2 Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDnTRO.TAUDnTROm bit of the trigger channel is output to TAUDTTOUTm. The trigger channel is specified by setting the corresponding TAUDnTRC.TAUDnTRCm bit to 1. It controls all lower channels for which TAUDnTRC.TAUDnTRCm = 0.

Set/reset conditions

The value of TAUDnTRO.TAUDnTROm bit is sent to TAUDTTOUTm only when an INTTAUDnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUDTTINm input edge/counter start

The type of trigger is set using the TAUDnCMORm.TAUDnMD[4:1] bits.

Prerequisites

Both the master and slave channels can be set as a trigger generation channel. A channel for which TAUDnTRC.TAUDnTRCm is set to 1 serves as a trigger generation channel regardless of the value of TAUDnTRE.TAUDnTREm.

If there is no channel for which TAUDnTRC.TAUDnTRCm is set to 1 or if TAUDnTRC.TAUDnTRC0 = 0, real-time output cannot take place.

This can be seen in **Figure 32.11, Real-Time Output**.

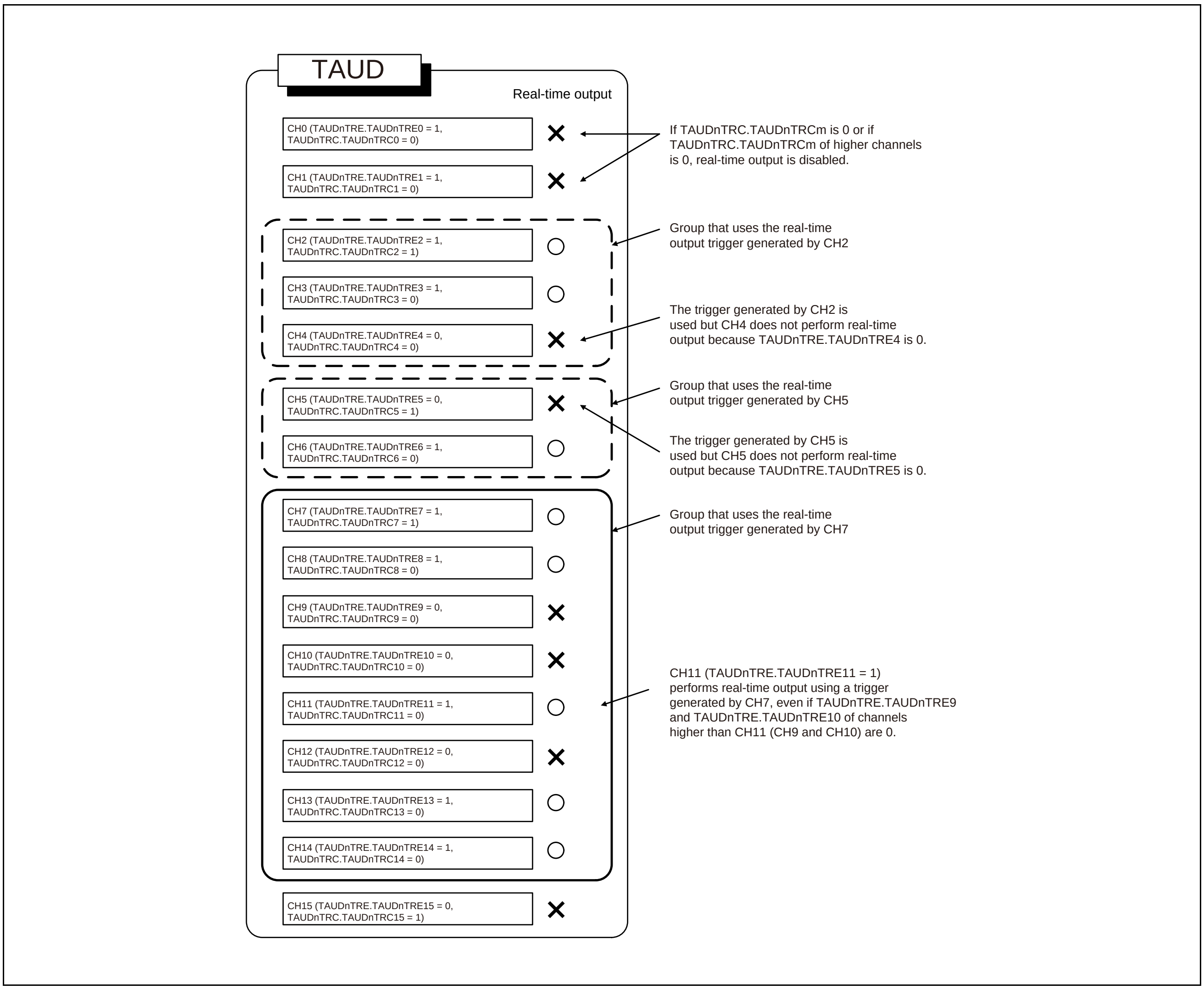


Figure 32.11 Real-Time Output

32.7.2.3 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUDTTOUTm is set when INTTAUDnIm occurs at the time of count start, and reset when INTTAUDnIm occurs due to a match between TAUDnCNTm and TAUDnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 32.52, Channel Output Modes**.

32.7.3 Channel Output Modes Controlled Synchronously by TAUDn Signals

This section lists the channel output modes that are controlled synchronously by TAUDn signals. The control bits used to specify a mode are listed in **Table 32.52, Channel Output Modes**.

32.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUDnIm of master channel serves as a set signal and INTTAUDnIm of the slave channel as a reset signal. If INTTAUDnIm of master channel and INTTAUDnIm of the slave channel are generated at the same time, INTTAUDnIm of the slave channel (reset signal) has priority over INTTAUDnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 32.52, Channel Output Modes**.

32.7.3.2 Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUDnTRO.TAUDnTROm) of a channel.

The phase period to which the dead time is added is specified using the TAUDnTDL.TAUDnTDLm bit; for positive phase set TAUDnTDL.TAUDnTDLm = 0 and for negative phase set TAUDnTDL.TAUDnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed, its new value is applied upon detection of INTTAUDnIm on the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on the master channel.

32.7.3.3 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM output at TAUDTTOUTm. For details, see **Section 32.15.7, Triangle PWM Output Function**.

Set/reset conditions

TAUDnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUDTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUDTTOUTm should be set to 0 before the function starts.

32.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 32.12, Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output**.

Set/reset conditions

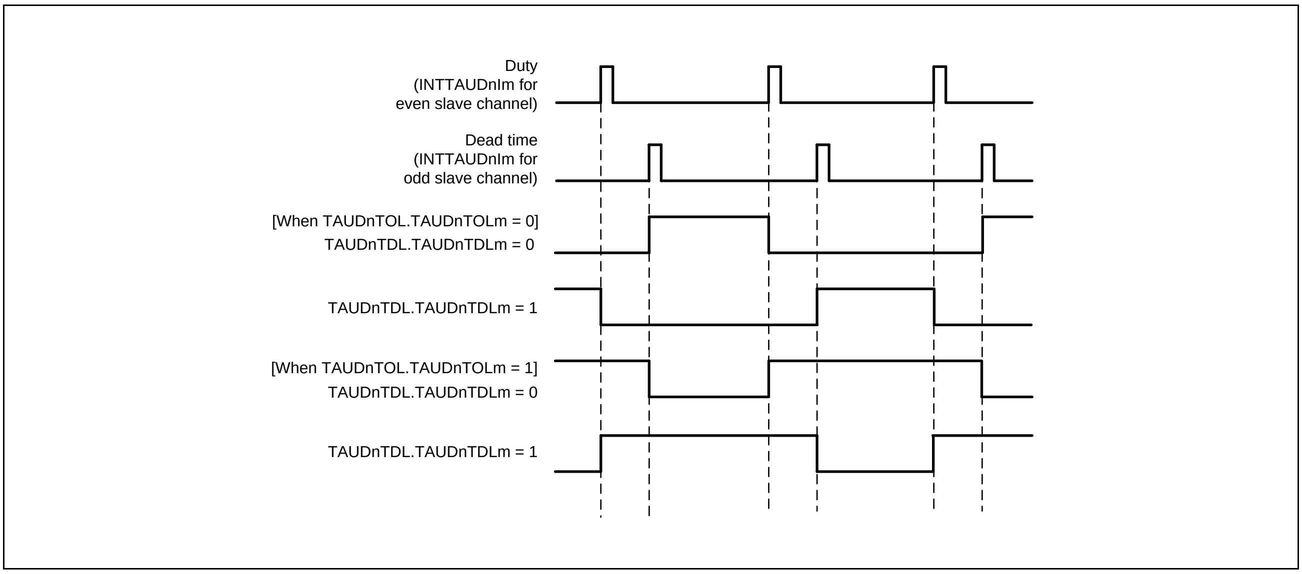


Figure 32.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm

- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

32.7.3.5 Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 32.13, Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output**.

Set/reset conditions

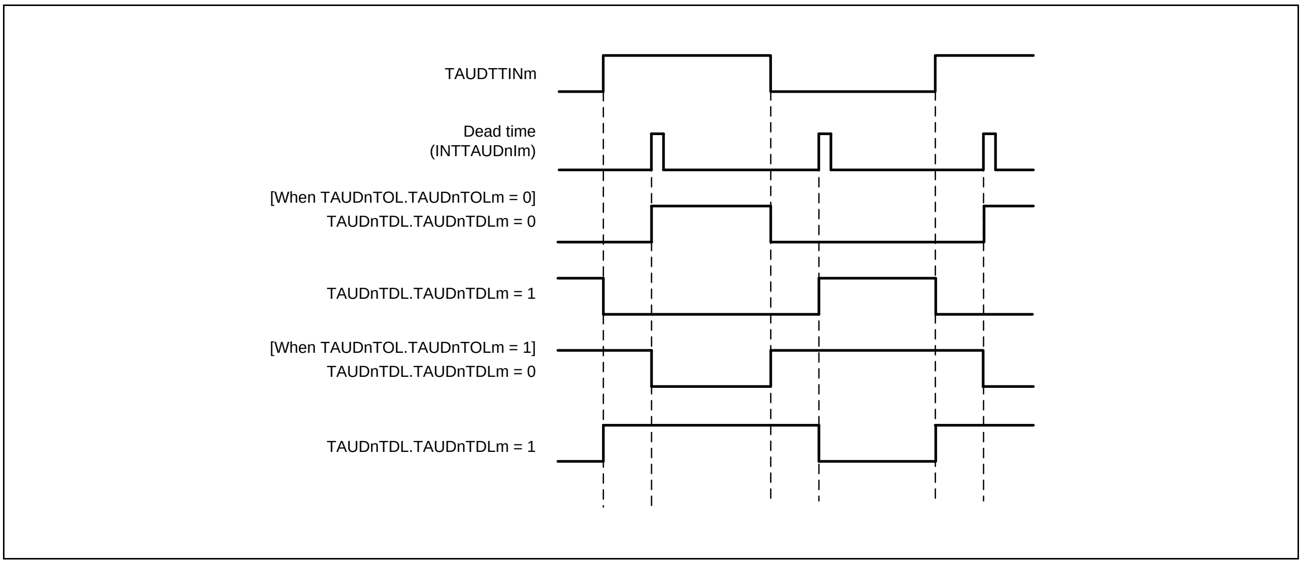


Figure 32.13 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

One-phase PWM output control requires a set of two channels:

- One even slave channel
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

32.7.3.6 Synchronous Channel Output Mode 2 with Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUDnTRO.TAUDnTROm), the modulation output bit (TAUDnTME.TAUDnTMEm), and the output level bit (TAUDnTOL.TAUDnTOLm) of a pair of slave channels.

For details, see **Section 32.16.3, Complementary Modulation Output Function**.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, four more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed during operation, its new value is applied upon detection of INTTAUDnIm at the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on an even slave channel.

32.7.3.7 Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a square wave while mode 2 has a triangular wave.

32.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUDnTS.TAUDnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

32.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUDnTS.TAUDnTSm is set to 1. The value of data register is also loaded when the counter starts.

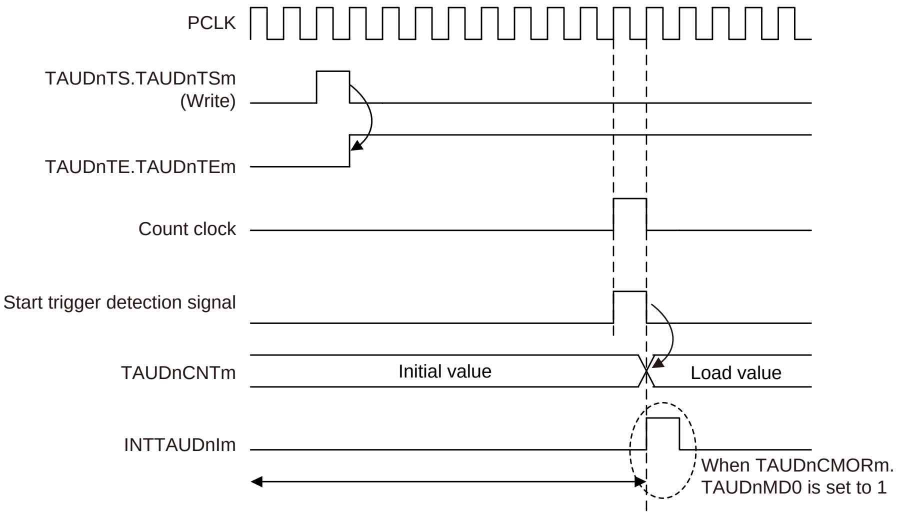


Figure 32.14 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

NOTE

Make sure to set TAUDnCMORm.TAUDnMD0 to 0 when using the count-up/-down mode.

32.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDnTS.TAUDnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

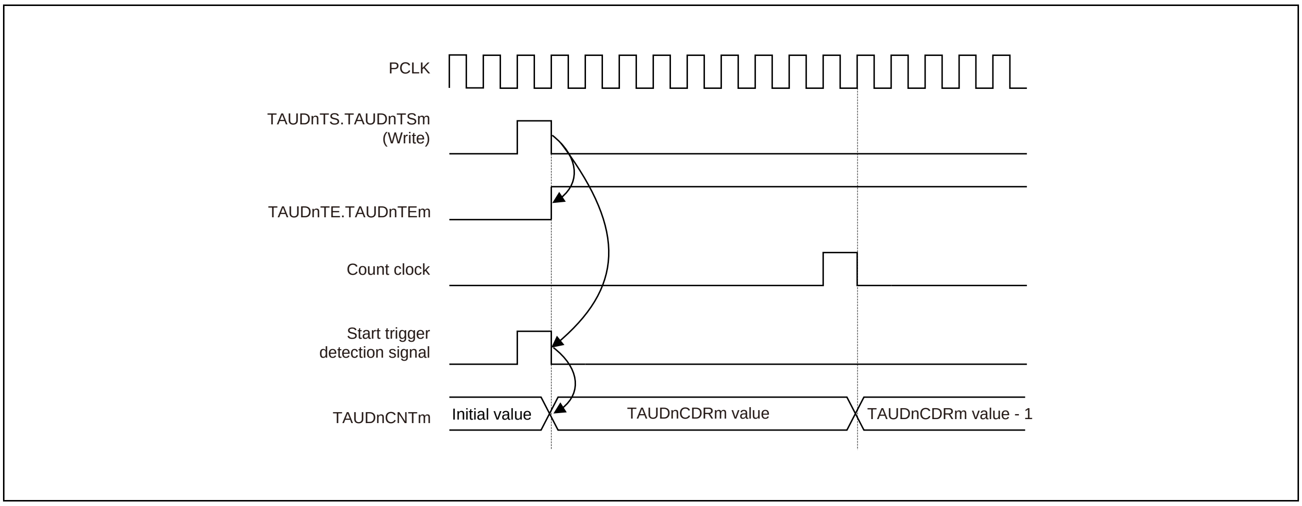


Figure 32.15 Start Timing in Event Count Mode

32.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUDTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

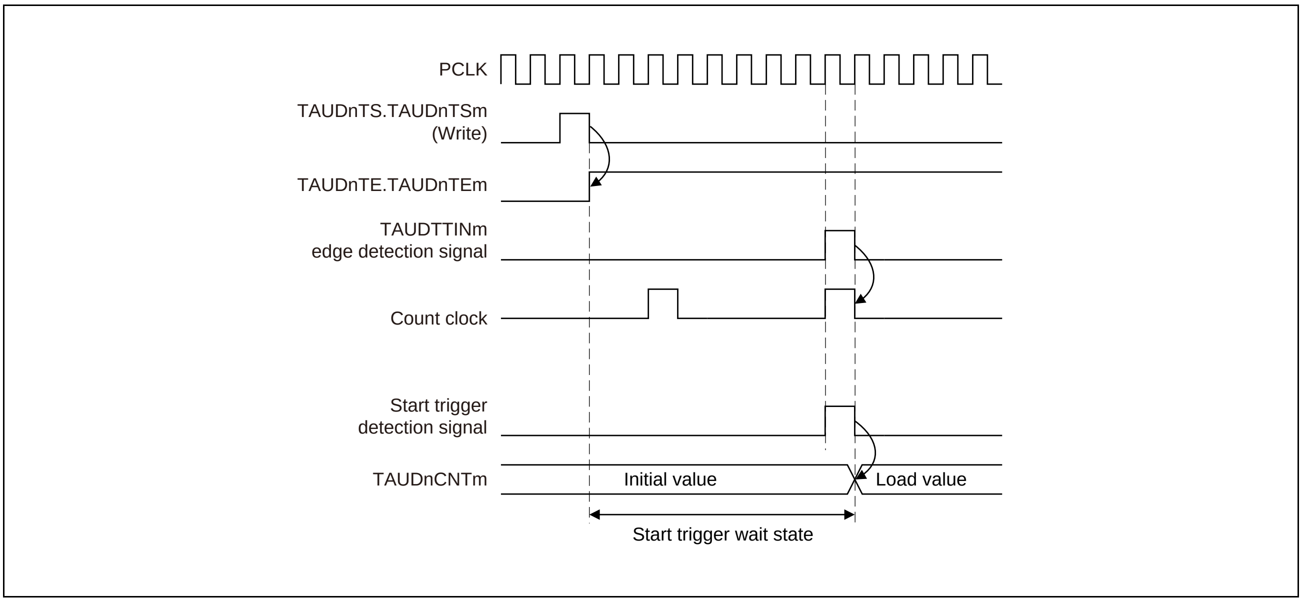


Figure 32.16 Count Start Timing in Other Operating Modes

32.9 TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDnIm is generated using the TAUDnCMORm.TAUDnMD0 bit. The generation of INTTAUDnIm when the TAUDnCMORm.TAUDnMD0 bit starts counting and the effect to TAUDTTOUTm depend on the selected function. For details, refer to the description of TAUDnCMORm.TAUDnMD0 of each function.

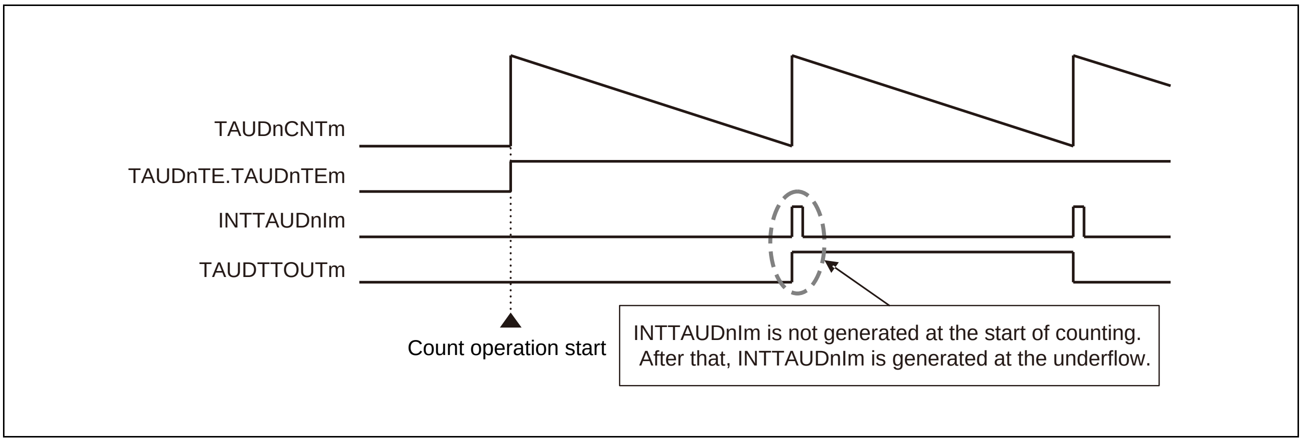


Figure 32.17 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 0)

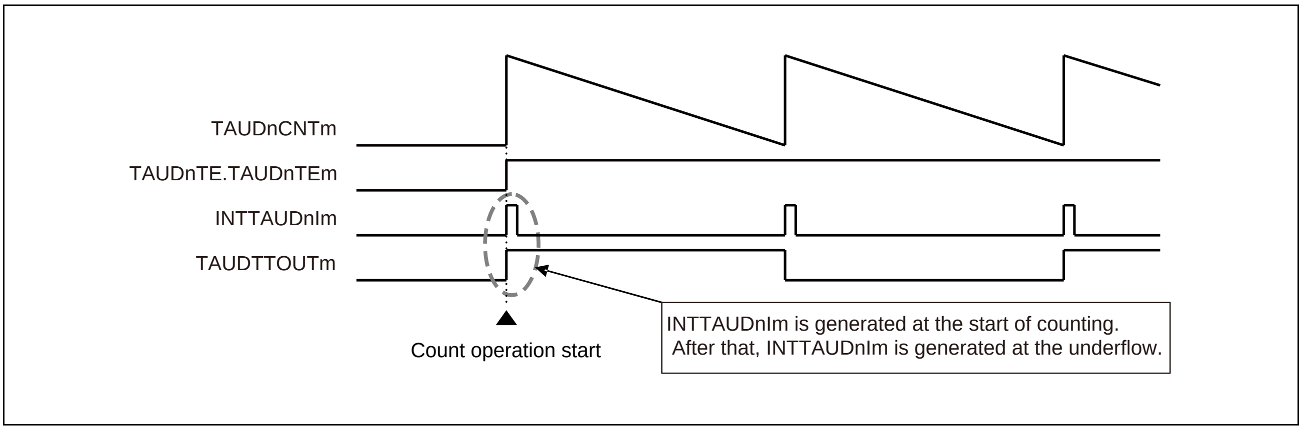


Figure 32.18 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 1)

32.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUDnCNTm = $FFFF_H$).
- Set TAUDnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUDTTINm input.
- The trigger detection settings (TAUDnCMORm.TAUDnSTS[2:0] and TAUDnCMURm.TAUDnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUDnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

32.10.1 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the TAUDTTINm input interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input pulse interval measurement function exceeds $FFFF_H$.

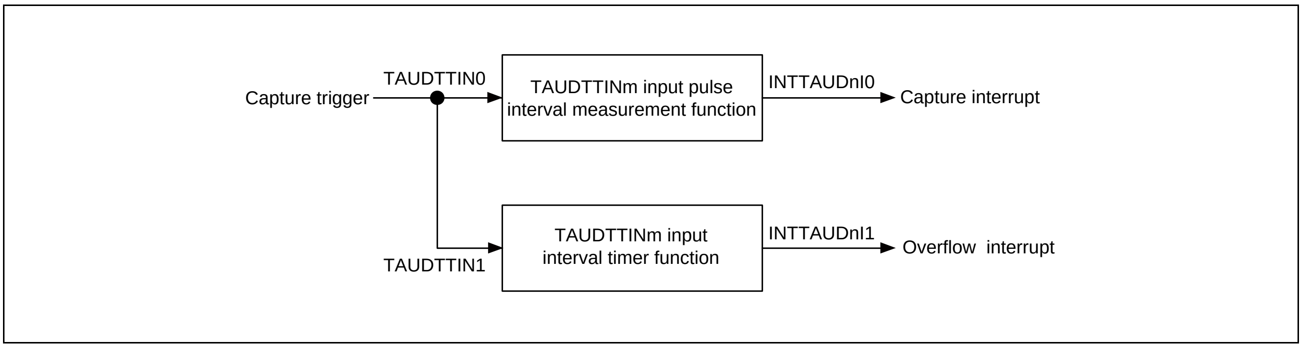


Figure 32.19 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

Timing diagram

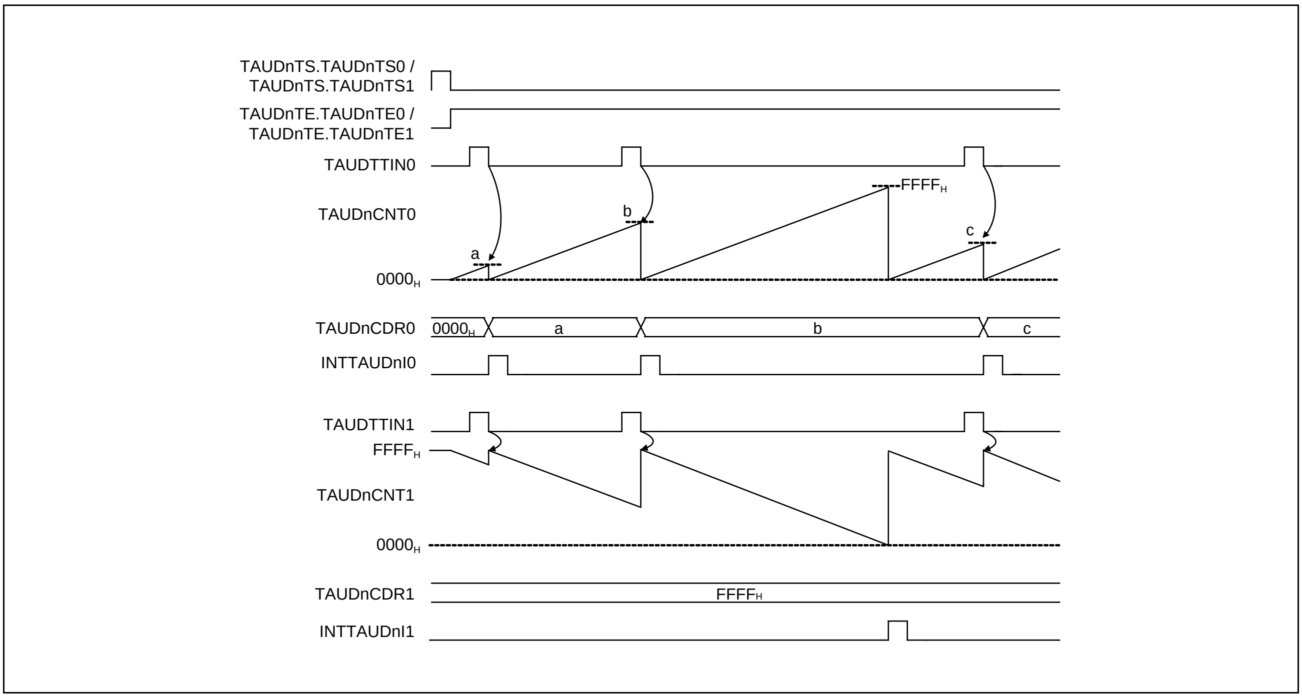


Figure 32.20 Interrupt Generation via Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

32.10.2 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at measuring the TAUDTTINm width) can detect the overflow when TAUDnCNTm of the TAUDTTINm input signal width measurement function exceeds $FFFF_H$.

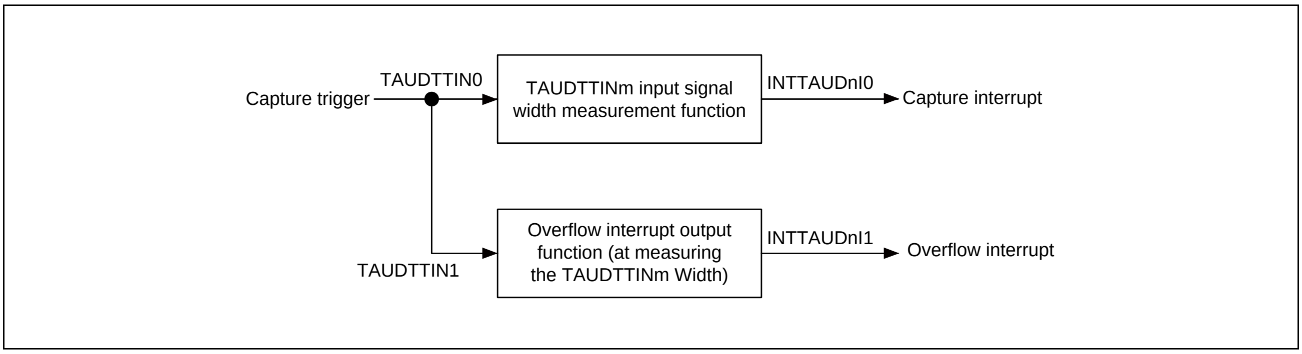


Figure 32.21 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

Timing diagram

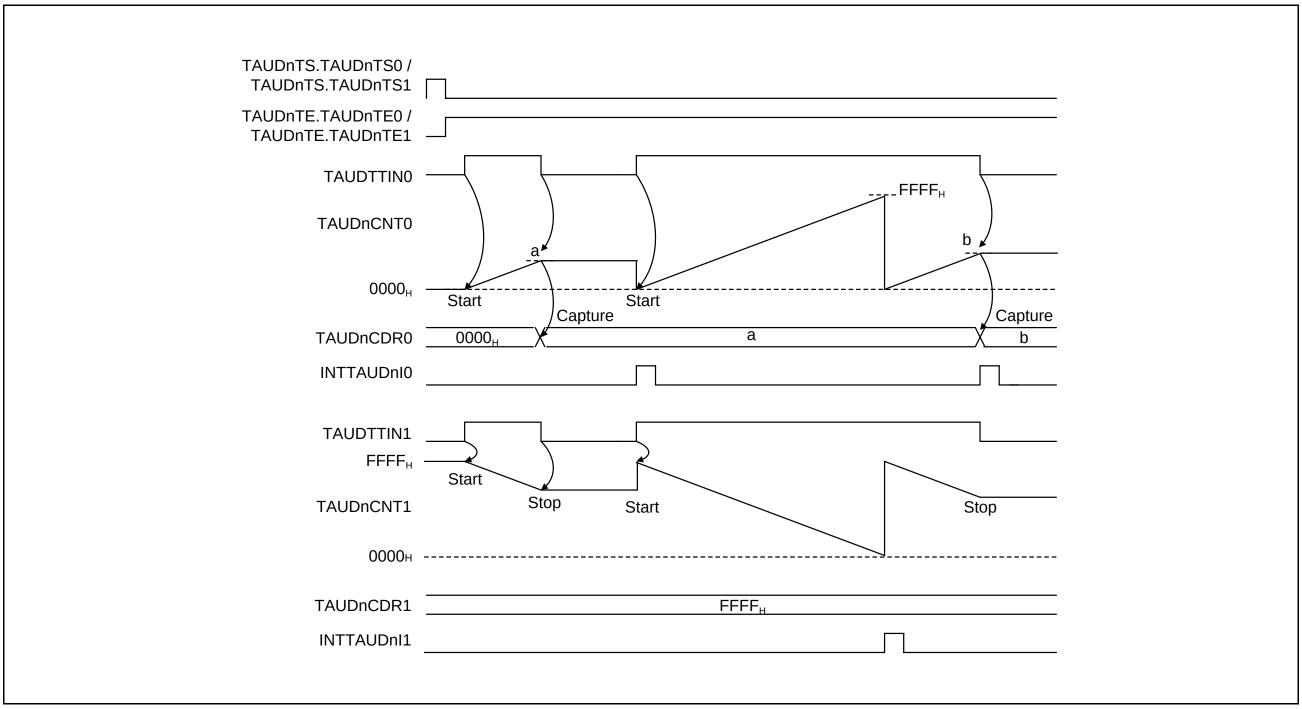


Figure 32.22 Interrupt Generation via Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

32.10.3 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUDnIm of the interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input position detection function exceeds $FFFF_H$.

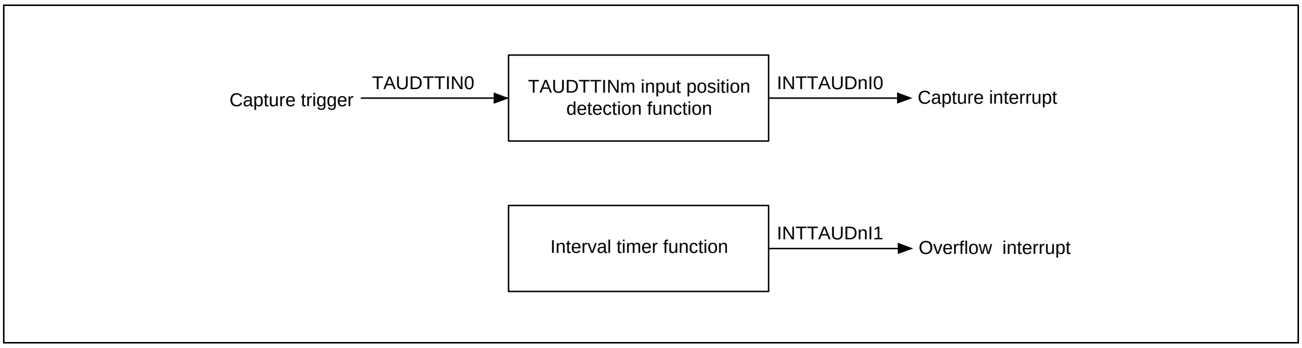


Figure 32.23 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

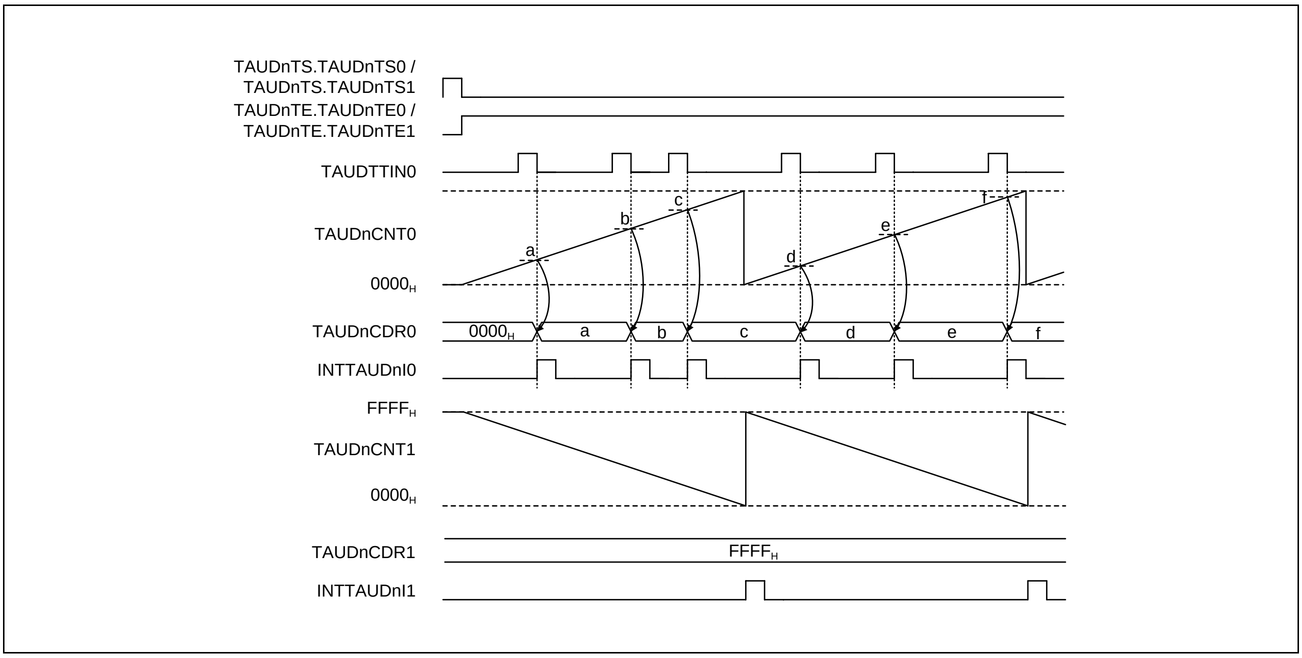


Figure 32.24 Interrupt Generation via Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

32.10.4 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at detecting the TAUDTTINm input period count) can detect the overflow when TAUDnCNTm of the TAUDTTINm input period count detection function exceeds $FFFF_H$.

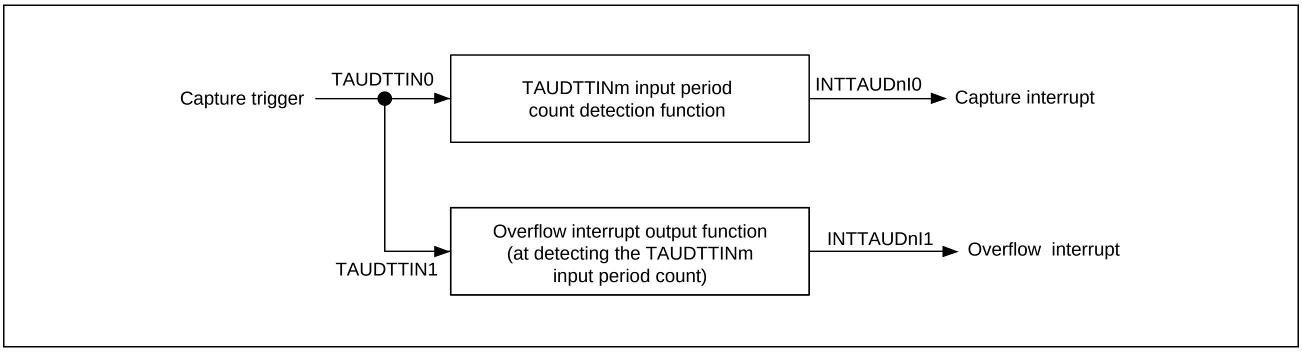


Figure 32.25 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

Timing diagram

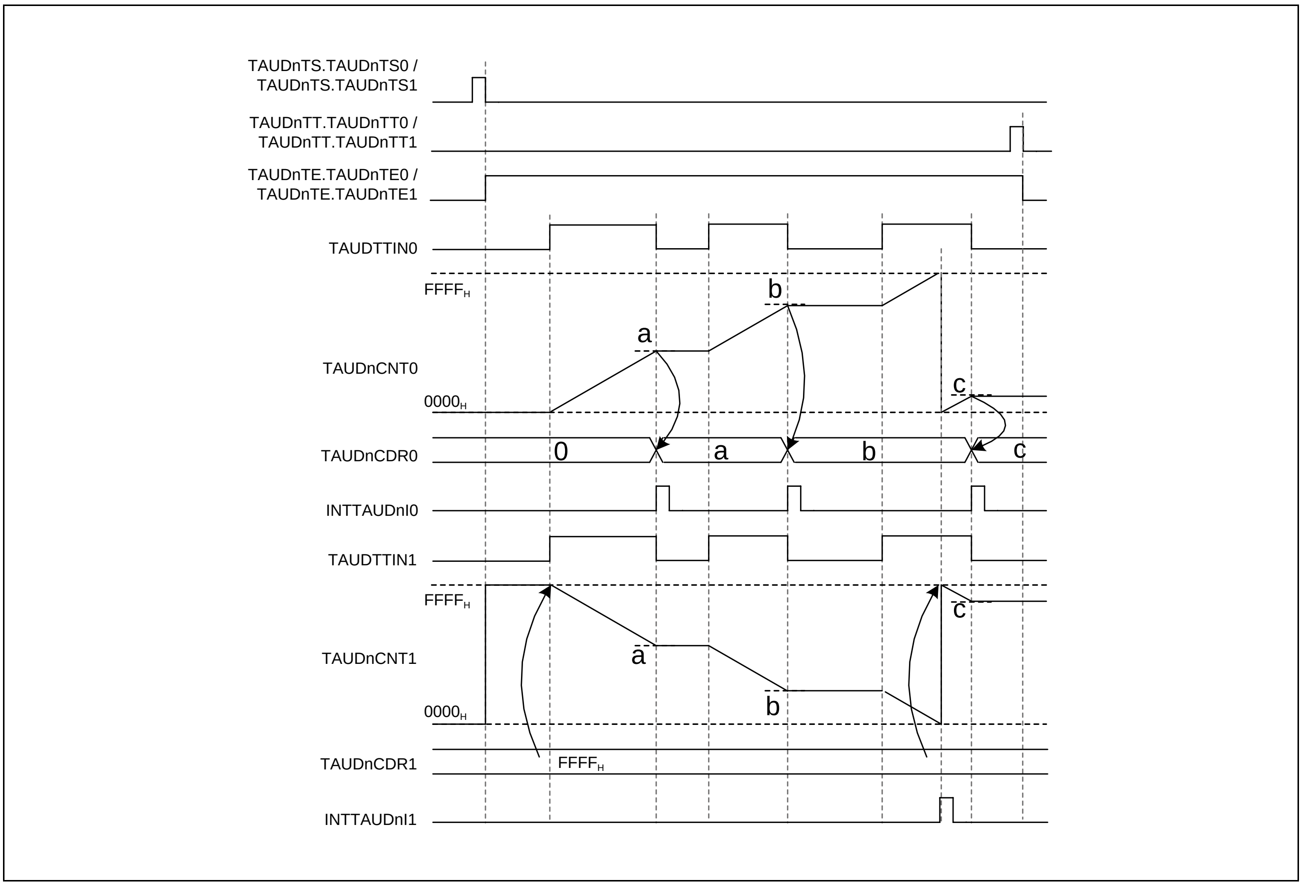


Figure 32.26 Interrupt Generation via Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

32.11 TAUDTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

Figure 32.27, Basic Edge Detection Timing shows when edge detection takes place.

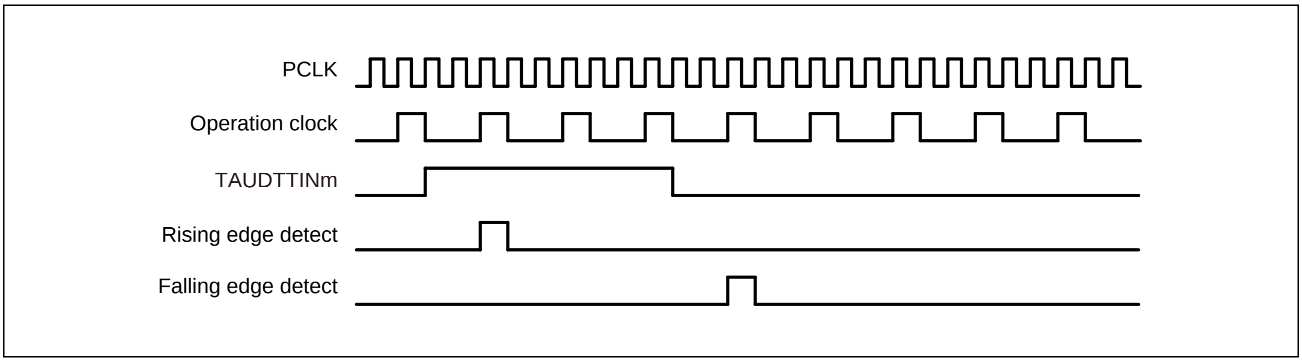


Figure 32.27 Basic Edge Detection Timing

Figure 32.27, Basic Edge Detection Timing shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnIm pin and TAUDn causes a delay time.

32.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the Timer Array Unit D. For a general overview of independent channel operation, see **Section 32.2, Overview**.

This section describes functions that generate interrupts at regular intervals or with a specified delay.

32.12.1 Interval Timer Function

32.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, see **Table 32.53, Contents of the TAUDnCMORm Register for Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, see **Section 32.7, Channel Output Modes**.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is written to TAUDnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and subsequently continues operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.12.1.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

32.12.1.3 Block Diagram and General Timing Diagram

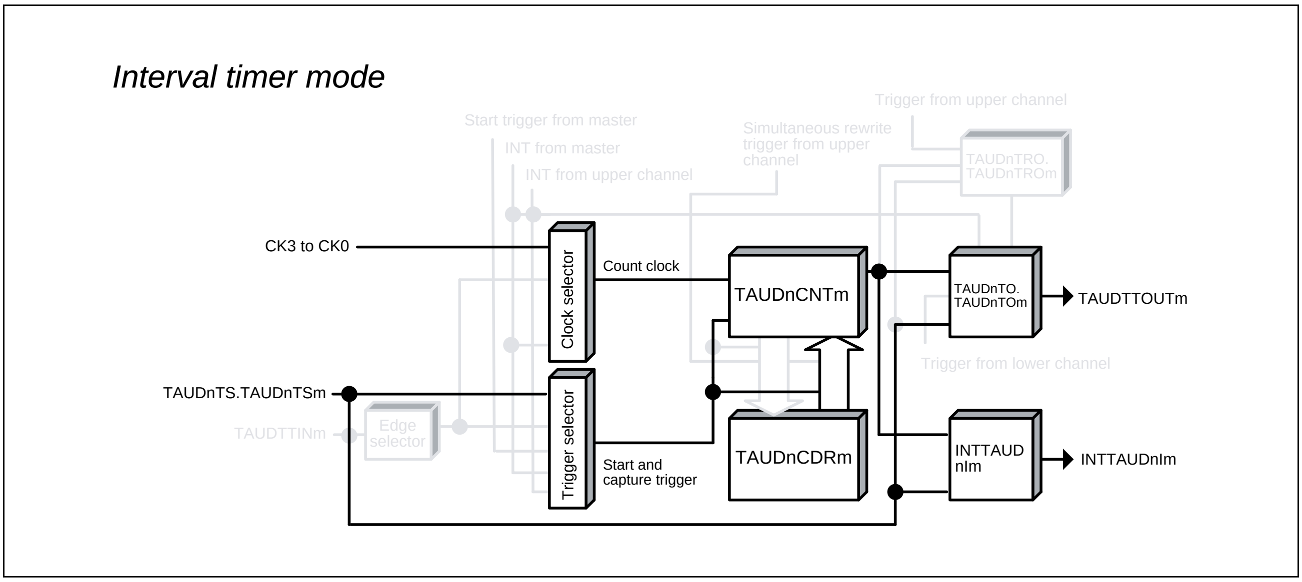


Figure 32.28 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).

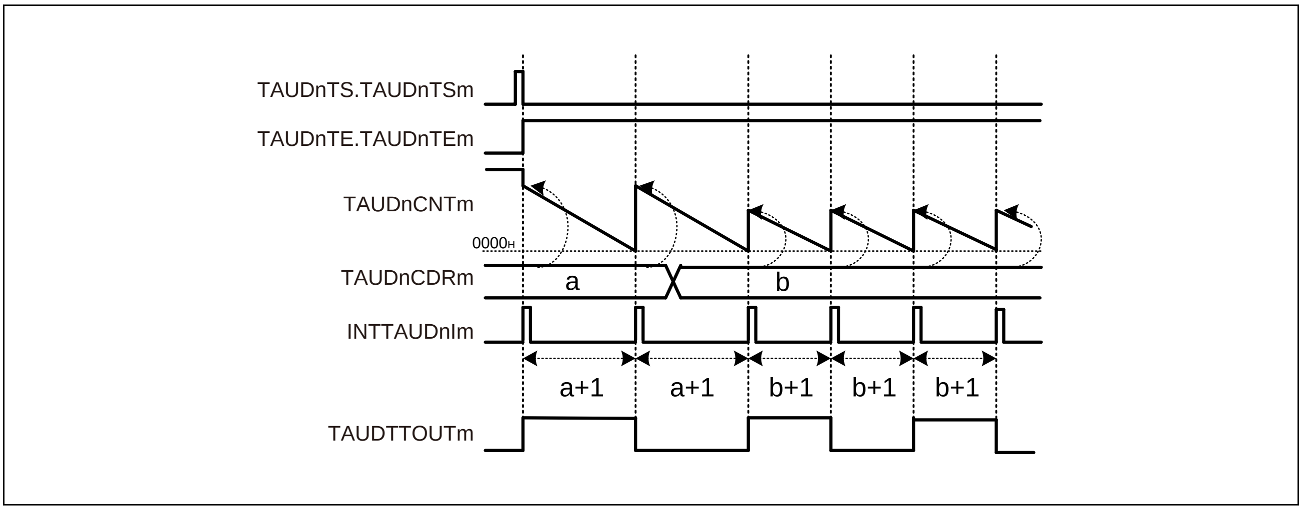


Figure 32.29 General Timing Diagram of Interval Timer Function

32.12.1.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.53 Contents of the TAUDnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Triggers the counter by software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.54 Contents of the TAUDnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

Table 32.55 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 32.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

Table 32.56 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.1.5 Operating Procedure for Interval Timer Function

Table 32.57 Operating Procedure for Interval Timer Function

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.53, Contents of the TAUDnCMORm Register for Interval Timer Function**, and **Table 32.54, Contents of the TAUDnCMURm Register for Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 32.55, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDTTOUTm toggles.
	During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H: ● The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. ● INTTAUDnIm is generated and TAUDTTOUTm toggles.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.12.1.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000$_H$, count clock = PCLK/2

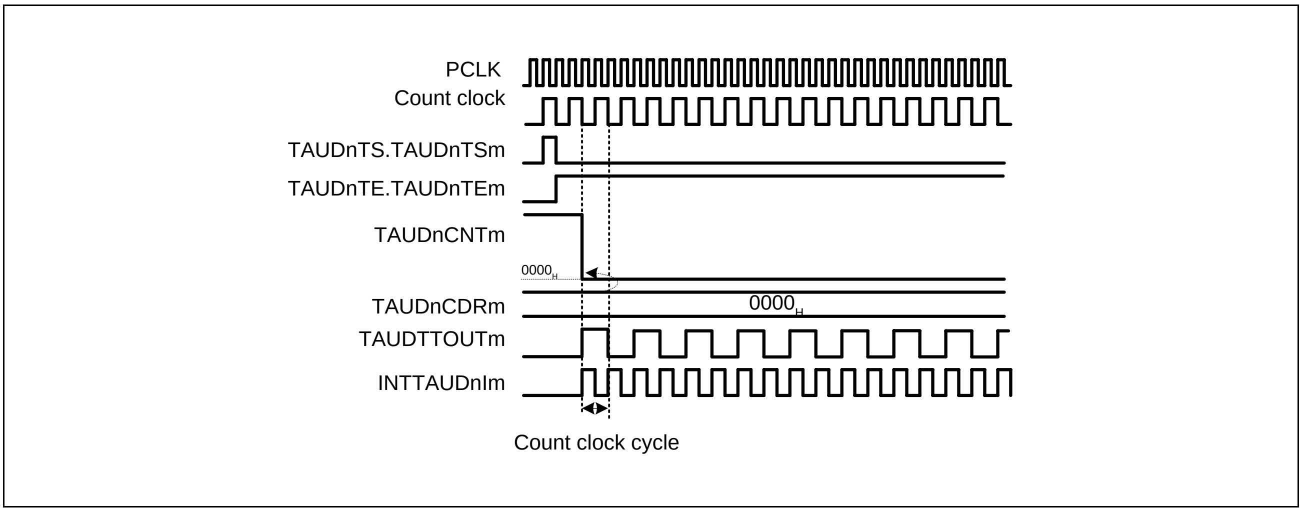


Figure 32.30 TAUDnCDRm = 0000$_H$, Count Clock = PCLK/2

- If TAUDnCDRm = 0000$_H$ and the count clock = PCLK/2, the TAUDnCDRm value is loaded into TAUDnCNTm every count clock, meaning that TAUDnCNTm is always 0000$_H$.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

(2) TAUDnCDRm = 0000$_H$, count clock = PCLK

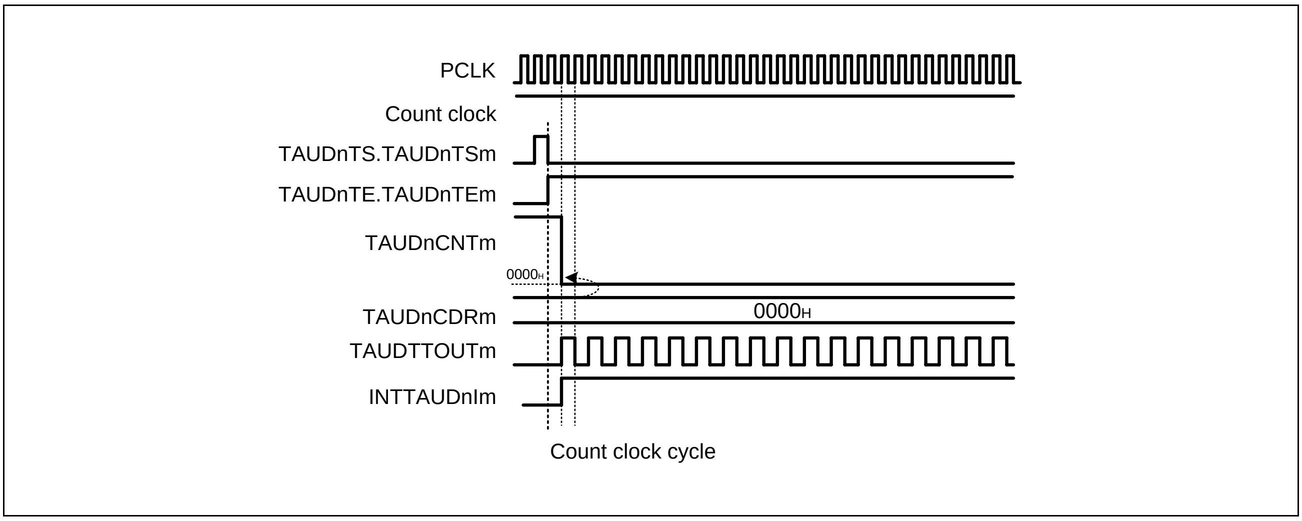


Figure 32.31 TAUDnCDRm = 0000$_H$, Count Clock = PCLK

- If TAUDnCDRm = 0000$_H$ and the count clock = PCLK, the TAUDnCDRm value is loaded into TAUDnCNTm every PCLK clock, meaning that TAUDnCNTm is always 0000$_H$.
- INTTAUDnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUDTTOUTm is toggled every PCLK clock.

(3) Operation Stop and Restart

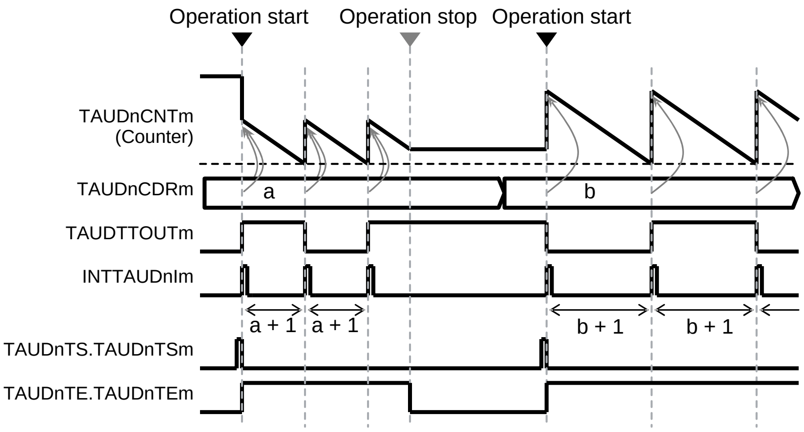


Figure 32.32 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

(4) Forced Restart (TAUDnCMORm.TAUDnMD0 = 1)

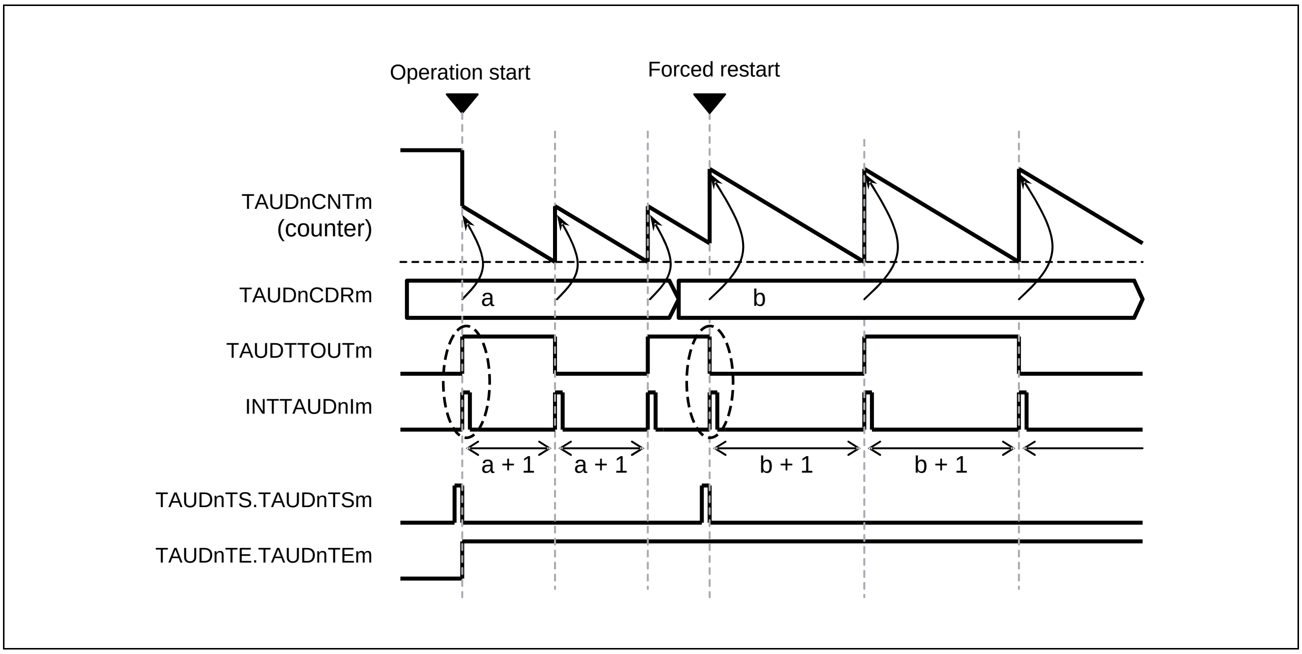


Figure 32.33 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUDnCDRm value is reflected to TAUDnCNTm and counting starts. Execute a forced restart to reflect the changed TAUDnCDRm value immediately.
- When a forced restart is made, an interrupt (INTTAUDnIm) is generated and TAUDTTOUTm is inverted.

(5) Forced Restart (TAUDnCMORm.TAUDnMD0 = 0)

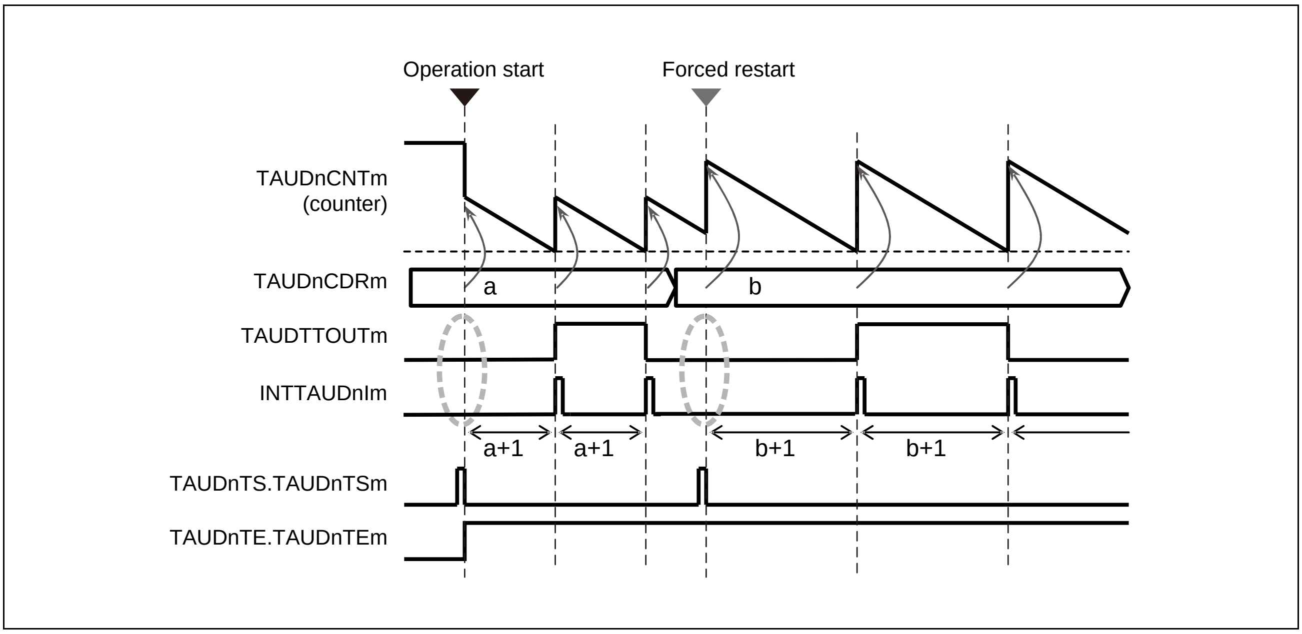


Figure 32.34 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 0)

- When a forced restart is made, an interrupt (INTTAUDnIm) is not generated and TAUDTTOUTm is not inverted.

32.12.2 TAUDTTINm Input Interval Timer Function

32.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals or when a valid TAUDTTINm input edge is detected. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to interval timer mode. See **Table 32.58, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**.
- The channel output mode should be set to independent channel output mode 1. See **Section 32.7, Channel Output Modes**.

Functional description

This function operates in an identical manner to the interval timer function (see **Section 32.12.1, Interval Timer Function**) except that this function is restarted by a valid TAUDTTINm input edge. The type of edge used as a trigger is specified using the TAUDnCMURm.TAUDnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

32.12.2.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

32.12.2.3 Block Diagram and General Timing Diagram

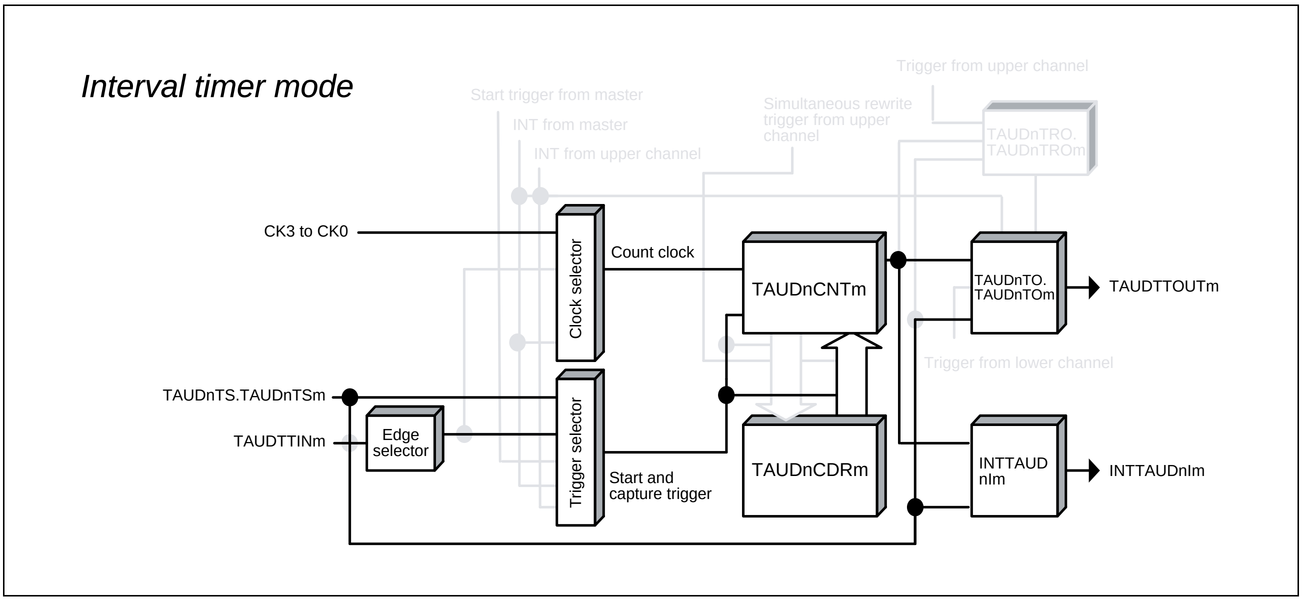


Figure 32.35 Block Diagram of TAUDTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).
- Rising edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

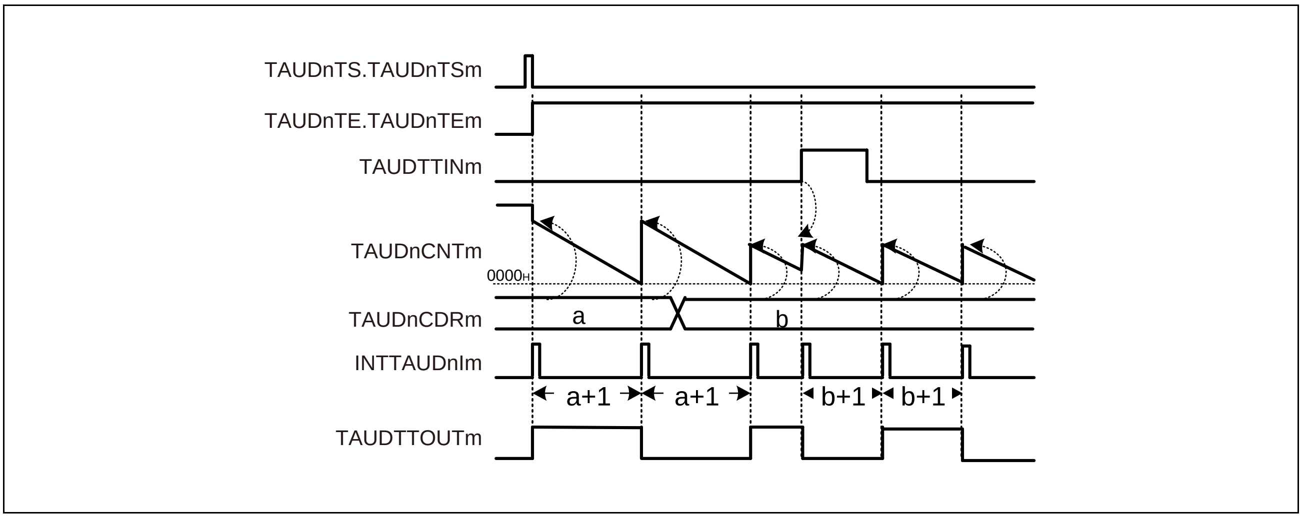


Figure 32.36 General Timing Diagram of TAUDTTINm Input Interval Timer Function

32.12.2.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.58 Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.59 Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 32.60 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 32.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 32.61 Simultaneous Rewrite Settings for TAUDTTINm Input Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.2.5 Operating Procedure for TAUDTTINm Input Interval Timer Function

Table 32.62 Operating Procedure for TAUDTTINm Input Interval Timer Function

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.58, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**, and **Table 32.59, Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 32.60, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDTTOUTm toggles.
	During Operation	The values of the TAUDnCMURm.TAUDnTIS[1:0] and the TAUDnCDRm register are changeable at any time. The TAUDnCNTm register can be read at all times. Detection of TAUDTTINm edge	TAUDnCNTm counts down. When the counter reaches 0000_H: ● The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. ● INTTAUDnIm is generated and TAUDTTOUTm toggles. When a TAUDTTINm input valid edge is detected during count operation, the TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 32.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUDTTINm input edge.

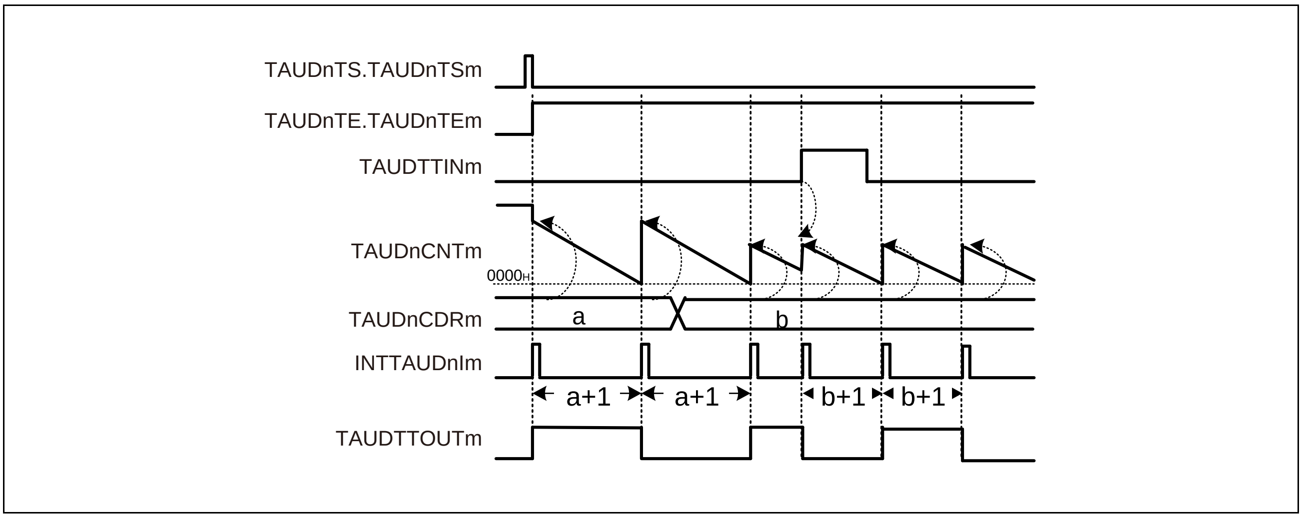


Figure 32.37 Counter Triggered by Rising TAUDTTINm Input Edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B), TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDTTINm input edge is detected, an interrupt is generated which causes TAUDTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

32.12.3 Clock Divide Function

32.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUDTTINm is divided by a factor related to TAUDnCDRm, and the resulting signal is output to TAUDTTOUTm.

Prerequisites

- TAUDTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See **Table 32.63, Contents of the TAUDnCMORm Register for Clock Divide Function**.)
- The channel output mode should be set to independent channel output mode 1. (See **Section 32.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value, using TAUDTTINm as a count clock.

When the counter value reaches 0000_H, INTTAUDnIm occurs and TAUDTTOUTm signal is toggled. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The value of TAUDnCDRm can be rewritten at any time. The changed value of TAUDnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The function can be restarted by setting TAUDnTS.TAUDnTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUDnTS.TAUDnTSm = 1 during operation (forced restart).

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

TAUDTTINm input signals are sampled at the frequency of the operation clock set by TAUDnCMORm.TAUDnCKS[1:0] bits. Therefore, the TAUDTTOUTm output clock cycle has an error of ± 1 operation clock cycle.

32.12.3.2 Equations

- When rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling and rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/(TAUDnCDRm + 1)

32.12.3.3 Block Diagram and General Timing Diagram

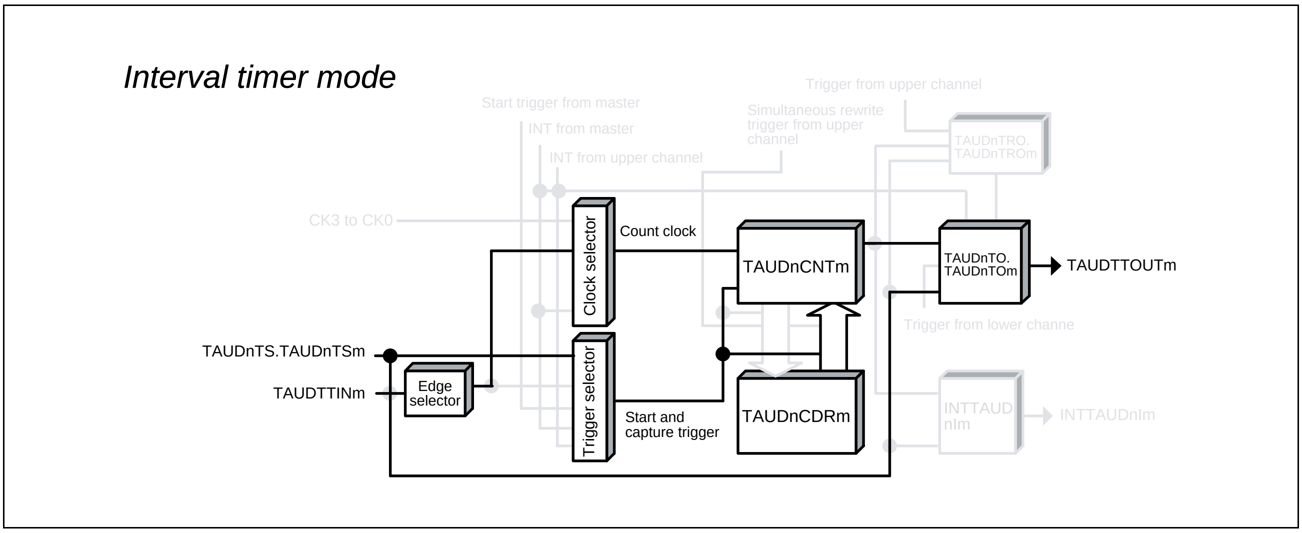


Figure 32.38 Block Diagram of Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

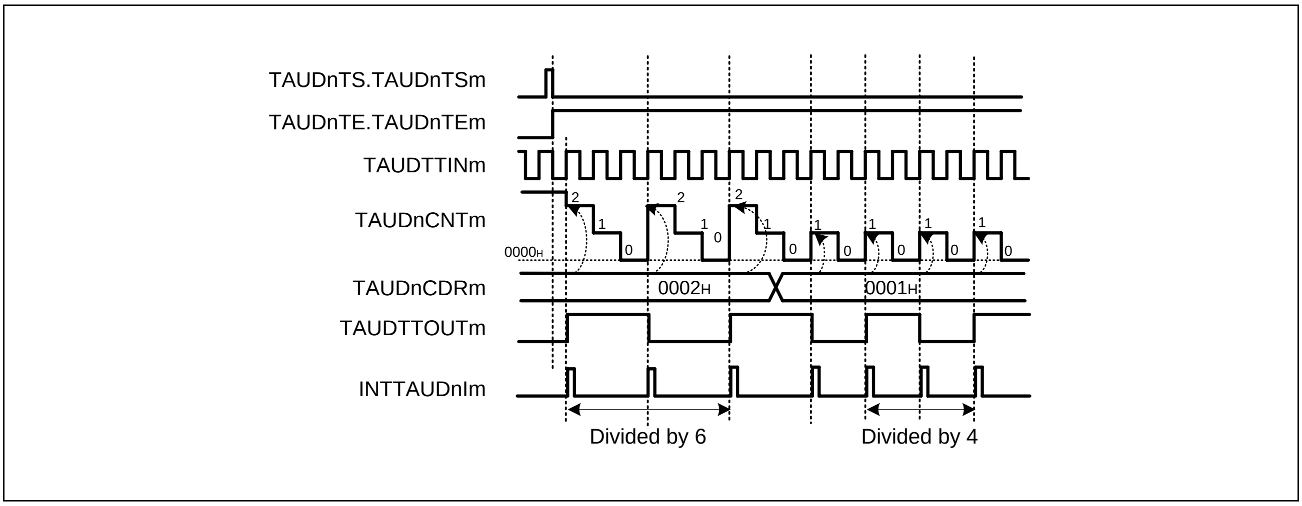


Figure 32.39 General Timing Diagram of Clock Divide Function

32.12.3.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.63 Contents of the TAUDnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.64 Contents of the TAUDnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 32.65 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the clock divide function. Therefore, these registers should be set to 0.

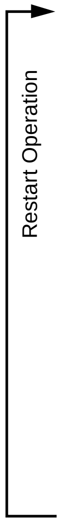
Table 32.66 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.3.5 Operating Procedure for Clock Divide Function

Table 32.67 Operating Procedure for Clock Divide Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.63, Contents of the TAUDnCMORm Register for Clock Divide Function**, and **Table 32.64, Contents of the TAUDnCMURm Register for Clock Divide Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 32.65, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 is set to 1, INTTAUDnIm is generated and TAUDTTOUTm is toggled.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When the counter reaches 0000_H: ● TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. ● INTTAUDnIm is generated. ● TAUDTTOUTm is toggled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.



32.12.3.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

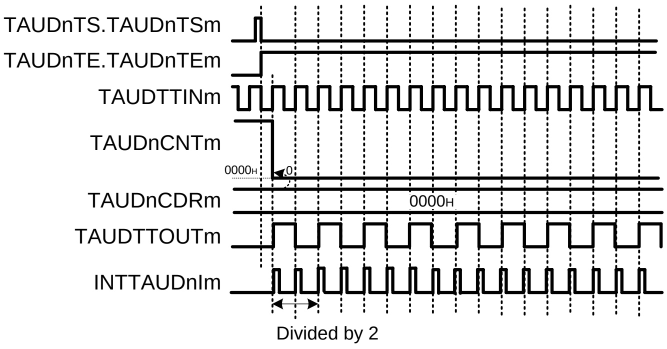


Figure 32.40 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If TAUDnCDRm is 0000_H, TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

Figure 32.40, TAUDnCDRm = 0000H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01B shows an operation timing example. Actually, there is a delay from TINm detection until TOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUDnIm pin and TAUDn.

(2) Operation Restart

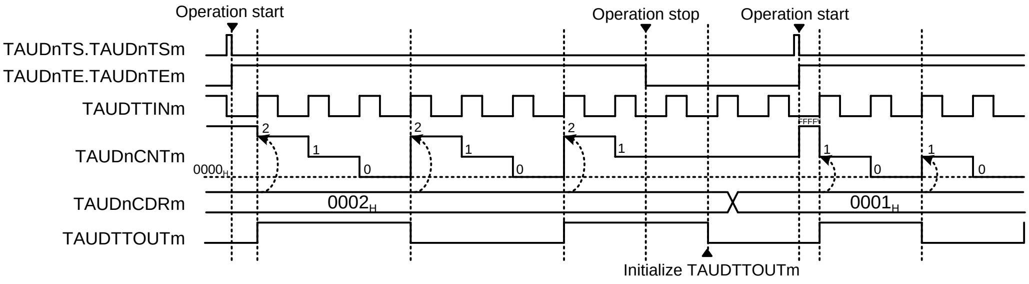


Figure 32.41 Operation Restart
(TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

To reset the value of TAUDTTOUTm:

- Set TAUDnTOE.TAUDnTOEm = 0 when the counter is stopped (TAUDnTE.TAUDnTEm = 0).
- Then, write either 0 or 1 to TAUDnTO.TAUDnTOm to set the new start value of TAUDTTOUTm.

(3) Forced Restart

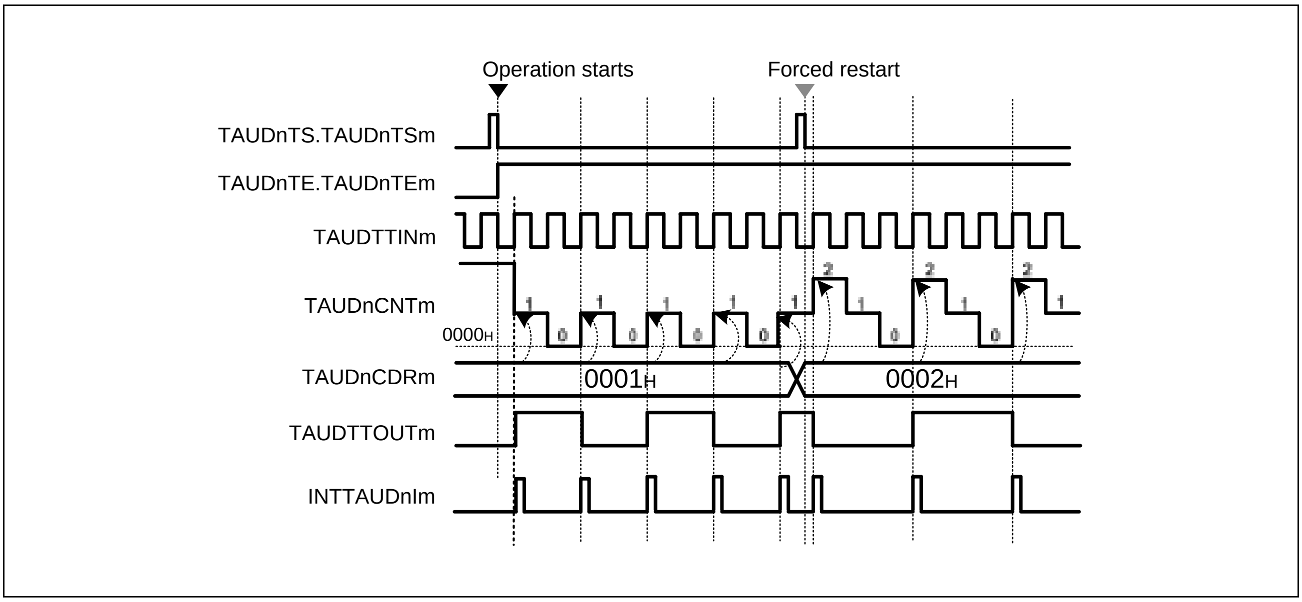


Figure 32.42 Forced Restart Operation
(TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm = 1 during operation.
- The value of TAUDnCDRm is written to TAUDnCNTm and the count operation restarts.
- TAUDTTOUTm restarts at the same level as before the forced restart.

32.12.4 External Event Count Function

32.12.4.1 Overview

Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of valid TAUDTTINm input edges are detected.

Prerequisites

- The operating mode should be set to the event count mode. (See **Table 32.68, Contents of the TAUDnCMORm Register for External Event Count Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When the counter starts, the current value of TAUDnCDRm is loaded into TAUDnCNTm.

When a valid TAUDTTINm input edge is detected, the value of TAUDnCNTm decrements by 1. TAUDnCNTm retains this value until a valid TAUDTTINm input edge is detected or the counter is restarted.

When the valid edge is detected for the (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDnTS.TAUDnTSm to 1 during operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

An edge type used as a trigger is specified by TAUDnCMURm.TAUDnTIS[1:0] bits.

- When TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 10_B, both edges are counted.

32.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

32.12.4.3 Block Diagram and General Timing Diagram

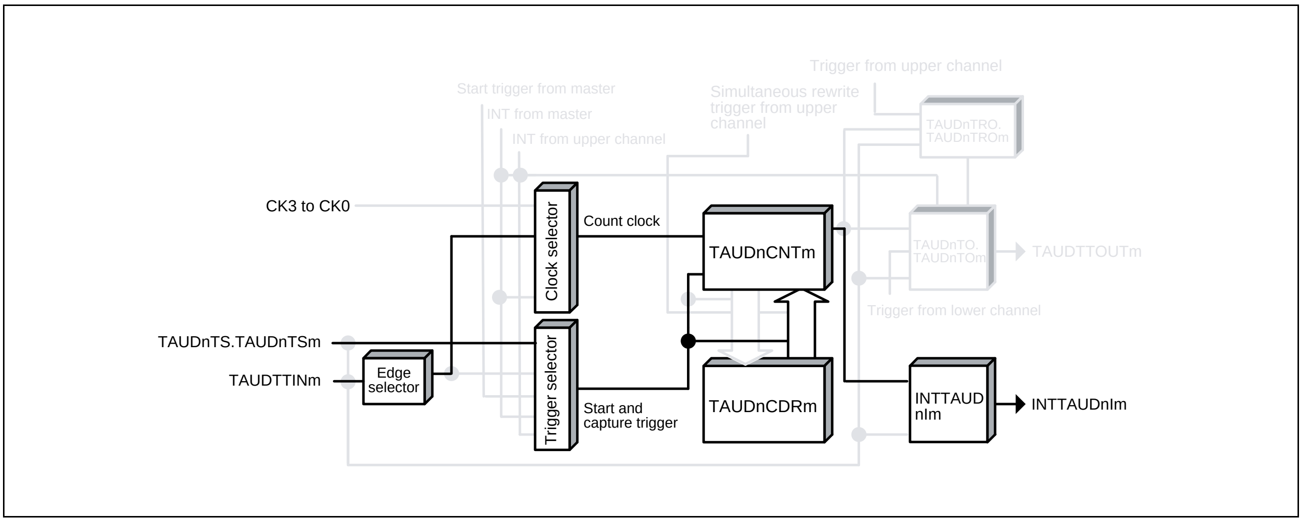


Figure 32.43 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

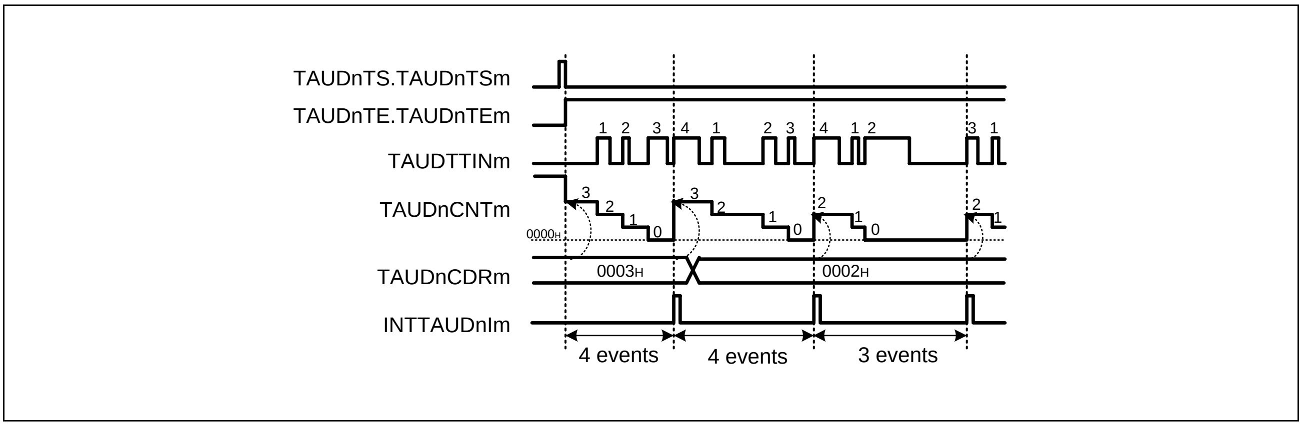


Figure 32.44 General Timing Diagram of External Event Count Function

32.12.4.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.68 Contents of the TAUDnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.69 Contents of the TAUDnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited.

(3) Channel Output Mode

The channel output mode is not used by this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the external event count function. Therefore, these registers should be set to 0.

Table 32.70 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.4.5 Operating Procedure for External Event Count Function

Table 32.71 Operating Procedure for External Event Count Function

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.68, Contents of the TAUDnCMORm Register for External Event Count Function**, and **Table 32.69, Contents of the TAUDnCMURm Register for External Event Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value and waits for TAUDTTINm input edge detection.
	During Operation	Detection of TAUDTTINm edge The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When effective edges are detected (TAUDnCDRm + 1) times: ● TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. ● INTTAUDnIm is generated. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

32.12.4.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000$_H$

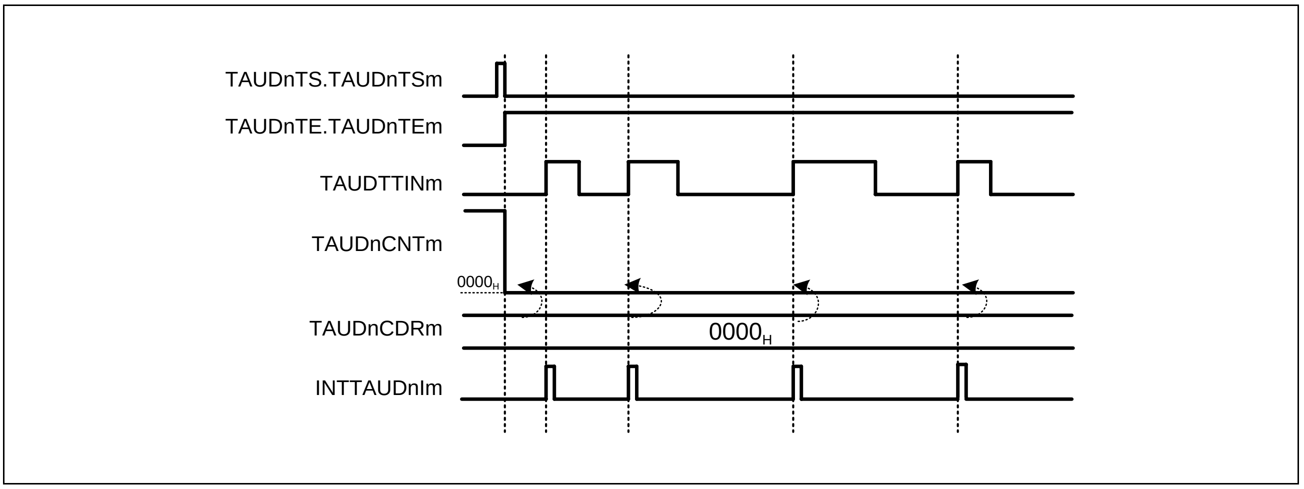


Figure 32.45 TAUDnCDRm = 0000$_H$, TAUDnCMURm.TAUDnTIS[1:0] = 01$_B$

- If 0000$_H$ = TAUDnCDRm, 0000$_H$ is loaded into TAUDnCNTm each time a valid TAUDTTINm input edge is detected.
  In other words, INTTAUDnIm is generated each time a valid TAUDTTINm input edge is detected.

(2) Operation Stop and Restart

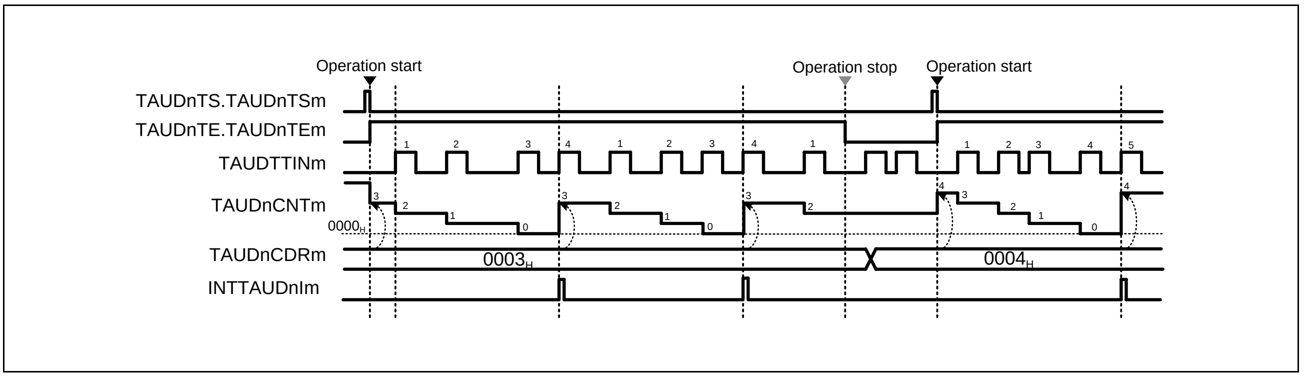


Figure 32.46 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01$_B$)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value. TAUDTTINm continues and TAUDnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm loads the TAUDnCDRm value and restarts count operation.

(3) Forced Restart

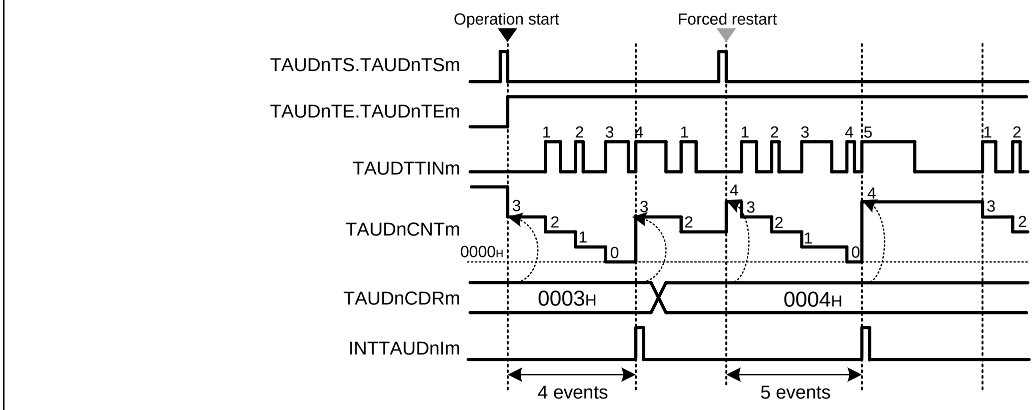


Figure 32.47 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

Once a forced restart is made, the changed TAUDnCDRm value is applied to TAUDnCNTm immediately.

- The counter can be restarted without making a stop by setting TAUDnTS.TAUDnTSm to 1 during operation.
- The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter awaits the next valid TAUDTTINm input edge.

32.12.5 Delay Count Function

32.12.5.1 Overview

Summary

This function generates interrupts (INTTAUDnIm), which have a defined delay to the TAUDTTINm input signal. TAUDTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites

- The operating mode should be set to one-count mode. See **Table 32.72, Contents of the TAUDnCMORm Register for Delay Count Function**.
- TAUDTTOUTm is not used with this function.
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

32.12.5.2 Equations

Delay between TAUDTTINm and INTTAUDnIm = count clock cycle × (TAUDnCDRm + 1)

32.12.5.3 Block Diagram and General Timing Diagram

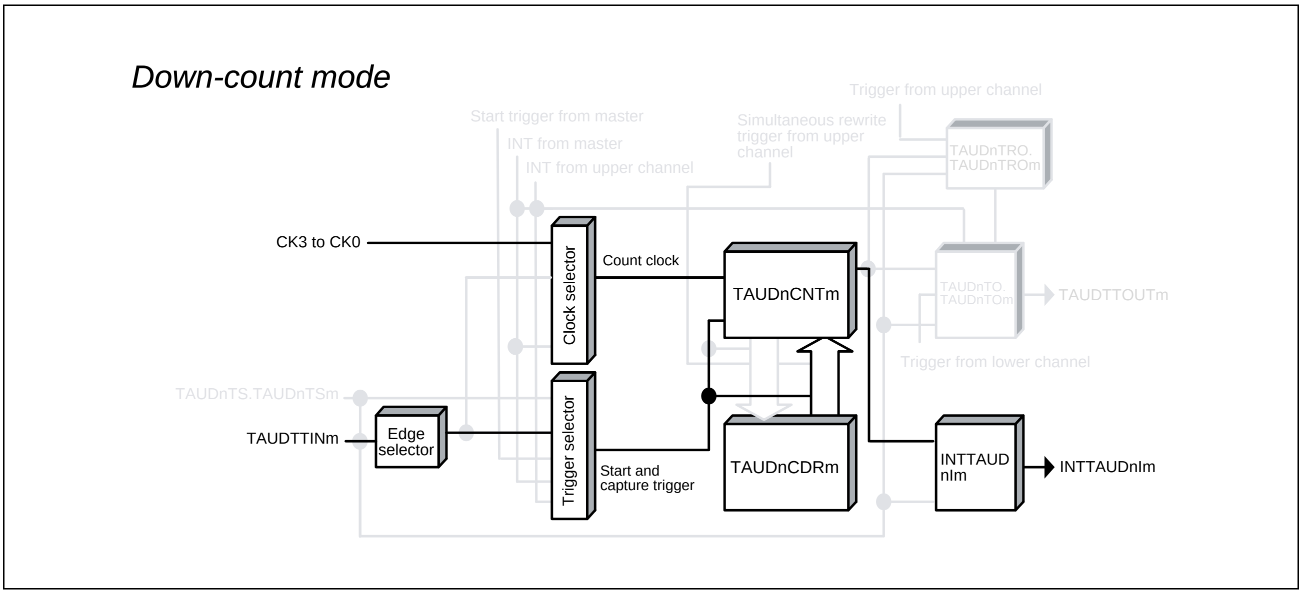


Figure 32.48 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

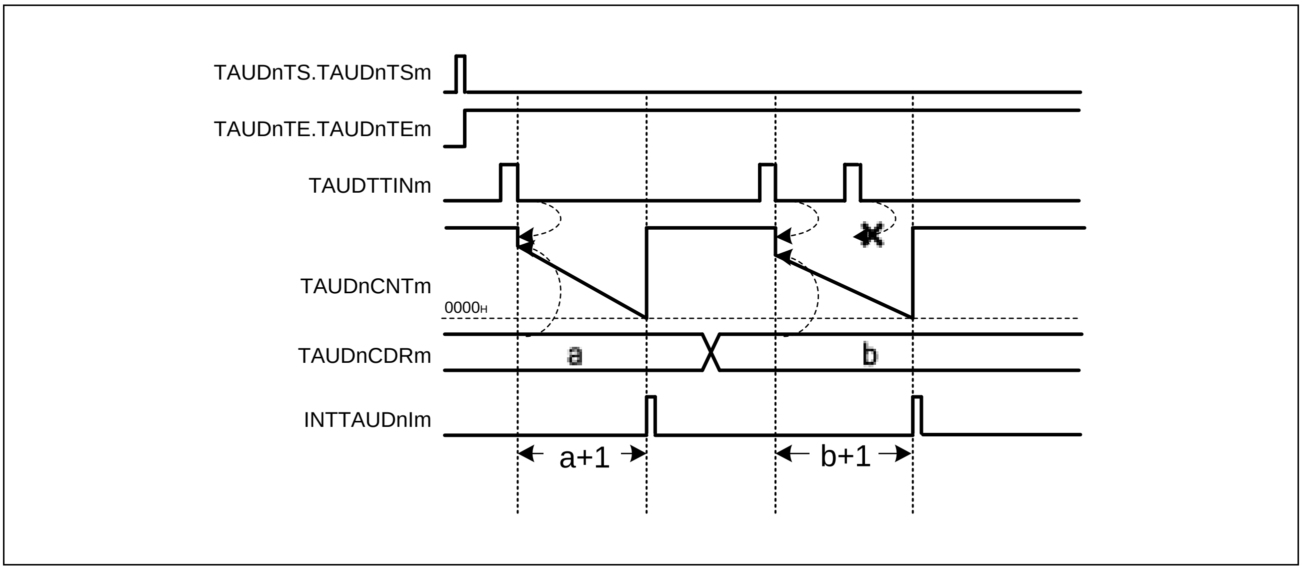


Figure 32.49 General Timing Diagram of Delay Count Function

32.12.5.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.72 Contents of the TAUDnCMORm Register for Delay Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables a start trigger during operation

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.73 Contents of the TAUDnCMURm Register for Delay Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the delay count function. Therefore, these registers should be set to 0.

Table 32.74 Simultaneous Rewrite Settings for Delay Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.5.5 Operating Procedure for Delay Count Function

Table 32.75 Operating Procedure for Delay Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.72, Contents of the TAUDnCMORm Register for Delay Count Function**, and **Table 32.73, Contents of the TAUDnCMURm Register for Delay Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, the TAUDnCDRm value is loaded in TAUDnCNTm.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H. INTTAUDnIm is generated. TAUDnCNTm stops counting, returns $FFFF_H$, and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its value.

Restart Operation

32.12.6 One-Pulse Output Function

32.12.6.1 Overview

Summary

This function generates an interrupt (INTTAUDnIm) when a valid TAUDTTINm input edge is detected and at a defined interval afterward. TAUDTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode should be set to pulse one-count mode. (See **Table 32.76, Contents of the TAUDnCMORm Register for One-Pulse Output Function**.)
- The channel output mode should be set to independent channel output mode 2. (See **Section 32.7, Channel Output Modes**.)
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. An interrupt is generated and TAUDTTOUTm is set to active level.

When the counter reaches 0001_H, an interrupt is generated and TAUDTTOUTm is set to the inactive level. The counter stops at 0000_H and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

32.12.6.2 Equations

Interval between TAUDTTINm and INTTAUDnIm = TAUDTTOUTm (timer output) width = count clock cycle × TAUDnCDRm

32.12.6.3 Block Diagram and General Timing Diagram

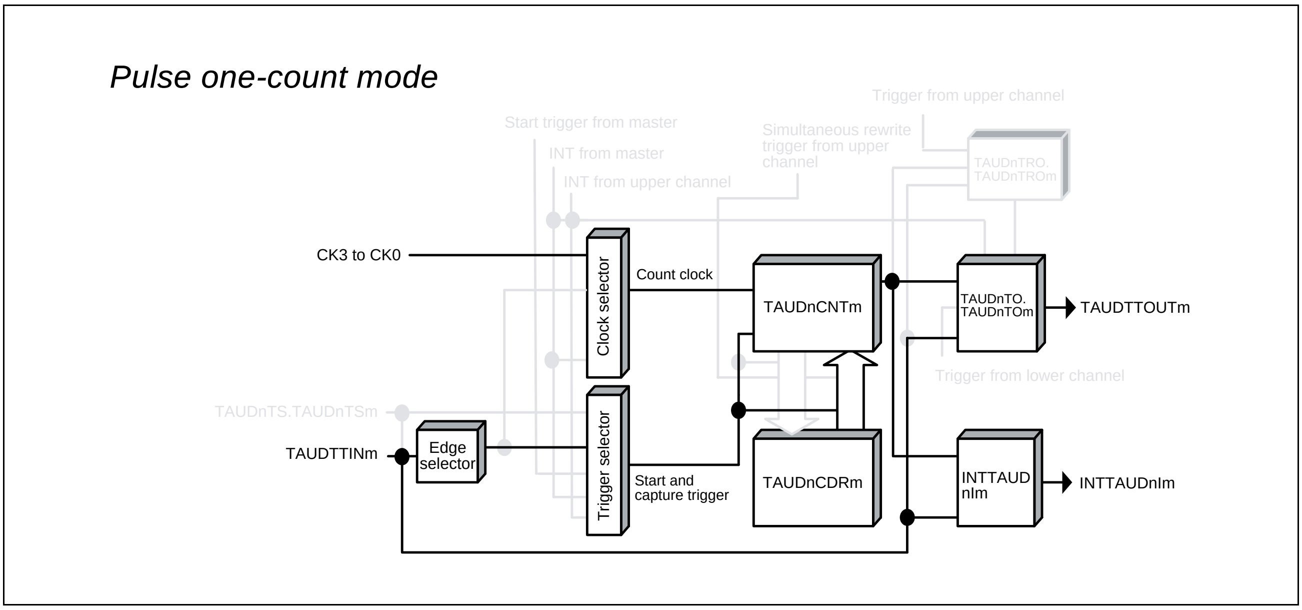


Figure 32.50 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

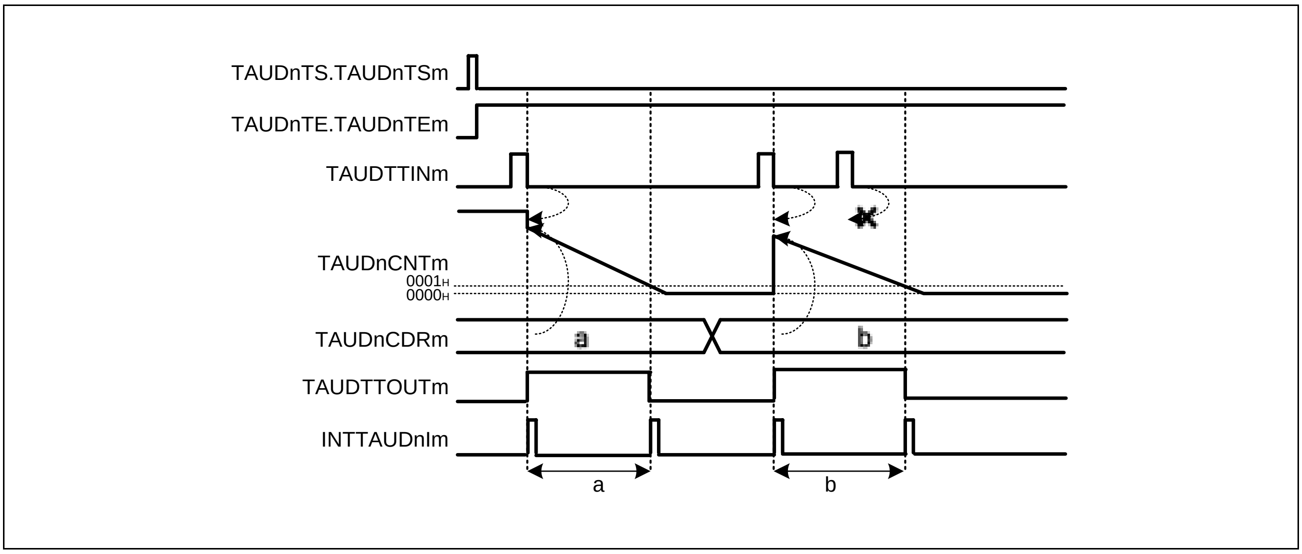


Figure 32.51 General Timing Diagram of One-Pulse Output Function

32.12.6.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.76 Contents of the TAUDnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables a start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.77 Contents of the TAUDnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 32.78 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode controlled by software.
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Table 32.52, Channel Output Modes**.

(4) Simultaneous Rewrite

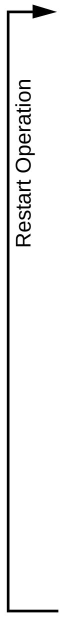
The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

Table 32.79 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.6.5 Operating Procedure for One-Pulse Output Function

Table 32.80 Operating Procedure for One-Pulse Output Function



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.76, Contents of the TAUDnCMORm Register for One-Pulse Output Function**, and **Table 32.77, Contents of the TAUDnCMURm Register for One-Pulse Output Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 32.78, Control Bit Settings in Independent Channel Output Mode 2**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	INTTAUDnIm is generated when TAUDnCNTm starts and TAUDTTOUTm is set to its active level. TAUDnCNTm counts down. When the counter reaches 0001_H: ● INTTAUDnIm is generated. ● TAUDTTOUTm is set to its inactive level. TAUDnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.12.7 TAUDTTINm Input Pulse Interval Measurement Function

32.12.7.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUDnCSRm.TAUDnOVF to measure the interval of the TAUDTTINm input signals.

Prerequisites

- The operating mode should be set to capture mode. See **Table 32.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function**.
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUDTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 32.81 Effects of Overflow

TAUDnCMORm. TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input is Detected	
	TAUDnCDRm	TAUDnCSRm. TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm. TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm set to 0, TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, the overflow bit (TAUDnCSRm.TAUDnOVF) can be cleared only by setting TAUDnCSCm.TAUDnCLOV = 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the interval of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, the overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm stops but retains its value. While the function is stopped, valid TAUDTTINm input edge detection and TAUDnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUDnCMORm.TAUDnCOS[1:0] = 10_B or 11_B, the value of TAUDnCNTm is not loaded into TAUDnCDRm when the first valid TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

32.12.7.2 Equations

TAUDTTINm input pulse interval = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

32.12.7.3 Block Diagram and General Timing Diagram

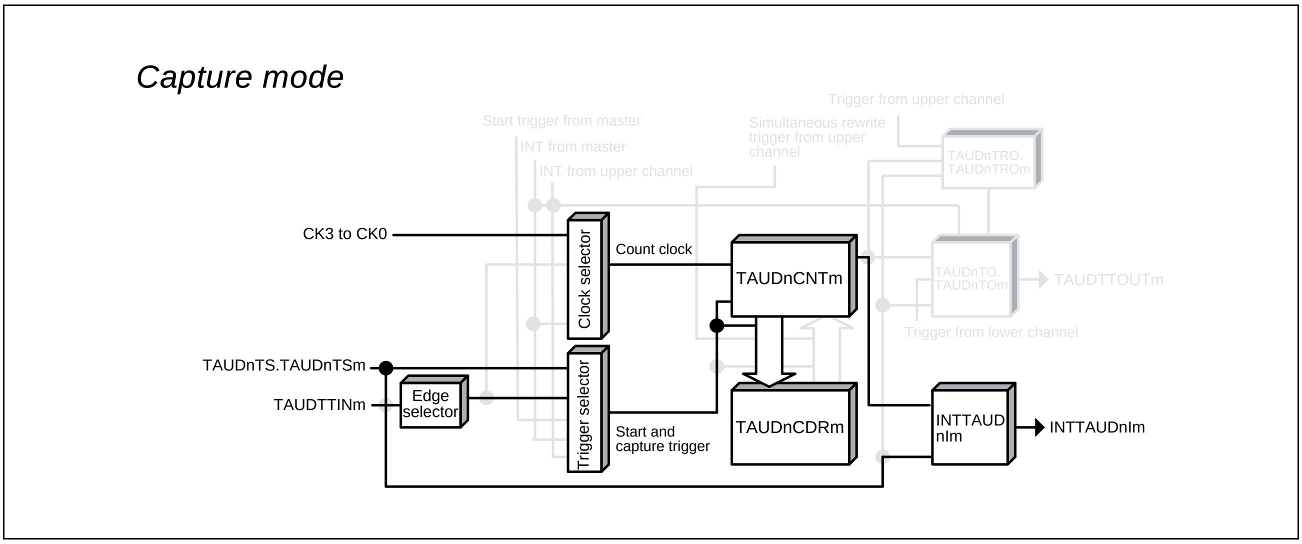


Figure 32.52 Block Diagram of TAUDTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 0).
- Falling edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- When a valid TAUDTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1 (TAUDnCMORm.TAUDnCOS[1:0] = 00_B).

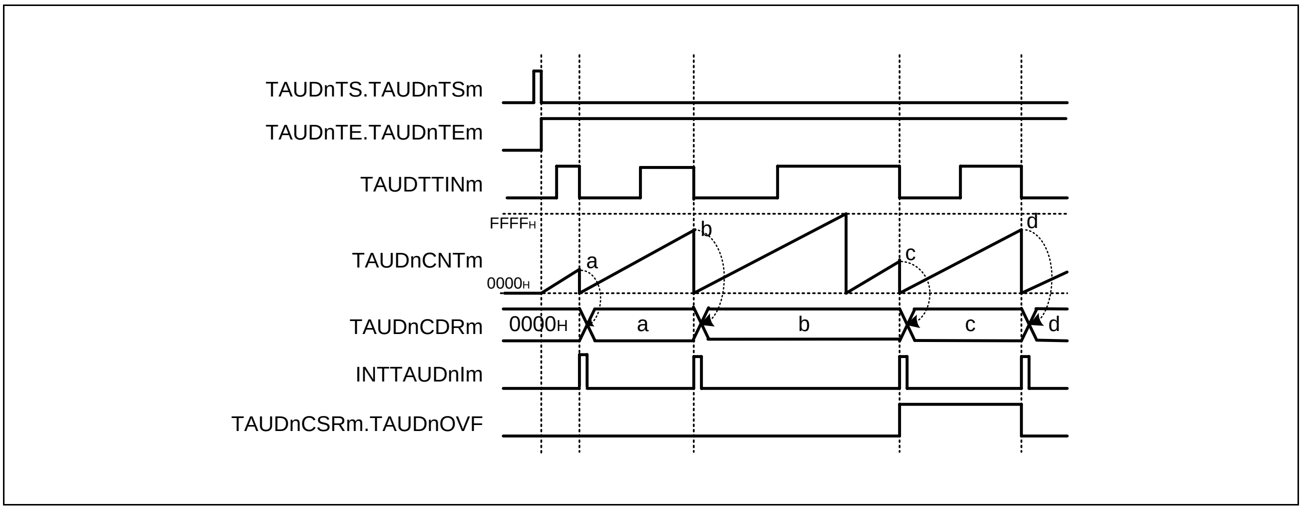


Figure 32.53 General Timing Diagram of TAUDTTINm Input Pulse Interval Measurement Function

32.12.7.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.82 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger.
7, 6	TAUDnCOS[1:0]	See **Table 32.81, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.83 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

Table 32.84 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.7.5 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

Table 32.85 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function**, and **Table 32.83, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
	During Operation	Detection of TAUDTTINm edge The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV can be written to 1. (TAUDnCSRm.TAUDnOVF bit is cleared to 0.)	TAUDnCNTm starts to count up from 0000_H. When a TAUDTTINm valid edge is detected: ● TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and returns to 0000_H. ● INTTAUDnIm is then generated. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

32.12.7.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00B

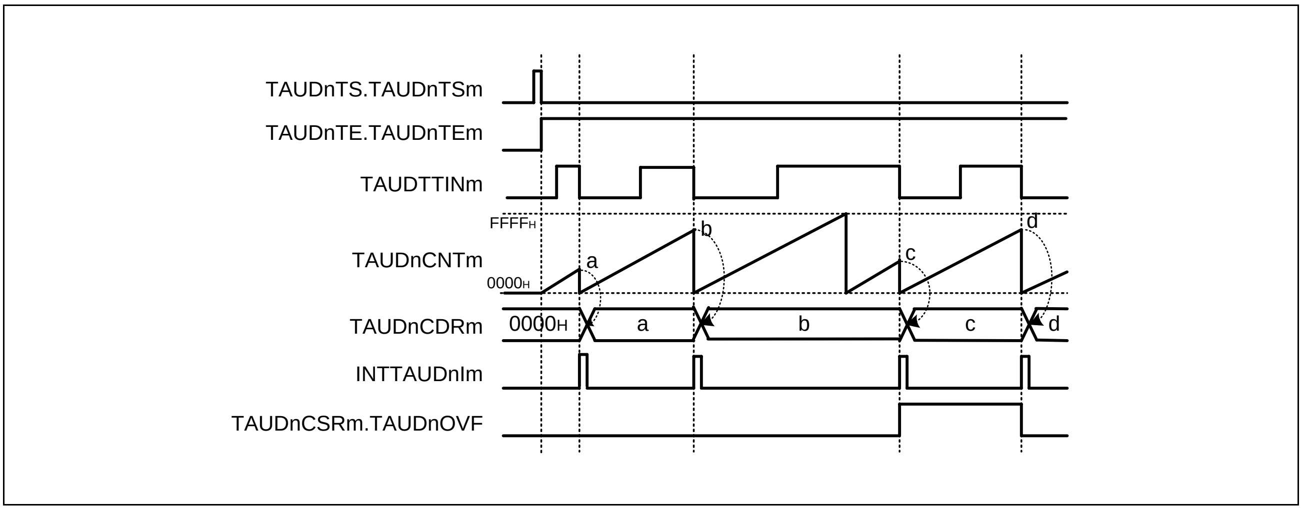


Figure 32.54 TAUDnCMORm.TAUDnCOS[1:0] = 00B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

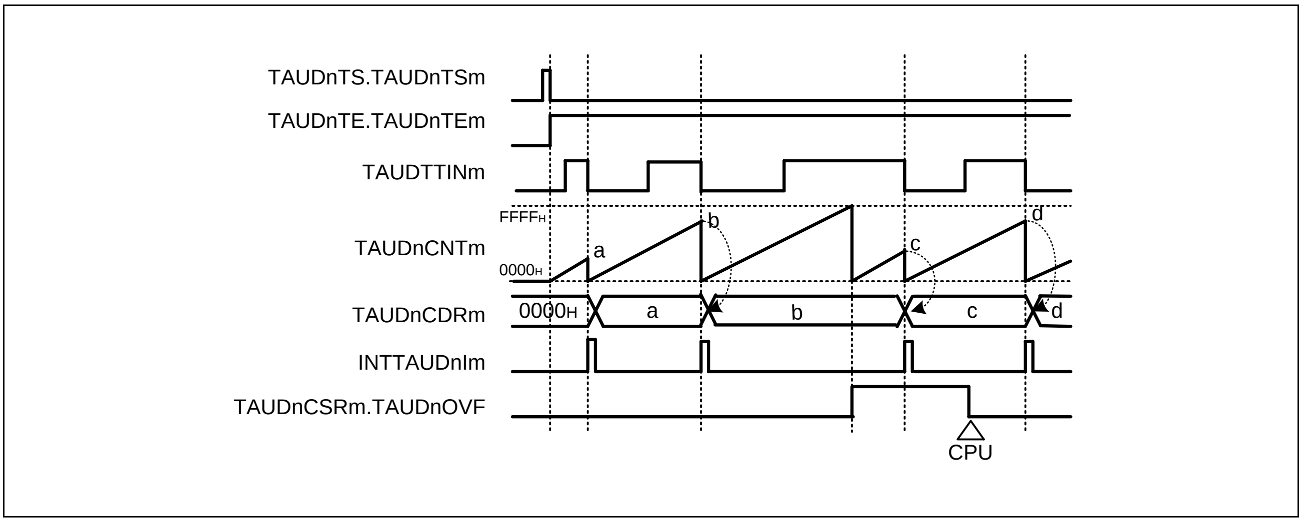


Figure 32.55 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to1.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

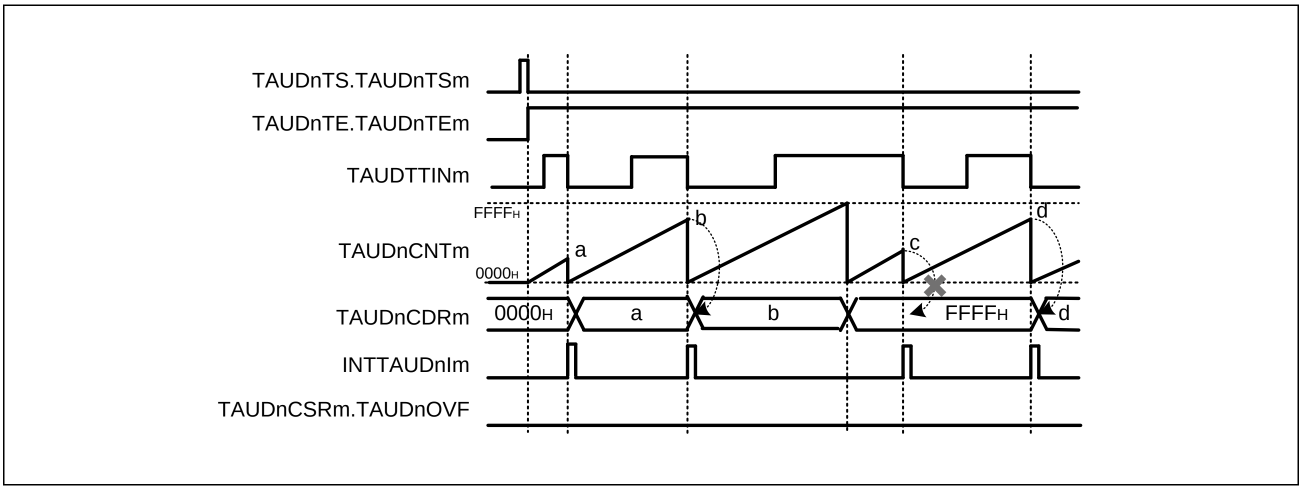


Figure 32.56 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

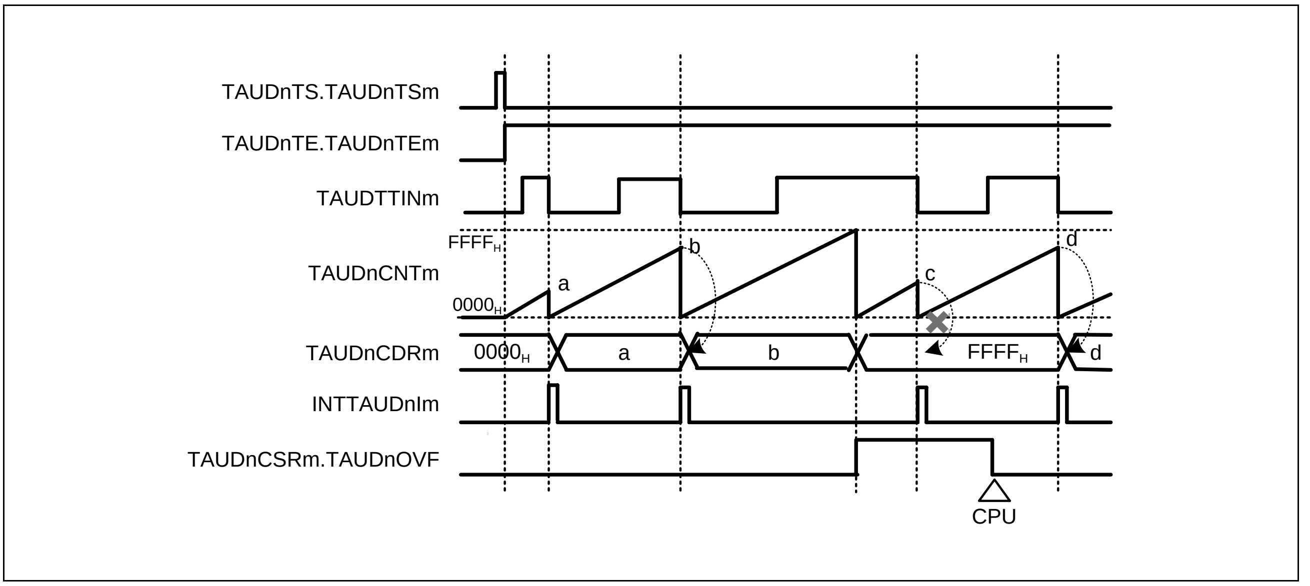


Figure 32.57 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

32.12.8 TAUDTTINm Input Signal Width Measurement Function

32.12.8.1 Overview

Summary

This function measures the width of a TAUDTTINm signal, by starting the count at one edge of TAUDTTINm and capturing the count value at the other edge.

Prerequisites

- The operating mode should be set to capture and one-count mode. See **Table 32.87, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function**.
- TAUDTTOUTm is not used with this function.
- TAUDnCMORm.TAUDnMD0 should be set to 0.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDTTINm start edge is detected, the counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm stop edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter retains its value (TAUDnCDRm +1) and awaits the next valid TAUDTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUDTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 32.86 Effects of Overflow

TAUDnCMORm. TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input Stop Edge is Detected	
	TAUDnCDRm	TAUDnCSRm. TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm. TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm stops counting TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, overflow bit TAUDnCSRm.TAUDnOVF can be cleared only by setting TAUDnCSCm.TAUDnCLOV to 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the width of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

This function cannot be forcibly restarted.

NOTE

When TAUDnCMORm.TAUDnCOS[1] = 1, the value of TAUDnCNTm is not loaded to TAUDnCDRm when the first valid TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

32.12.8.2 Equations

TAUDTTINm input signal width = count clock cycle × [(TAUDnCSRm.TAUDnOVF × (FFFF$_H$ + 1)) + TAUDnCDRm capture value + 1]

32.12.8.3 Block Diagram and General Timing Diagram

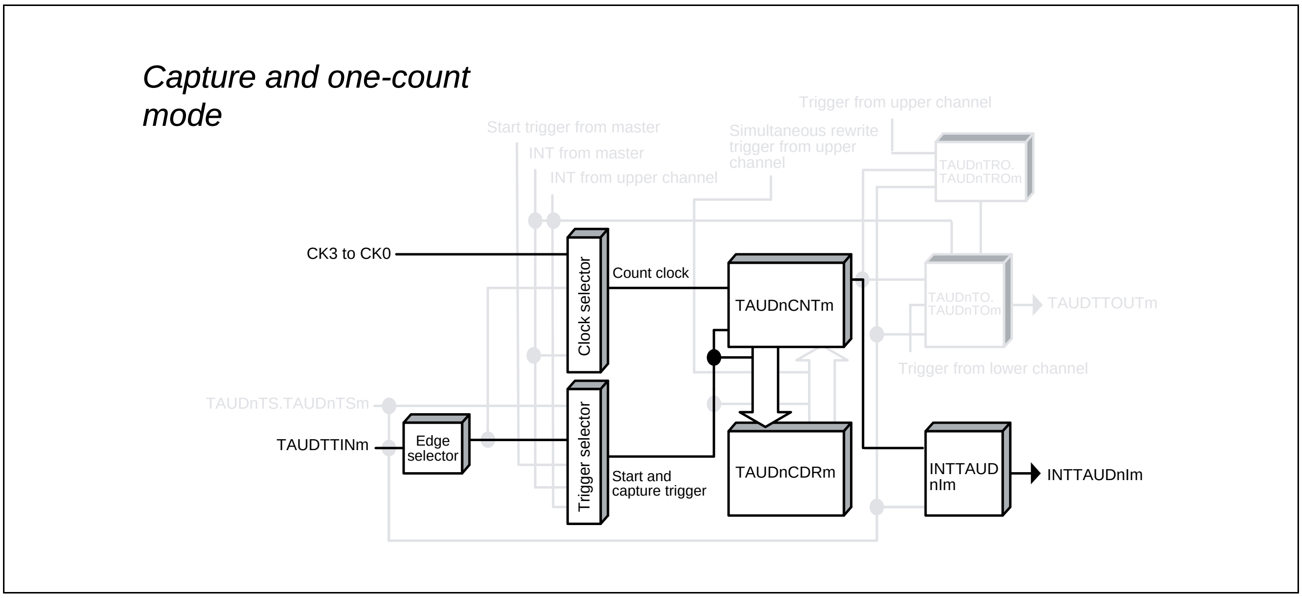


Figure 32.58 Block Diagram of TAUDTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11$_B$)
- When a valid TAUDTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1. (TAUDnCMORm.TAUDnCOS[1:0] = 00$_B$)

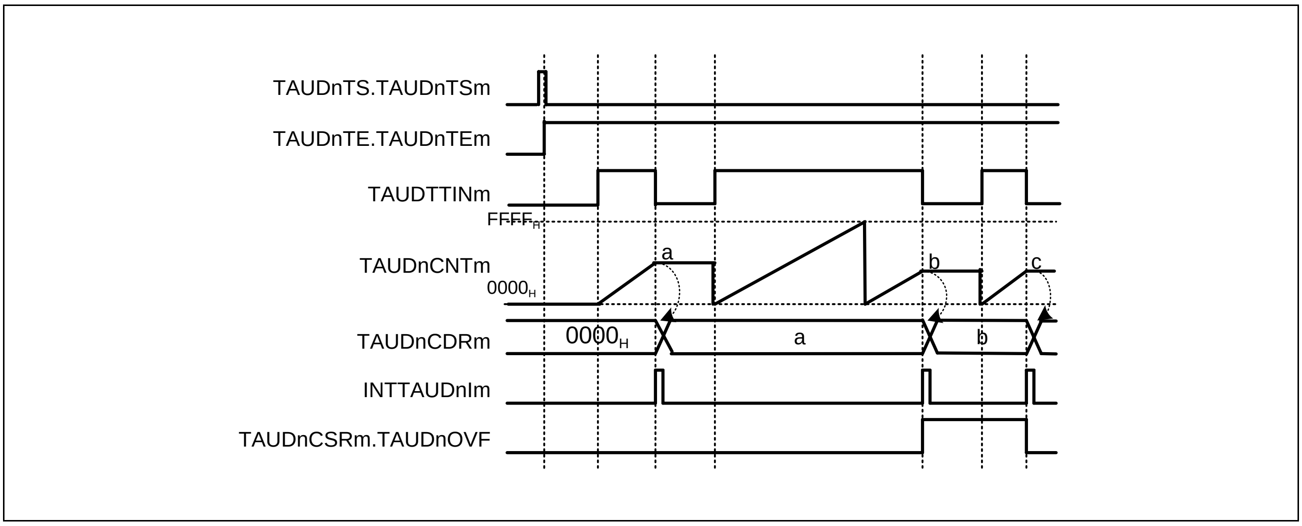


Figure 32.59 General Timing Diagram of TAUDTTINm Input Signal Width Measurement Function

32.12.8.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.87 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	See **Table 32.86, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0110: Capture and one-count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.88 Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

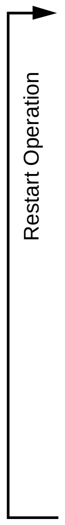
The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width measurement function. Therefore, these registers should be set to 0.

Table 32.89 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.8.5 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

Table 32.90 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.87, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function**, and **Table 32.88, Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a TAUDTTINm start edge is detected, TAUDnCNTm starts to count up.
During Operation	TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV bit can be set to 1.	TAUDnCNTm starts to count up from 0000_H. When TAUDTTINm valid edge is detected: ● TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and retains its value. INTTAUDnIm is then generated. Counting stops at the “value that transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

32.12.8.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

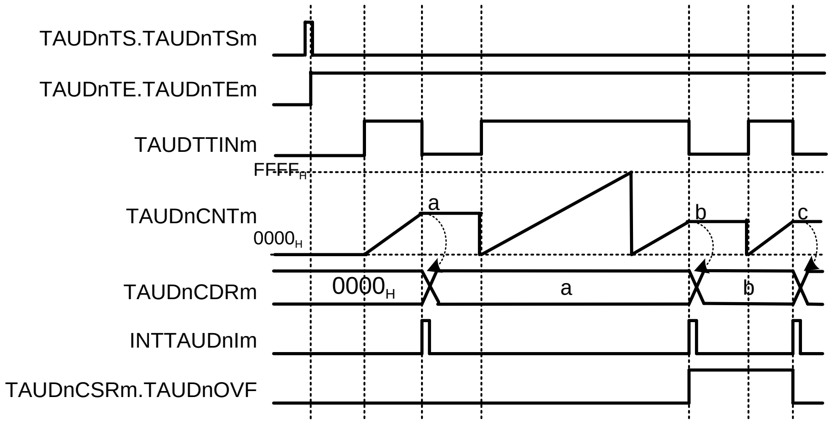


Figure 32.60 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01B

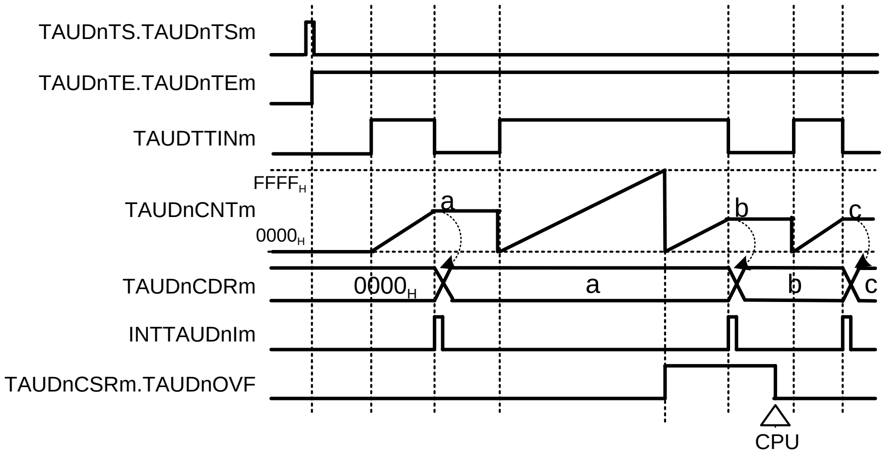


Figure 32.61 TAUDnCMORm.TAUDnCOS[1:0] = 01B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

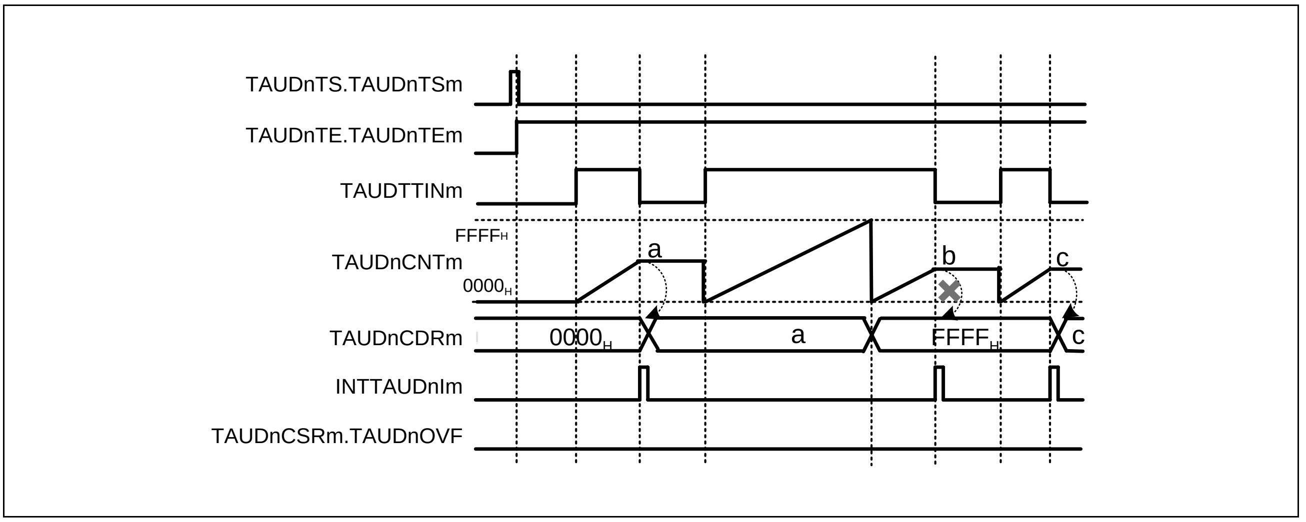


Figure 32.62 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

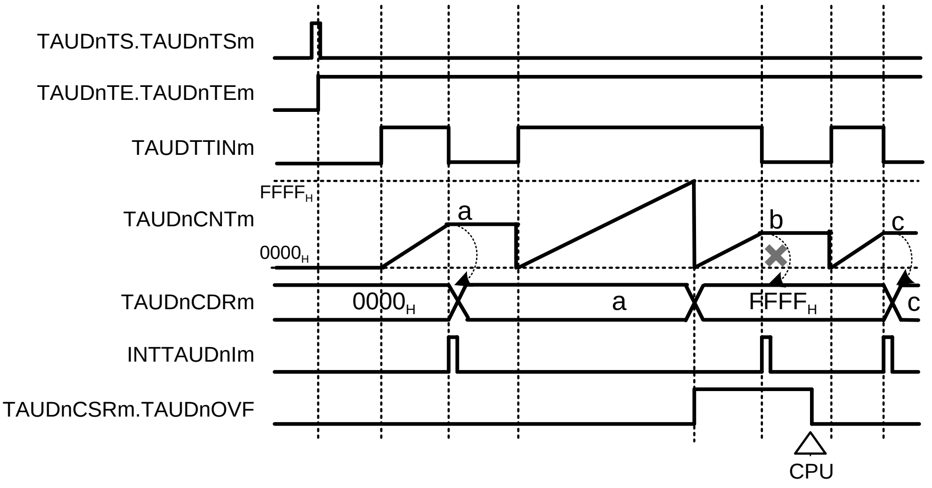


Figure 32.63 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

32.12.9 TAUDTTINm Input Position Detection Function

32.12.9.1 Overview

Summary

This function measures the input signal duration by capturing the count value at the valid edge of TAUDTTINm.

Prerequisites

- The operating mode should be set to count capture mode. (See **Table 32.91, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter starts counting from 0000_H. When a valid TAUDTTINm input edge is detected, the current value of TAUDnCNTm is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUDTTINm input signal is sampled at the frequency of the operation clock, specified by TAUDnCMORm.TAUDnCKS[1:0] bits.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.12.9.2 Equations

Functional duration at a TAUDTTINm input pulse =

count clock cycle × (TAUDnCDRm capture value + 1)

32.12.9.3 Block Diagram and General Timing Diagram

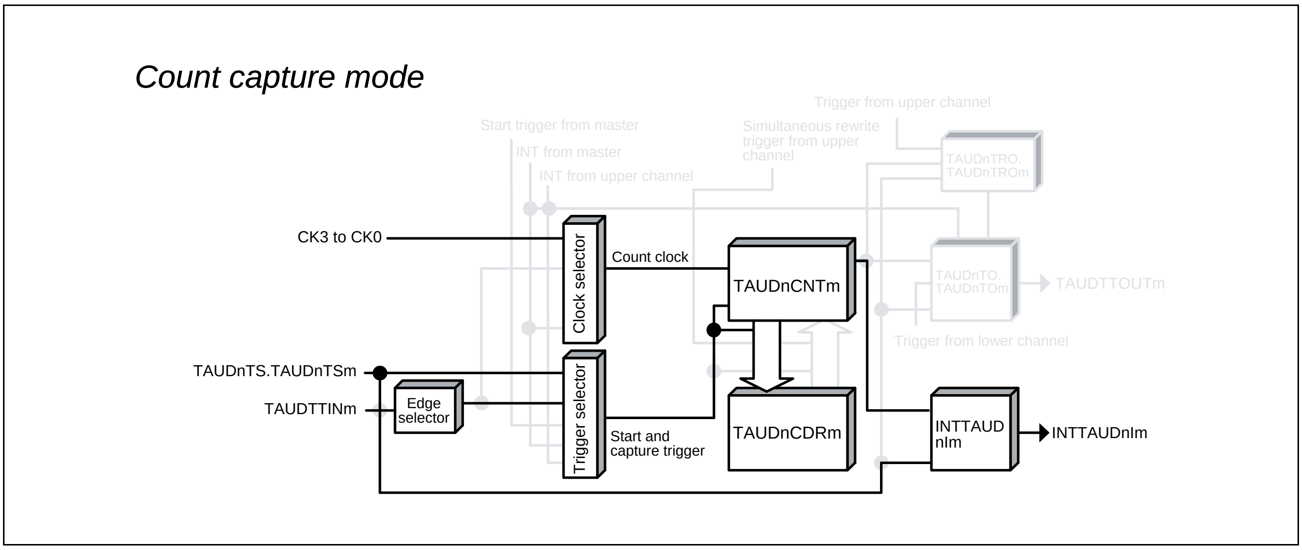


Figure 32.64 Block Diagram of TAUDTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

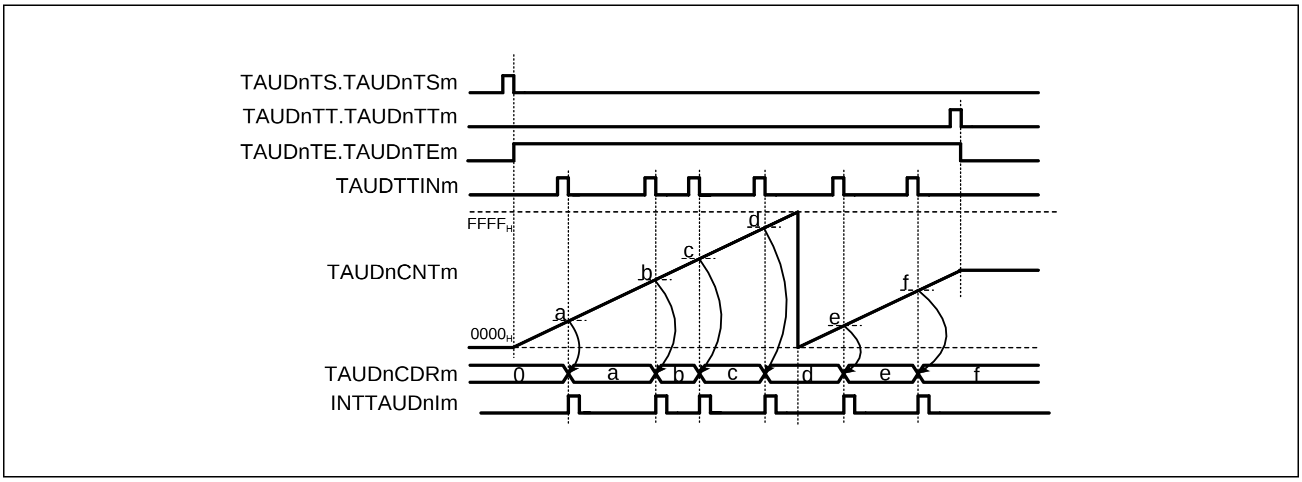


Figure 32.65 General Timing Diagram of TAUDTTINm Input Position Detection Function

32.12.9.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.91 Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external capture trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1011: Count capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.92 Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

The channel output mode is not used by this function.

(4) Simultaneous Rewrite

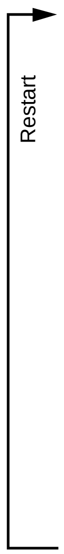
Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input position detection function. Therefore, these registers should be set to 0.

Table 32.93 Simultaneous Rewrite Settings for TAUDTTINm Input Position Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.9.5 Operating Procedure for TAUDTTINm Input Position Detection Function

Table 32.94 Operating Procedure for TAUDTTINm Input Position Detection Function



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.91, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function**, and **Table 32.92, Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs.
During Operation	The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm edge is detected: ● TAUDnCNTm transfers (captures) its own value to TAUDnCDRm. ● Outputs INTTAUDnIm. ● The counter is not cleared to 0000_H and TAUDnCNTm continues counting. Afterwards, this procedure is repeated. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

32.12.9.6 Specific Timing Diagrams

(1) Operation Stop and Restart

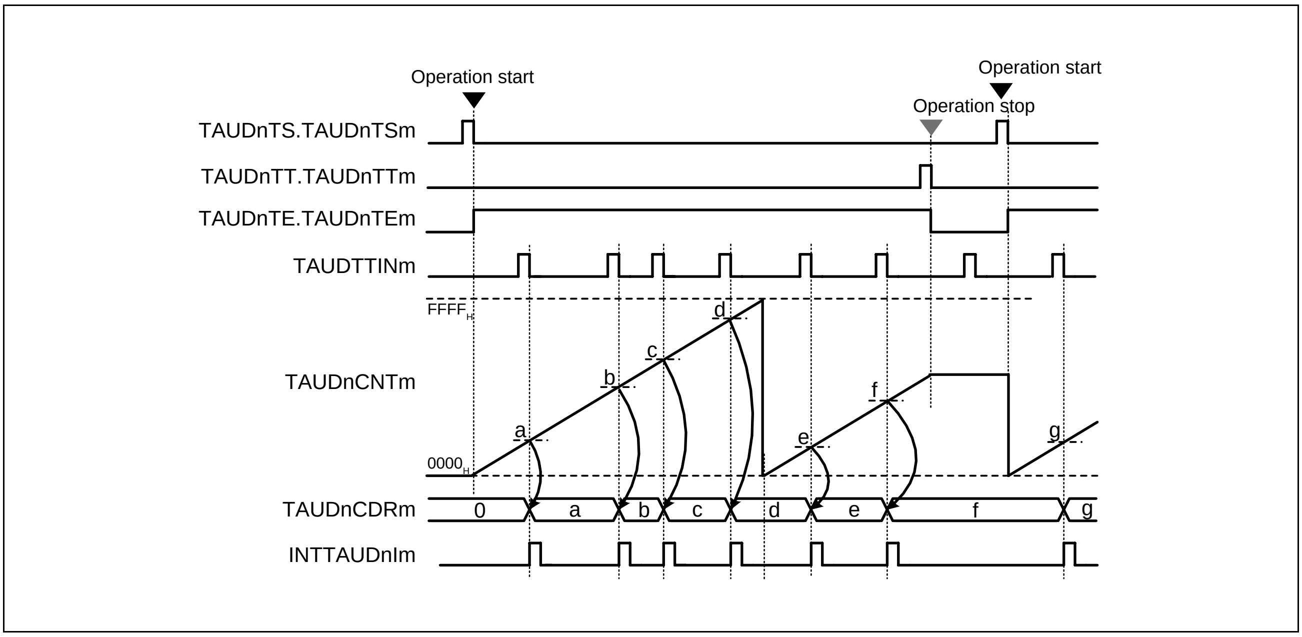


Figure 32.66 Operation Stop and Restart
(TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- The counter can stop operating by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

32.12.10 TAUDTTINm Input Period Count Detection Function

32.12.10.1 Overview

Summary

This function measures the cumulative width of a TAUDTTINm input signal.

Prerequisites

- The operating mode should be set to capture and gate count mode. (See **Table 32.95, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter awaits a valid TAUDTTINm input edge.

When a valid TAUDTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUDTTINm input stop edge is detected, the current TAUDnCNTm value is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The counter stops and retains its value (TAUDnCDRm + 1) until the next valid TAUDTTINm input start edge is detected.

When the next valid TAUDTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. TAUDTTINm input signal is sampled at the frequency of an operation clock set by the TAUDnCMORm.TAUDnCKS[1:0] bits.
2. As this function is to measure the TAUDTTINm input signal width, setting TAUDnTS.TAUDnTSm to 1 is disabled while TAUDnTE.TAUDnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

32.12.10.2 Equations

Cumulative TAUDTTINm input width =
count clock cycle × (TAUDnCDRm capture value + 1)

32.12.10.3 Block Diagram and General Timing Diagram

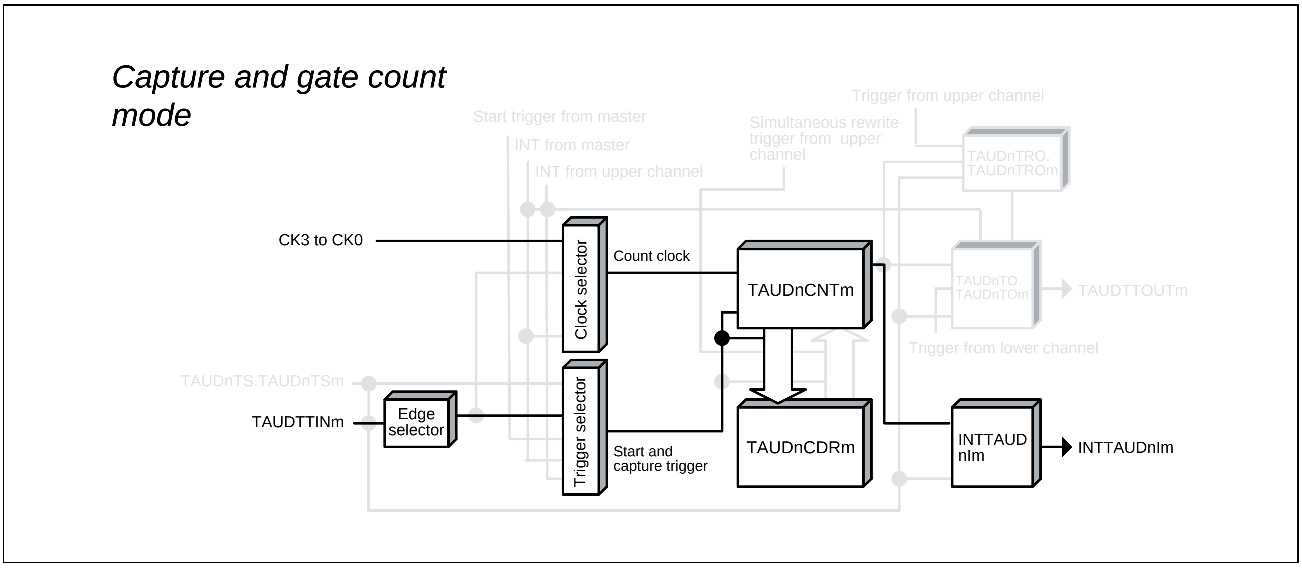


Figure 32.67 Block Diagram of TAUDTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

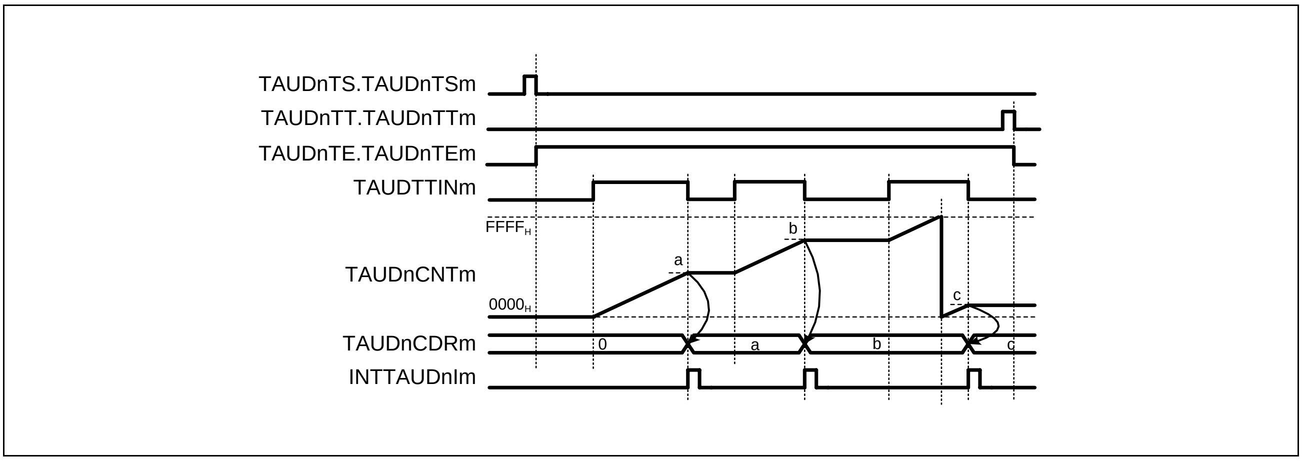


Figure 32.68 General Timing Diagram of TAUDTTINm Input Period Count Detection Function

32.12.10.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.95 Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1101: Capture and gate count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.96 Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

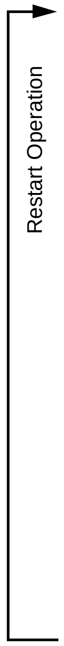
Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input period count detection function. Therefore, these registers should be set to 0.

Table 32.97 Simultaneous Rewrite Settings for TAUDTTINm Input Period Count Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.10.5 Operating Procedure for TAUDTTINm Input Period Count Detection Function

Table 32.98 Operating Procedure for TAUDTTINm Input Period Count Detection Function



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.95, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function**, and **Table 32.96, Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.
During Operation	Detection of TAUDTTINm edge The TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time.	When a TAUDTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUDnCNTm starts counting up from the stop value. When TAUDnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDnCDRm and INTTAUDnIm is generated. Counting stops at the “value transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

32.12.10.6 Specific Timing Diagrams

(1) Operation stop and restart

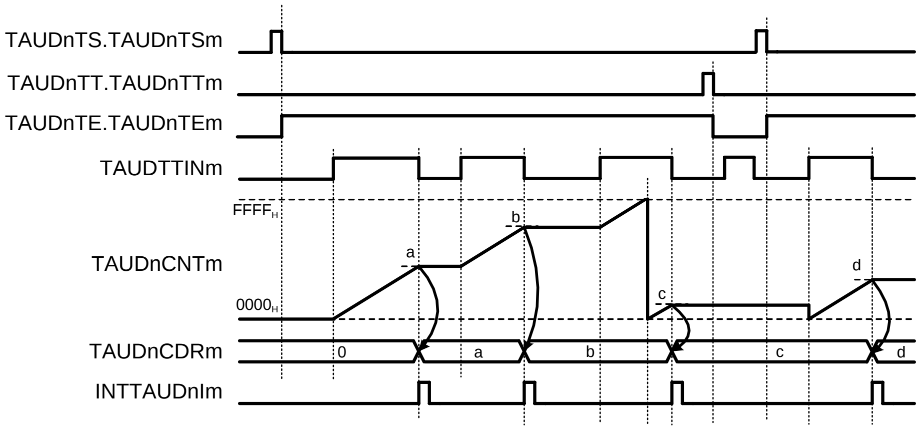


Figure 32.69 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter is stopped, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

32.12.11 TAUDTTINm Input Pulse Interval Judgment Function

32.12.11.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUDnCNTm) and the value in the channel data register (TAUDnCDRm) when a TAUDTTINm input pulse occurs. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge mode. See **Table 32.99, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function**.
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid edge is detected or TAUDnTS.TAUDnTSm is set to 1, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. TAUDnCNTm reloads the value of TAUDnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUDTTINm valid edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:

- If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
- If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.

32.12.11.2 Block Diagram and General Timing Diagram

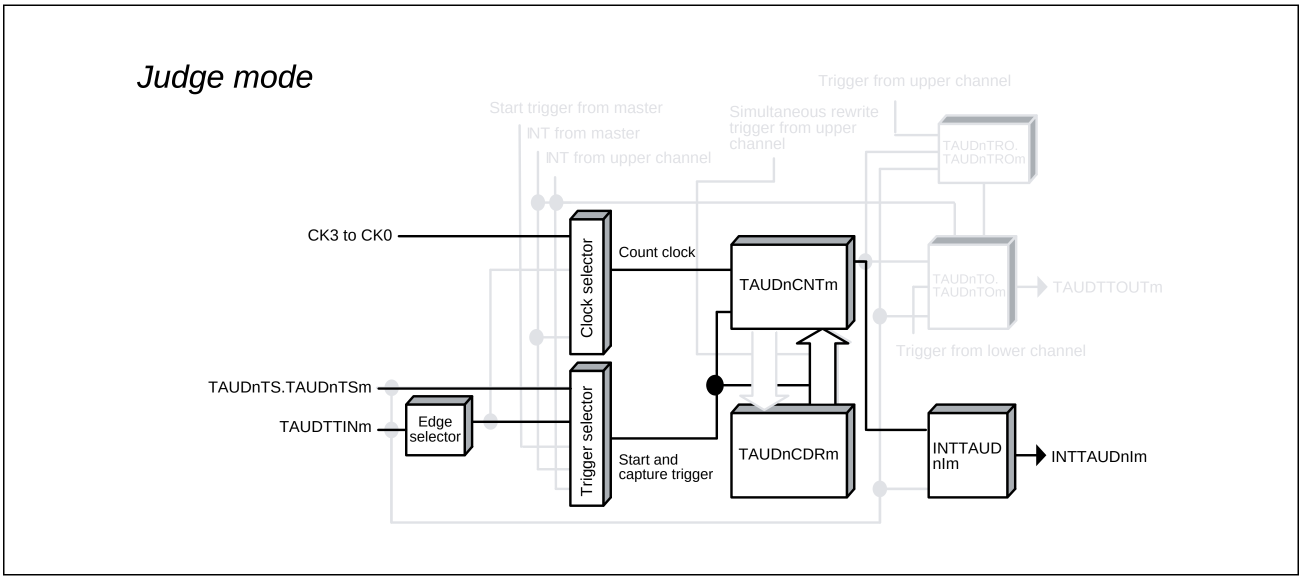


Figure 32.70 Block Diagram of TAUDTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

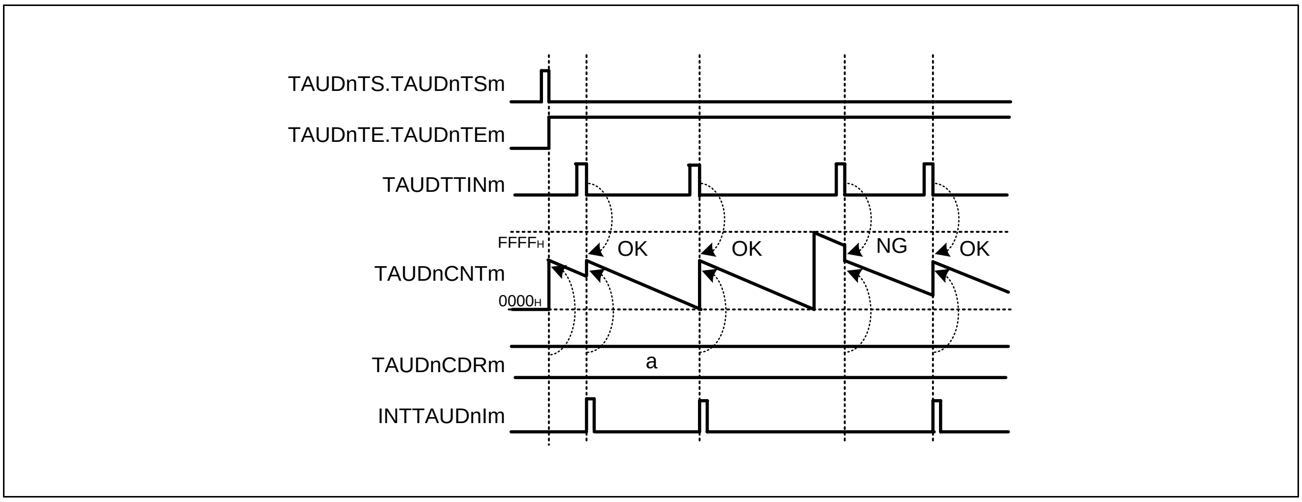


Figure 32.71 General Timing Diagram of TAUDTTINm Input Pulse Interval Judgment Function

32.12.11.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.99 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0001: Judge mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.100 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval judgment function. Therefore, these registers should be set to 0.

Table 32.101 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.11.4 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

Table 32.102 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.99, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function**, and **Table 32.100, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm.
During Operation	The following register can be changed at any time: ● TAUDnCDRm register	When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. If a TAUDTTINm input edge is detected, then TAUDnCNTm starts to count down from the value of TAUDnCDRm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

32.12.12 TAUDTTINm Input Signal Width Judgment Function

32.12.12.1 Overview

Summary

This function compares the count value (TAUDnCNTm) for the high or low level width of a TAUDTTINm input signal and the TAUDnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDnIm.

Prerequisites

- The operating mode should be set to judge and one-count mode. (See **Table 32.103, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDTTINm input start edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid stop edge is detected, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. The counter TAUDnCNTm retains its value until the next valid TAUDTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUDTTINm stop edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. The counter then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

- The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:
  - If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
  - If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.
- The TAUDnCMURm.TAUDnTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B), TAUDTTINm rising edge is used as a start edge and TAUDTTINm falling edge as a stop edge.
  - For low width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 10_B), TAUDTTINm falling edge is used as a start edge and TAUDTTINm rising edge as a stop edge.
- Setting TAUDnTS.TAUDnTSm to 1 is prohibited during operation.

32.12.12.2 Block Diagram and General Timing Diagram

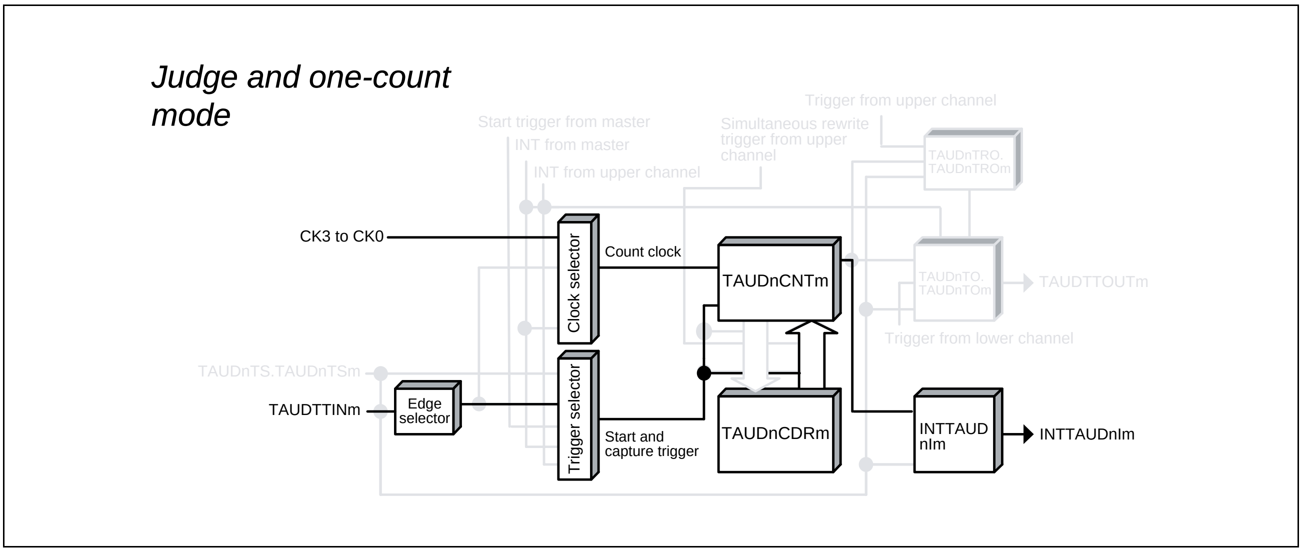


Figure 32.72 Block Diagram of TAUDTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm (TAUDnCMORm.TAUDnMD0 = 0).
- TAUDTTINm valid start edge = rising edge, TAUDTTINm valid stop edge = falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

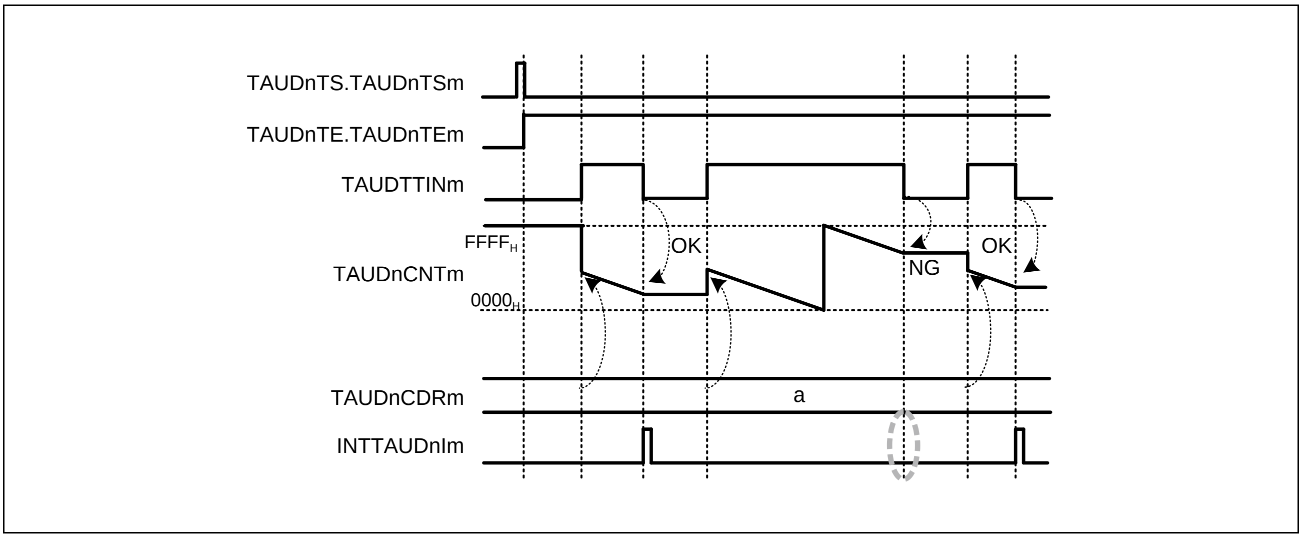


Figure 32.73 General Timing Diagram of TAUDTTINm Input Signal Width Judgment Function

32.12.12.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.103 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0111: Judge and one-count mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.104 Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width judgment function. Therefore, these registers should be set to 0.

Table 32.105 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.12.4 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

Table 32.106 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.103, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function**, and **Table 32.104, Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.
During Operation	The following register can be changed at any time: ● TAUDnCDRm register	Upon detection of a TAUDTTINm start edge, TAUDnCNTm starts count down from the value of TAUDnCDRm. When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

32.12.13 Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

32.12.13.1 Overview

Summary

This function measures the width of an individual TAUDTTINm input signal. An interrupt is generated if the TAUDTTINm input width is longer than $FFFF_H + 1$.

Prerequisites

- The operation mode must be set to One-Count Mode (see **Table 32.107, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**).
- TAUDTTOUTm is not used for this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUDTTINm input start edge is detected, TAUDnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

32.12.13.2 Block Diagram and General Timing Diagram

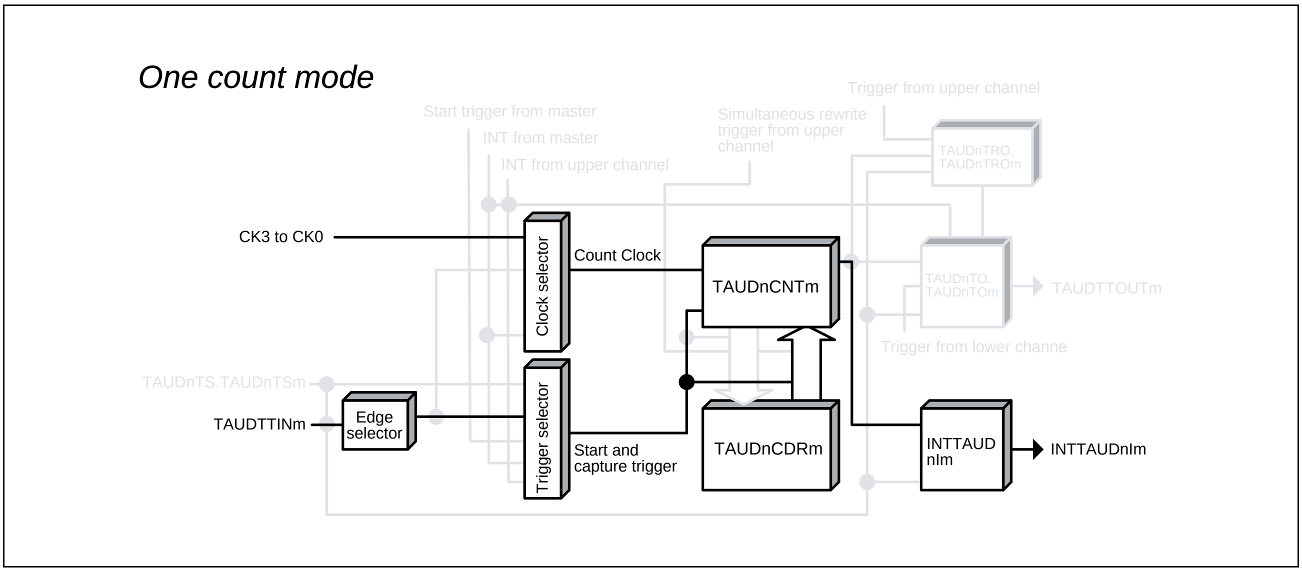


Figure 32.74 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

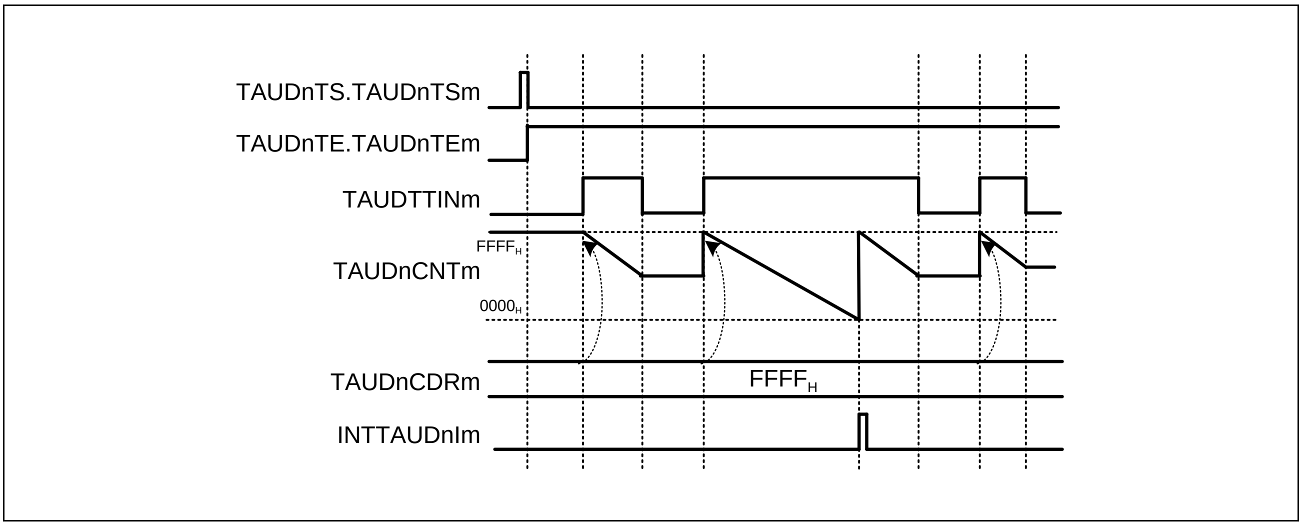


Figure 32.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

32.12.13.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.107 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables the start trigger during operation

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.108 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 32.109 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Table 32.110 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.107, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**, and **Table 32.108, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**. Set the value of TAUDnCDRm register to FFFF_{H}.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value (FFFF_{H}).
During Operation	The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_{H}: INTTAUDnIm is generated. When TAUDTTINm input stop edge is detected during count operation: ● TAUDnCNTm stops and retains its current value. When TAUDTTINm input start edge is detected while the counter is stopped: ● TAUDnCNTm loads the TAUDnCDRm value (FFFF_{H}) again, and continues to count down. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

32.12.14 Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

32.12.14.1 Overview

Summary

This function measures the cumulative width of a TAUDTTINm input signal. If the cumulative TAUDTTINm input width is longer than $FFFF_H$, an interrupt is generated and an overflow interrupt can be output.

Prerequisites

- The operation mode must be set to Gate Count Mode, (see **Table 32.111, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**).
- TAUDTTOUTm is not used with this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUDTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is loaded to TAUDnCNTm and the counter continues to count down until a TAUDTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

32.12.14.2 Block Diagram and General Timing Diagram

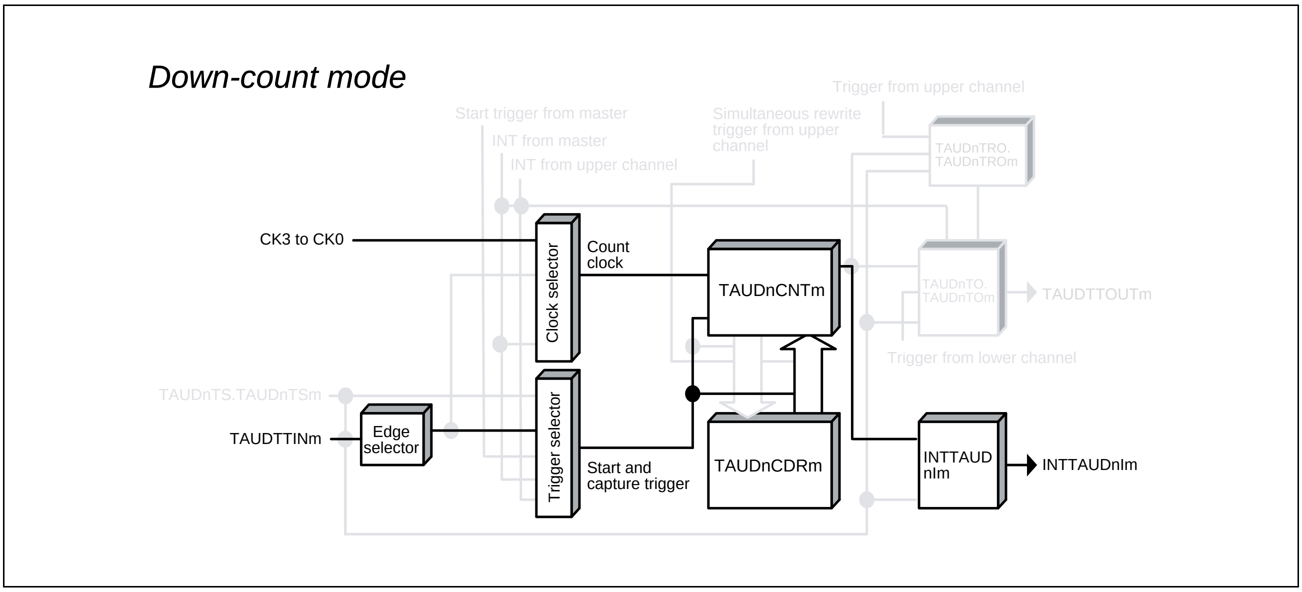


Figure 32.76 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

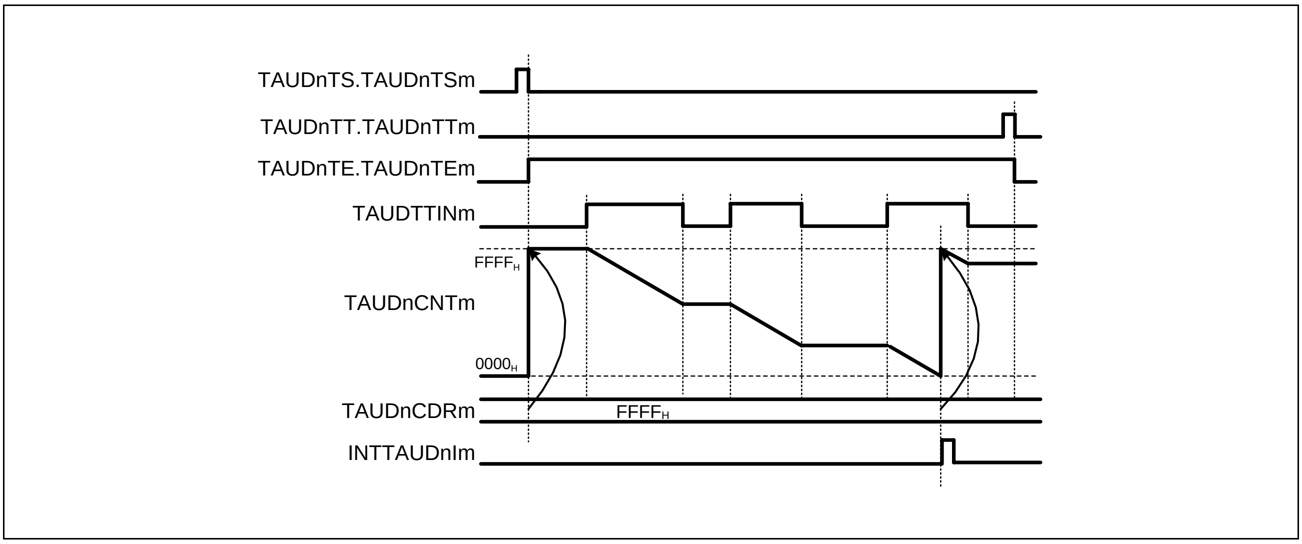


Figure 32.77 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

32.12.14.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.111 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1100: Gate count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.112 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 32.113 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.14.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Table 32.114 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Restart Operation

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 32.111, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**, and **Table 32.112, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**. Set the value of TAUDnCDRm register to FFFF$_{H}$.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value (FFFF$_{H}$).
During Operation	The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000$_{H}$: ● INTTAUDnIm is generated. ● TAUDnCNTm loads the TAUDnCDRm value (FFFF$_{H}$) and continues to count down. When TAUDTTINm input stop edge is detected during count operation: ● TAUDnCNTm stops and retains the current value. When TAUDTTINm input start edge is detected while the counter is stopped: ● TAUDnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

32.12.15 One-Phase PWM Output Function

32.12.15.1 Overview

Summary

This function adds dead time to a TAUDTTINm input signal. The resulting PWM signal is output via TAUDTTOUTm of the channel and TAUDTTOUTm of upper channels.

Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDnTDE.TAUDnTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See **Table 32.116, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**.)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWM output. (See **Section 32.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt occurs. The counter is reset to $FFFF_H$ and waits for the next valid TAUDTTINm input start edge.

Table 32.115 TAUDTTOUTm to which Dead Time is Added and State of TAUDTTINm

TAUDnCMURm. TAUDnTIS[1:0]	TAUDnTOL. TAUDnTOLm	TAUDTTOUTm to which Dead Time is Added	TAUDnTDL. TAUDnTDLm	TAUDTTINm State when Added
10	0	TAUDTTOUTm low	0	High
			1	Low
	1	TAUDTTOUTm high	0	High
			1	Low
11	0	TAUDTTOUTm low	0	Low
			1	High
	1	TAUDTTOUTm high	0	Low
			1	High

Conditions

- TAUDnCMURm.TAUDnTIS[1:0] bits specify the type of width measurement:
  - TAUDnCMURm.TAUDnTIS[1:0] = 10_B: Uses both edges as valid edges for detection (Low width measurement).
  - TAUDnCMURm.TAUDnTIS[1:0] = 11_B: Uses both edges as valid edges for detection (High width measurement).
- The TAUDnTDL.TAUDnTDLm bit specifies the operation of TAUDTTOUTm for each channel when an interrupt or valid TAUDTTINm edge is detected on the lower channel:
  - If TAUDnTDL.TAUDnTDLm = 0, an interrupt is used as a TAUDTTOUTm set trigger and a valid TAUDTTINm edge as a TAUDTTOUTm reset trigger.
  - If TAUDnTDL.TAUDnTDLm = 1, a valid TAUDTTINm edge is used as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
- This function cannot make a forced restart.

32.12.15.2 Block Diagram and General Timing Diagram

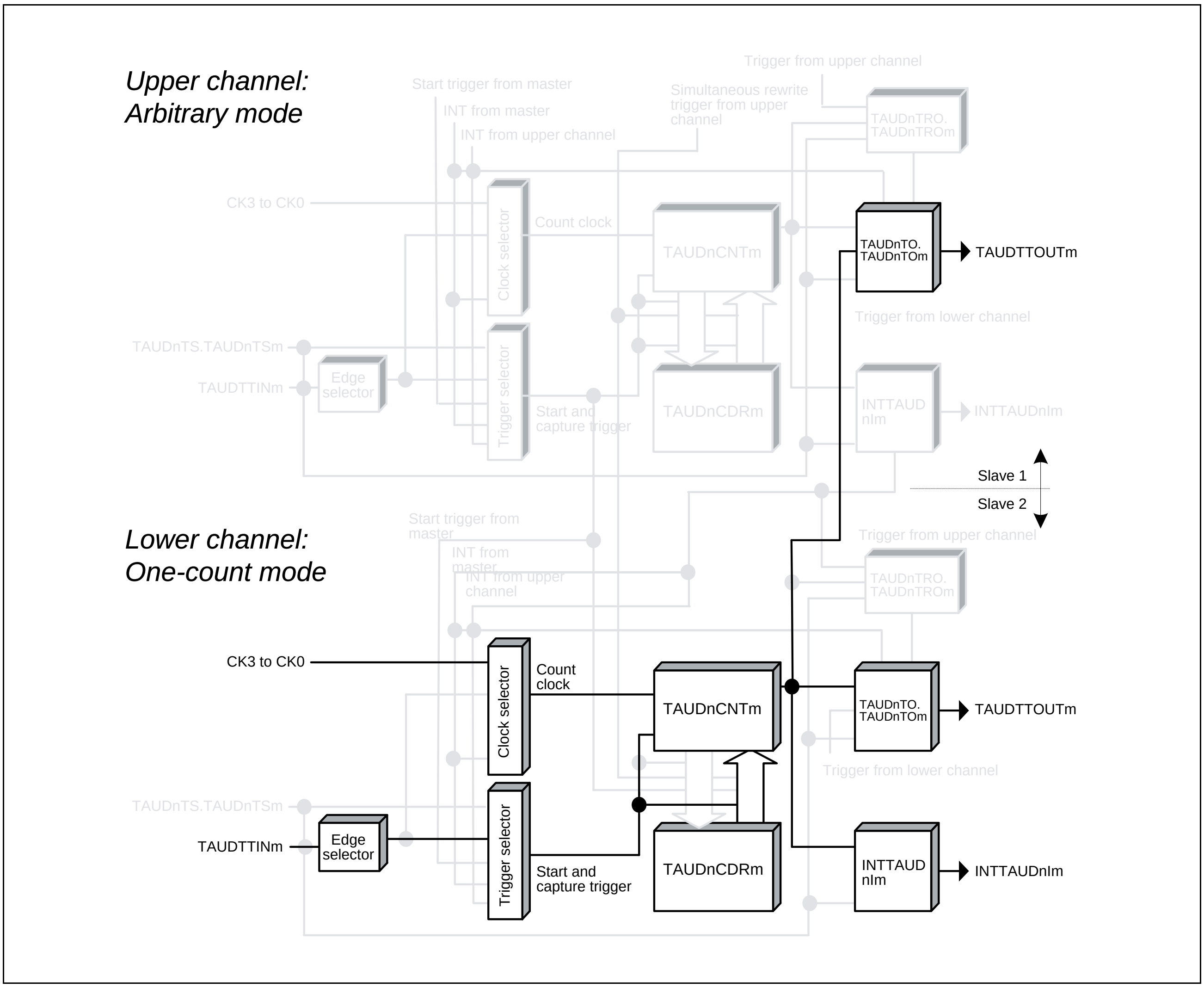


Figure 32.78 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

This setting considers a duty as an active high.

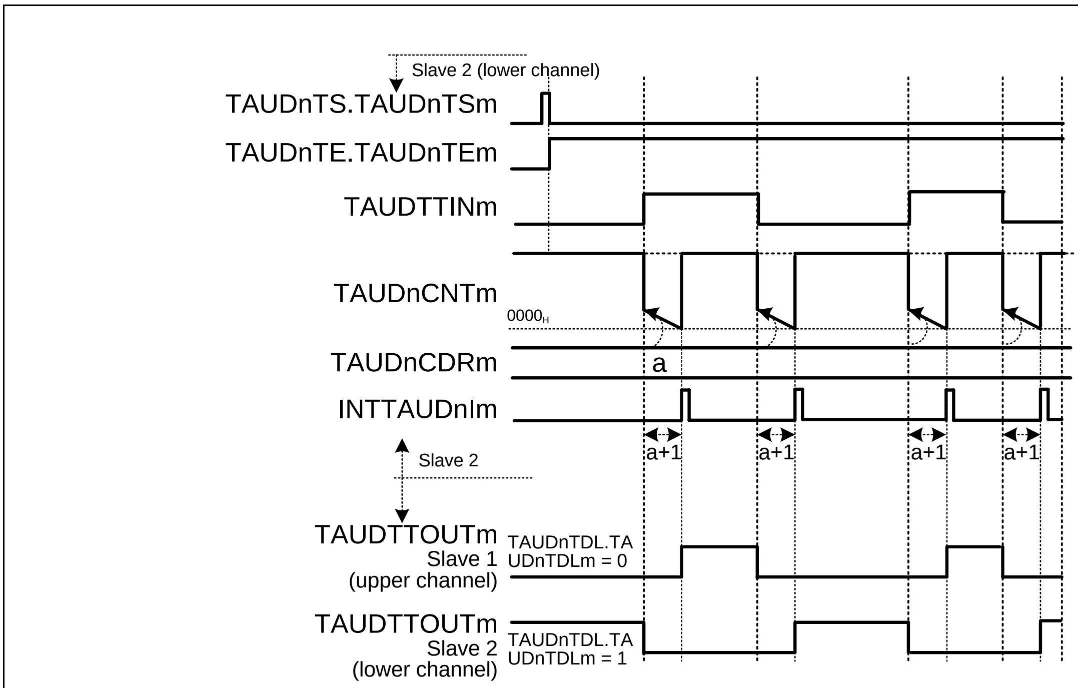


Figure 32.79 General Timing Diagram of One-Phase PWM Output Function

32.12.15.3 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.116 Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Lower Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.117 Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode for Lower Channels

Table 32.118 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from upper channels.

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the one-phase PWM output function. Therefore, these registers should be set to 0.

Table 32.119 Simultaneous Rewrite Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.12.15.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

TAUDnCMORm register for upper channels can be set arbitrarily.

(2) TAUDnCMURm for Upper Channels

TAUDnCMURm register for upper channels can be set arbitrarily.

(3) Channel Output Mode for Upper Channels

Table 32.120 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from lower channels.

(4) Simultaneous Rewrite for Upper Channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

32.12.15.5 Operating Procedure for One-phase PWM Output Function

Table 32.121 Operating Procedure for One-phase PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Table 32.116, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**, and **Table 32.117, Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function**. Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Section 32.12.15.4, Register Settings for Upper Channels**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 32.118, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTOE.TAUDnTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUDnTS.TAUDnTSm = 1 for slave channel 2. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 (slave channel 2) and TAUDnCNTm waits for detection of TAUDTTINm start edge. TAUDnCNTm loads TAUDnCDRm value.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm of slave channel 2 counts down. When the counter reaches 0000_H: ● INTTAUDnIm is generated. ● TAUDnCNTm stops counting. TAUDTTOUTm is changed by a TAUDTTINm edge detection signal and slave channel 2 INTTAUDnIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm = 1 for slave channel 2. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation

32.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDnTRO.TAUDnTROm bit in real time.

32.13.1 Real-Time Output Function Type 1

32.13.1.1 Overview

Summary

This function outputs a value of the TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 32.122, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 32.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter of the upper channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of the data register of the upper channel (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUDnIm is generated and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1). TAUDnCNTm then reloads the TAUDnCDRm value to continue operation subsequently.

The TAUDTTOUTm signal changes only when an interrupt is generated, and when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel’s TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.

- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.13.1.2 Equations

INTTAUDnIm generation cycle = count clock cycle × (TAUDnCDRm value + 1)

32.13.1.3 Block Diagram and General Timing Diagram

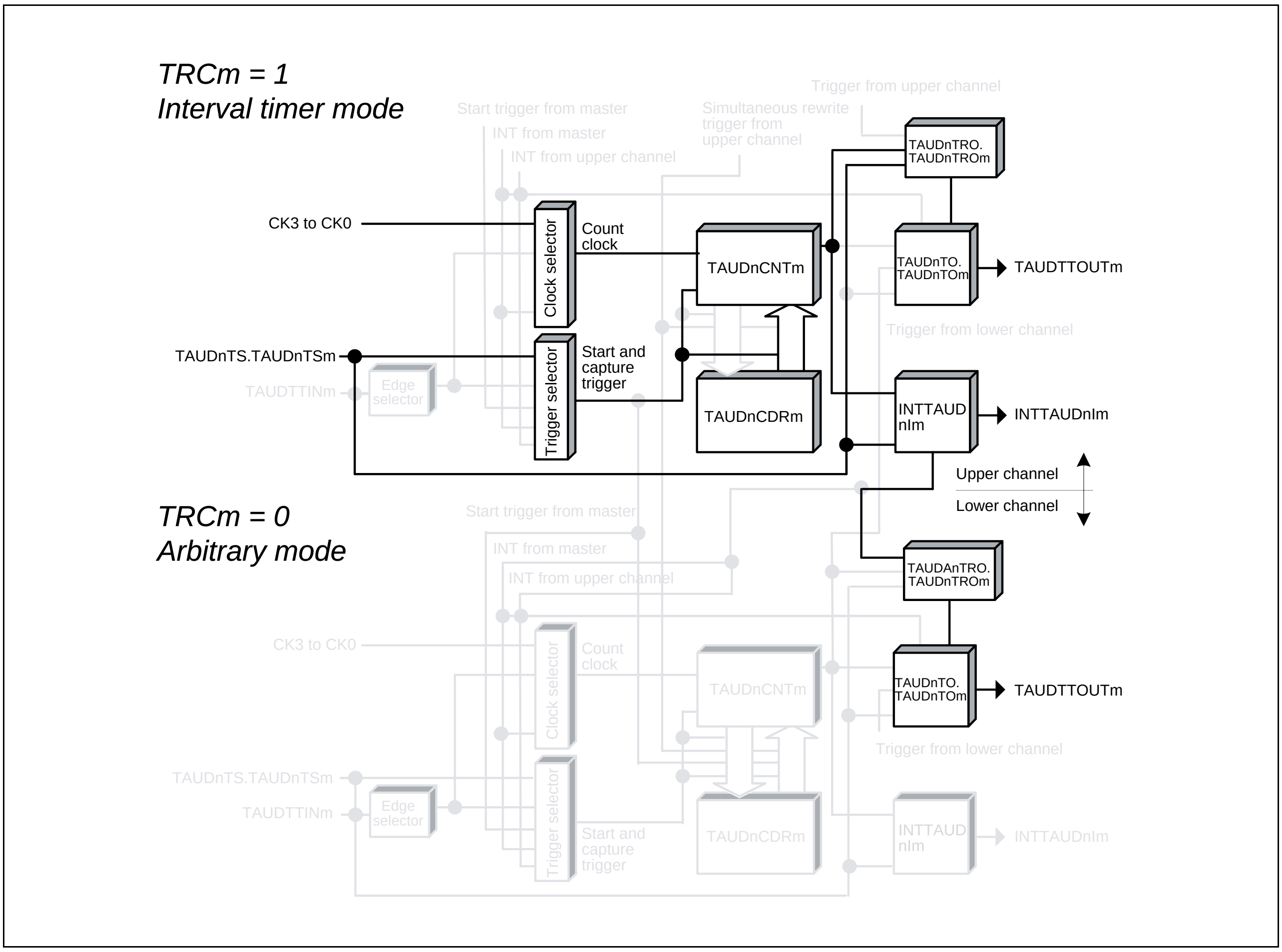


Figure 32.80 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

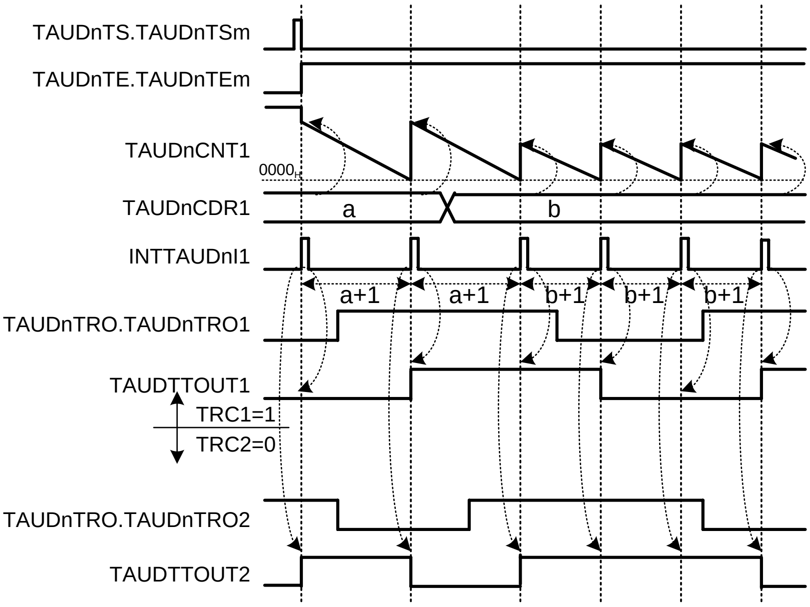


Figure 32.81 General Timing Diagram of Real-Time Output Function Type 1

32.13.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.122 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS [1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.123 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Upper Channels

Table 32.124 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

Table 32.125 Simultaneous Rewrite Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.13.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for Lower Channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel Output Mode for Lower Channels

Table 32.126 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

32.13.1.6 Operating Procedure for Real-Time Output Function Type 1

Table 32.127 Operating Procedure for Real-Time Output Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 32.122, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**, and **Table 32.123, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 32.13.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register (only channels with TAUDnTRC.TAUDnTRCm = 1) Set channel output mode by setting the control bits as described in **Table 32.124, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 32.126, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	TAUDnCDRm and TAUDnTRO.TAUDnTROm can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_H: ● The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. ● INTTAUDnIm is generated. ● TAUDTTOUTm outputs the current value of the real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. Both TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation

32.13.1.7 Specific Timing Diagrams

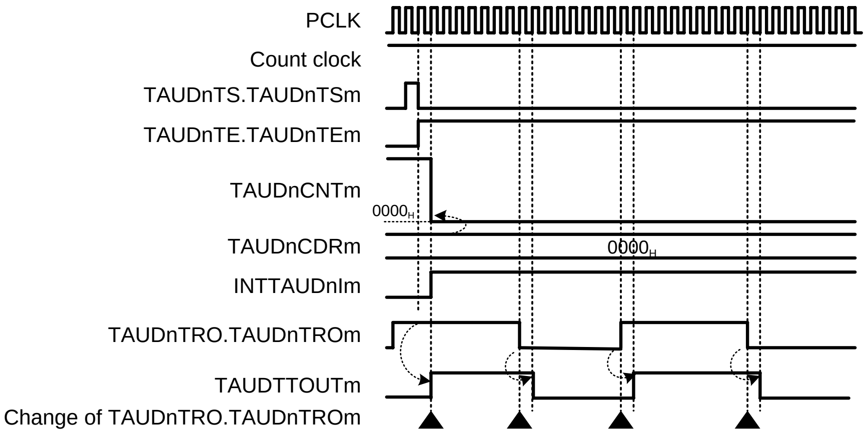


Figure 32.82 TAUDnCDRm = 0000$_H$, TAUDnCMORm.TAUDnMD0 = 1

- The value of TAUDTTOUTm changes according to the setting of TAUDnTRO.TAUDnTROm with a delay of one PCLK cycle.

32.13.2 Real-Time Output Function Type 2

32.13.2.1 Overview

Summary

This function outputs the value of TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDTTOUTm control of the other channels.
- The operating mode for the upper channel should be set to capture mode. (See **Table 32.128, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 32.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter for upper channels is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The counter starts to count up.

When a valid TAUDTTINm input edge is generated on one of upper channels, an interrupt occurs and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1).

The TAUDTTOUTm signal changes only when an interrupt is generated, and when TAUDTTOUTm value is different from the current value of TAUDnTRO.TAUDnTROm during the occurrence of the interrupt.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel’s TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.13.2.2 Block Diagram and General Timing Diagram

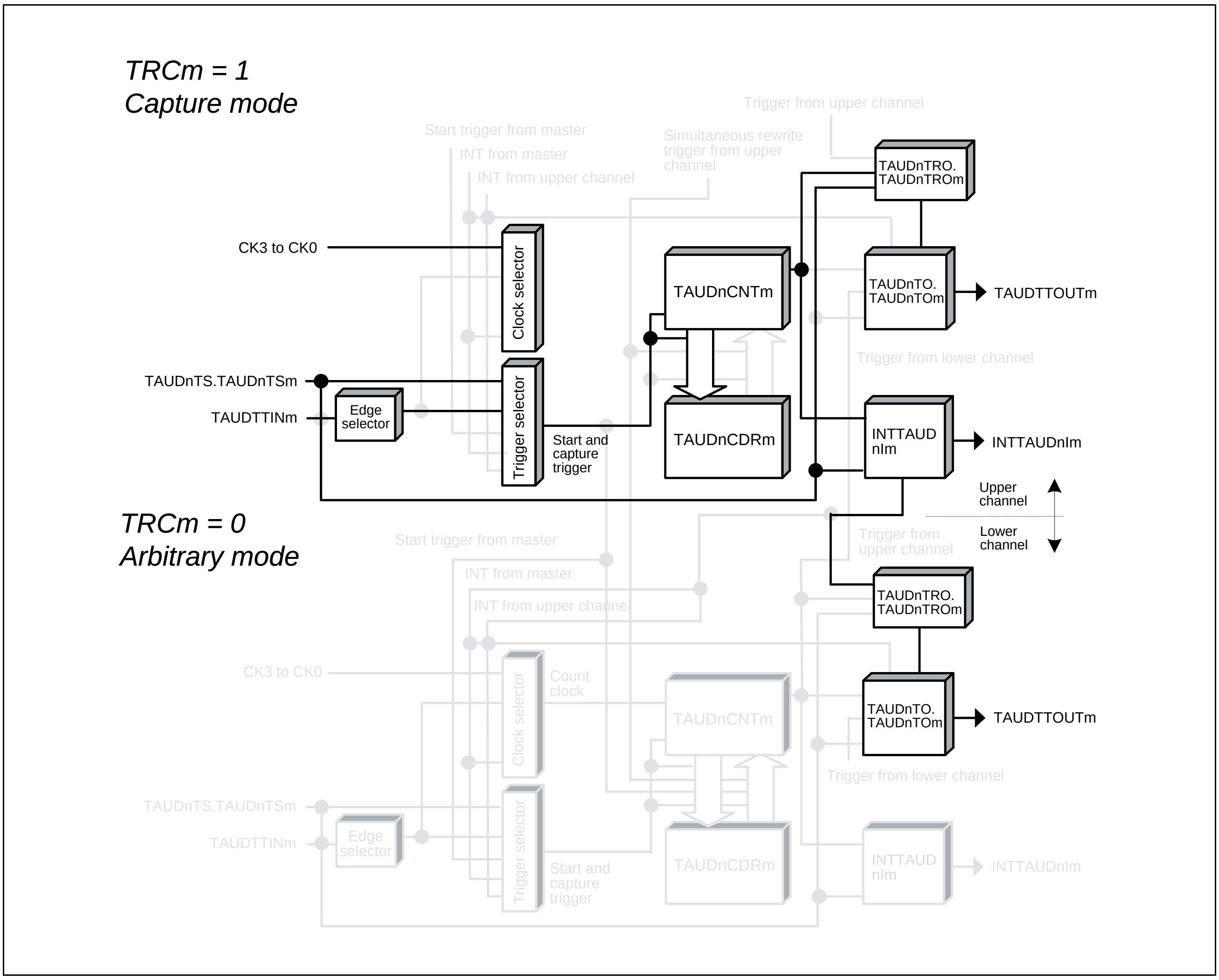


Figure 32.83 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)

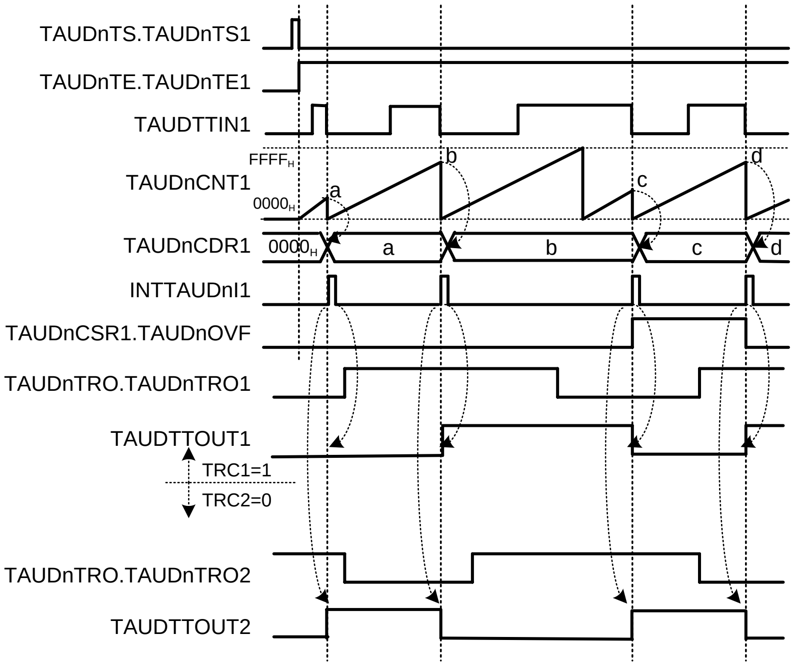


Figure 32.84 General Timing Diagram of Real-Time Output Function Type 2

32.13.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.128 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.129 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Upper Channels

Table 32.130 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

Table 32.131 Simultaneous Rewrite Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.13.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for Lower Channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel Output Mode for Lower Channels

Table 32.132 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

32.13.2.5 Operating Procedure for Real-Time Output Function Type 2

Table 32.133 Operating Procedure for Real-Time Output Function Type 2

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 32.128, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**, and **Table 32.129, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2**. Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Register** Settings for Lower Channels**, Register Settings for Lower Channels**. The TAUDnCDRm register functions as a capture register (only channels with TAUDnTRC.TAUnTRCm = 1). Set channel output mode by setting the control bits as described in **Table 32.130, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 32.132, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
	During Operation	TAUDnTRO.TAUDnTROm can be changed at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm input edge is detected: ● TAUDnCNTm captures the TAUDnCDRm value, and the counter is cleared to 0000_H. ● INTTAUDnIm is generated. ● When the TAUDTTINm input valid edge is detected immediately after the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is set to 1. When detected before the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is cleared to 0. TAUDTTOUTm outputs the current value of real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm, TAUDnCSRm.TAUDnOVF, and TAUDTTOUTm retain their current values.

32.13.2.6 Specific Timing Diagrams

(1) Operation Start and Stop

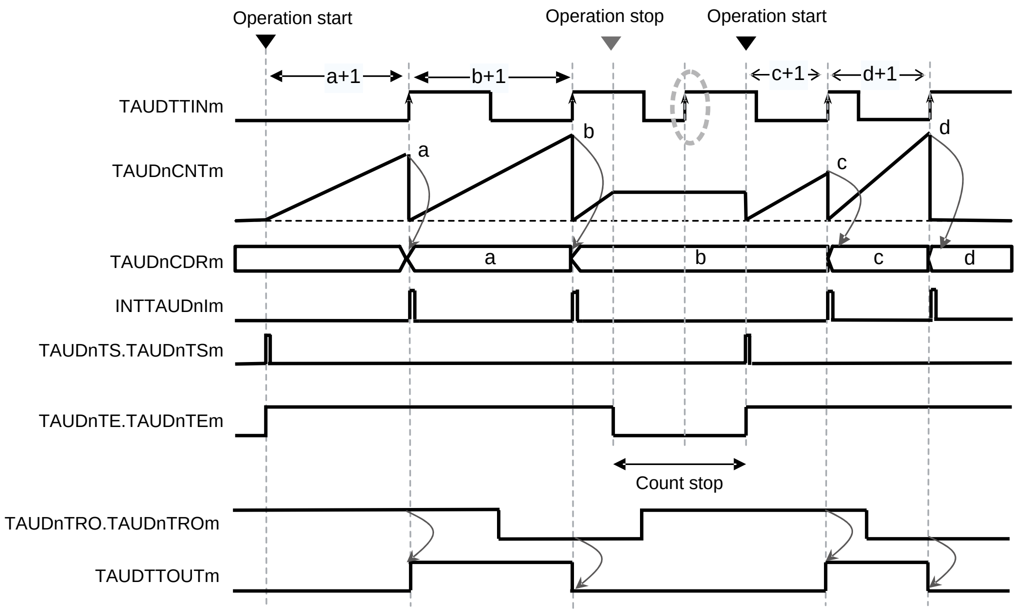


Figure 32.85 Operation Start and Stop (TAUDnCMORm.TAUDnMD0 = 0)

- When TAUDnTS.TAUDnTSm is set to 1, the counter starts counting up.
- When a valid input edge is detected, the current value of the counter is written to the data register (TAUDnCDRm) and an interrupt is generated.
- TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) and the counter resets and starts to count up again.
- The TAUDTTOUTm signal only changes when an interrupt is generated, and then only when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.
- If the counter is stopped (TAUDnTE.TAUDnTEm = 0), valid input edges are ignored and no interrupt is generated.

32.14 Independent Channel Simultaneous Rewrite Functions

This section describes functions that carry out simultaneous rewrite.

32.14.1 Simultaneous Rewrite Trigger Generation Function Type 1

32.14.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 32.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**.)
- For the operating mode that can be set for lower channels, see **Table 32.51, Channel Functions and the Methods They Use for Simultaneous Rewrite**.
- TAUDTTOUTm is not used for any channel in this function.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDnCDRm buf) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000_H, an interrupt occurs on the channel. The current value of the corresponding TAUDnCDRm buffer is loaded into TAUDnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUDnRDC.TAUDnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit should be set to 0 for all other channels in which simultaneous rewrite should take place.

- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.14.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUDnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[Triangle PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUDnCDRm + 1 and value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1 must be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

32.14.1.3 Block Diagram and General Timing Diagram

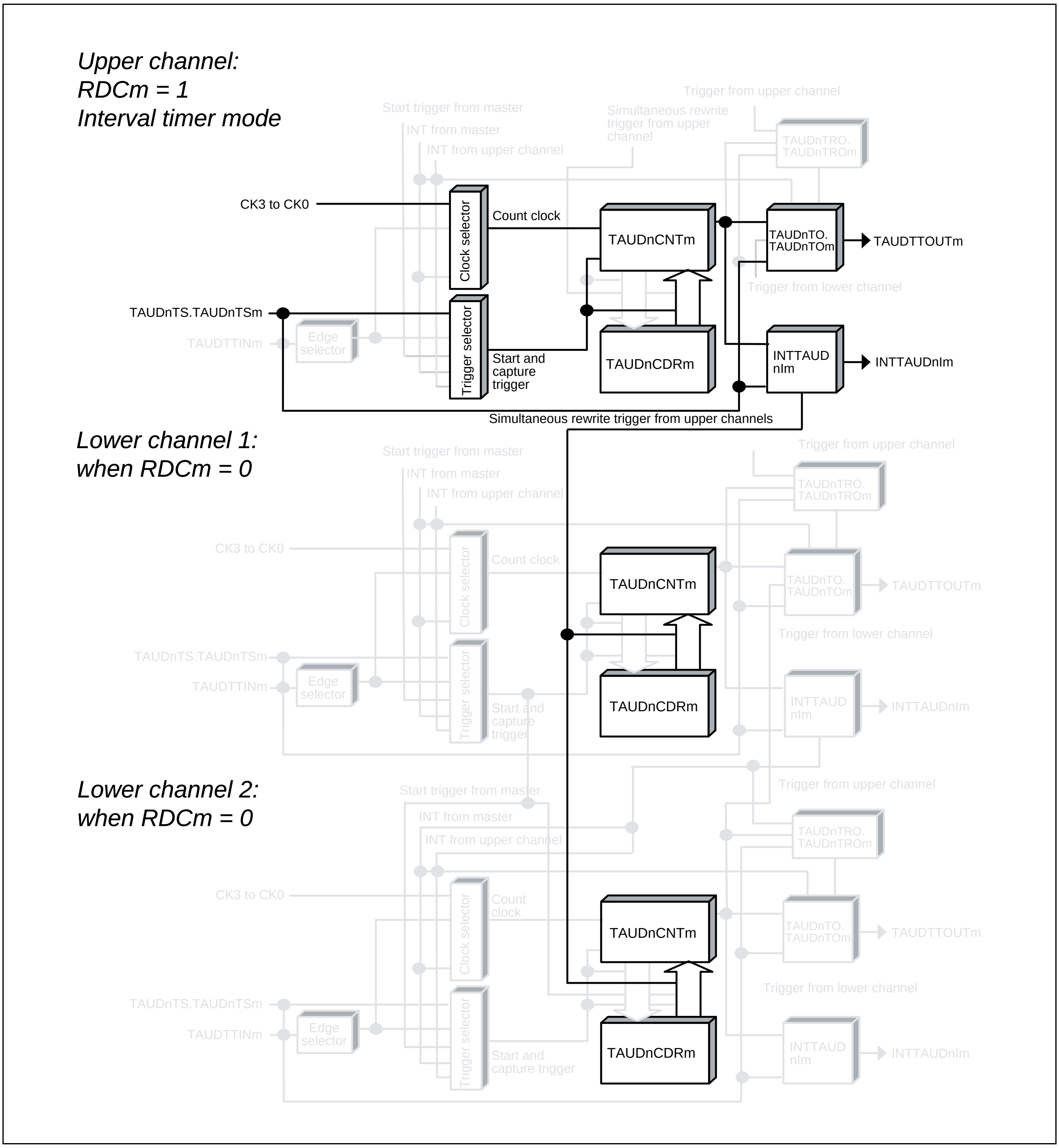


Figure 32.86 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

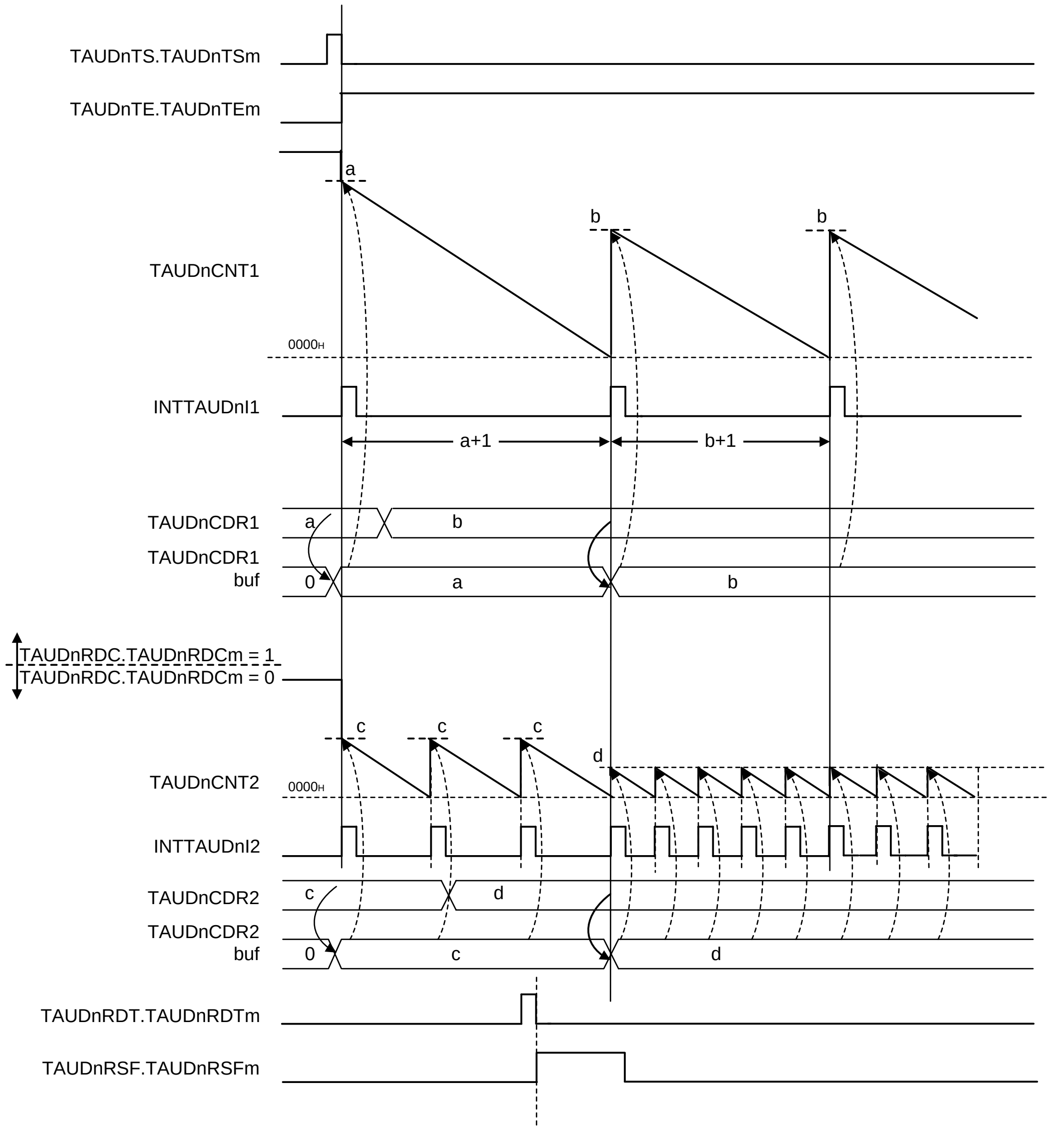


Figure 32.87 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

32.14.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.134 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.135 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Upper Channels

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous Rewrite for Upper Channels

Table 32.136 Simultaneous Rewrite Settings for Upper Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

32.14.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

TAUDnCMORm register for lower channels must follow the TAUDnCMORm register settings in the operating mode which can be set. (See **Table 32.51, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(2) TAUDnCMURm for Lower Channels

TAUDnCMURm register for lower channels must follow the TAUDnCMURm register settings in the operating mode which can be set. (See **Table 32.51, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(3) Channel Output Mode for Lower Channels

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 32.50, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous Rewrite for Lower Channels

Table 32.137 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 32.138 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Table 32.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**, and **Table 32.135, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 32.14.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated.
During Operation	TAUDnRDT.TAUDnRDTm and TAUDnCDRm.TAUDnCDR are changeable. TAUDnRSF.TAUDnRSFm can be always read.	TAUDnCNTm counts down. When the counter reaches 0000_H: ● The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. ● INTTAUDnIm is generated. If INTTAUDnIm is generated on the channel where TAUDnRDC.TAUDnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

32.14.2 Simultaneous Rewrite Trigger Generation Function Type 2

32.14.2.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operation mode of the upper channel must be set to Capture Mode (see **Table 32.139, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**).
- For the operation mode that can be set for a lower channel, see **Table 32.51, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter for the upper channel starts to count up, and then the counter for lower channels start to count according to the selected operating mode.

When a TAUDTTINm input edge occurs on the upper channel, an interrupt is generated. The trigger is detected by the lower channel(s), which then also generate an interrupt.

When TAUDnRDC.TAUDnRDCm = 1 on the upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, an interrupt is generated when the function starts. For details see **Section 32.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

32.14.2.2 Block Diagram and General Timing Diagram

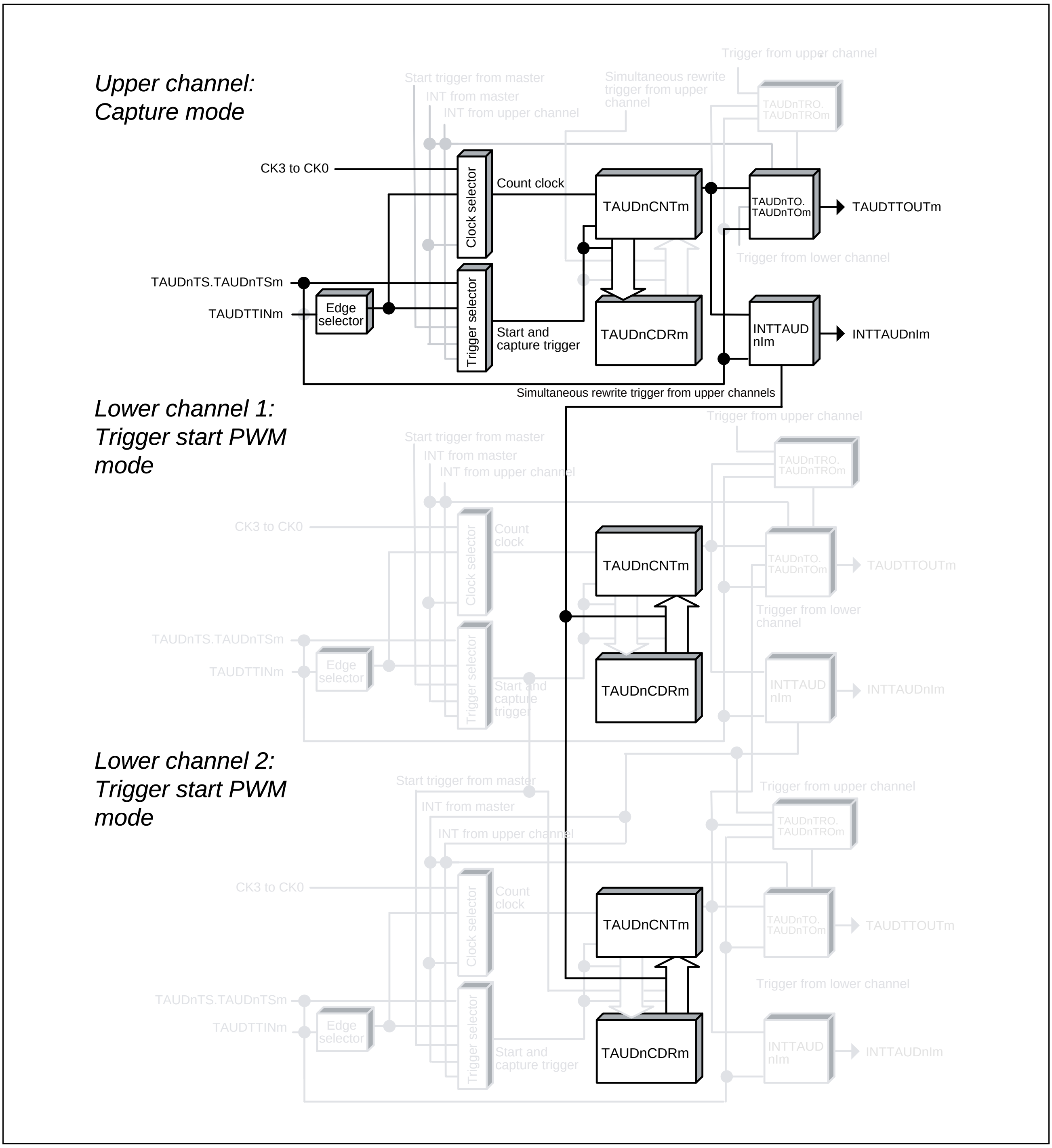


Figure 32.88 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- Upper channel (CH1) generates simultaneous rewrite trigger.

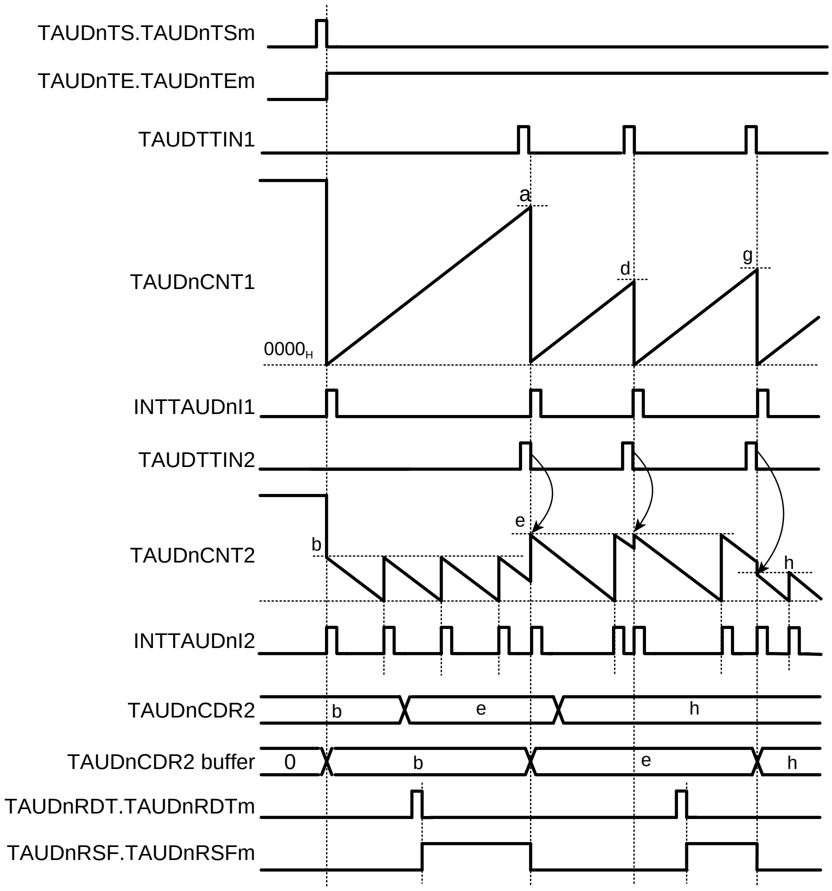


Figure 32.89 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

32.14.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.139 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.140 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Upper Channels

The channel output mode is not used by this function.

(4) Simultaneous Rewrite for Upper Channels

Table 32.141 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

32.14.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.142 Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Lower Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.143 Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Lower Channels

Output can be made according to the trigger start PWM mode setting.

(4) Simultaneous Rewrite for Lower Channels

Table 32.144 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

Table 32.145 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Set the TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Table 32.139, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 32.140, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. Set the TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Table 32.142, Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 32.143, Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000$_H$. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnRDT.TAUDnRDTm can be set at any time. TAUDnRSF.TAUDnRSFm can be read at any time.	TAUDnCNTm counts up from 0000$_H$. When a TAUDTTINm valid edge is detected: ● TAUDnCNTm transfers (captures) its value to TAUDnCDRm and returns to 0000$_H$. ● INTTAUDnIm is generated. Simultaneous rewrite is controlled when INTTAUDnIm is generated from the channel where TAUDnRDC.TAUDnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and it retains its current value.

Restart Operation

32.15 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit D. For a general overview of synchronous channel operation, see **Section 32.2, Overview**.

This section describes functions that generate PWM signals at regular intervals.

32.15.1 PWM Output Function

32.15.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUDTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 32.146, Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operating mode for the slave channels should be set to one-count mode. (See **Table 32.149, Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1. (See **Section 32.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm, and the counter starts counting down from the TAUDnCDRm value. If an INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:
  When the master channel counter reaches 0000_H and the pulse cycle time has passed, INTTAUDnIm is generated. The counter loads TAUDnCDRm value into TAUDnCNTm and counts down.
- Slave channel:
  When INTTAUDnIm is generated on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts counting down from the TAUDnCDRm value. TAUDTTOUTm signal is set to the active level.
  When the counter reaches to 0000_H (duty time has elapsed), INTTAUDnIm is generated and a TAUDTTOUTm signal is set to an inactive level. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm (start of the next pulse cycle) of the master channel.

The counter can stop operating by setting the TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Set Simultaneous rewrite with this function. See **Section 32.6, Simultaneous Rewrite**.

32.15.1.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)) × 100

- Duty cycle = 0%
  TAUDnCDRm (slave) = 0000$_H$
- Duty cycle = 100%
  TAUDnCDRm (slave) $\geq$ TAUDnCDRm (master) + 1

32.15.1.3 Block Diagram and General Timing Diagram

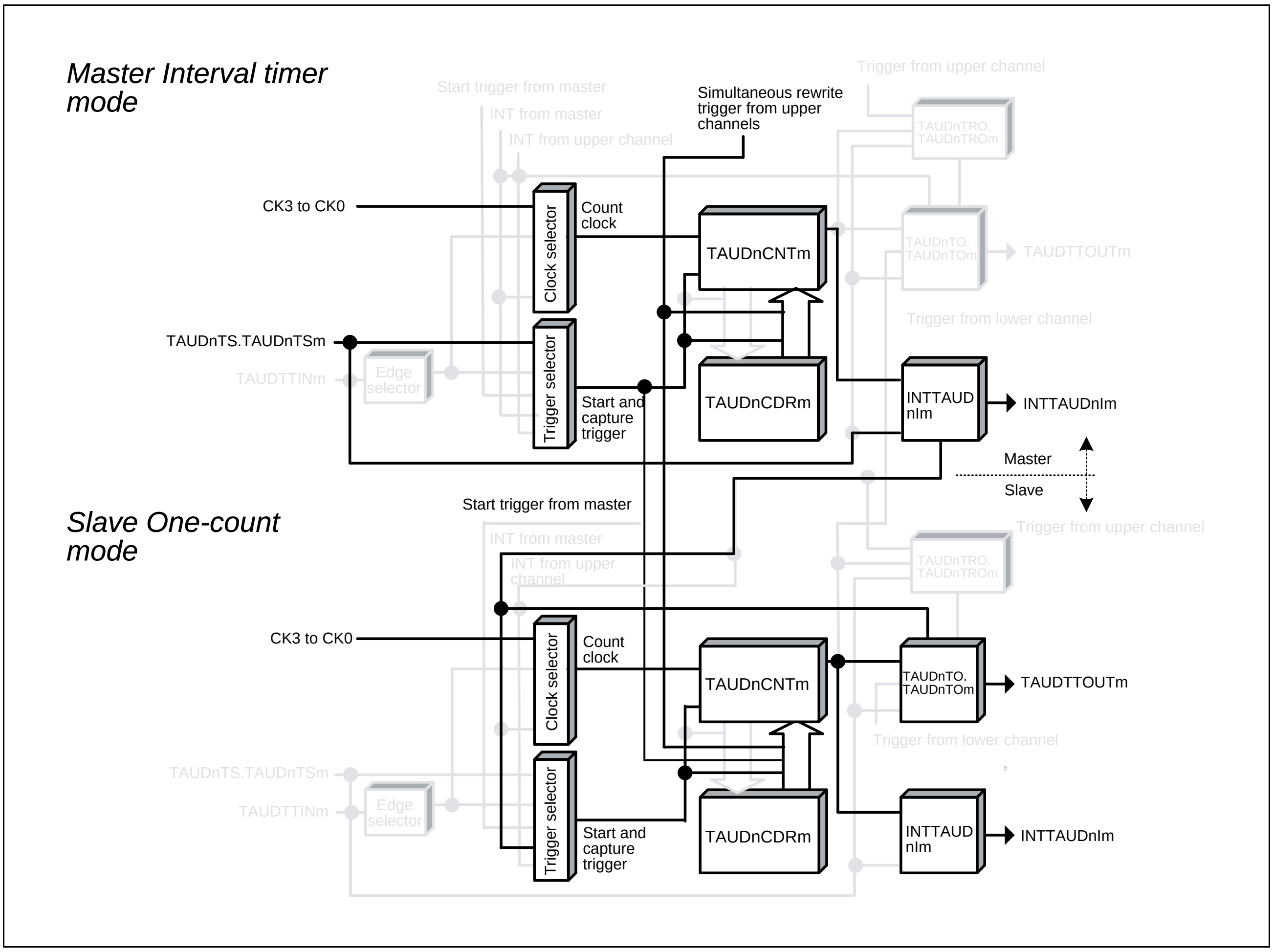


Figure 32.90 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

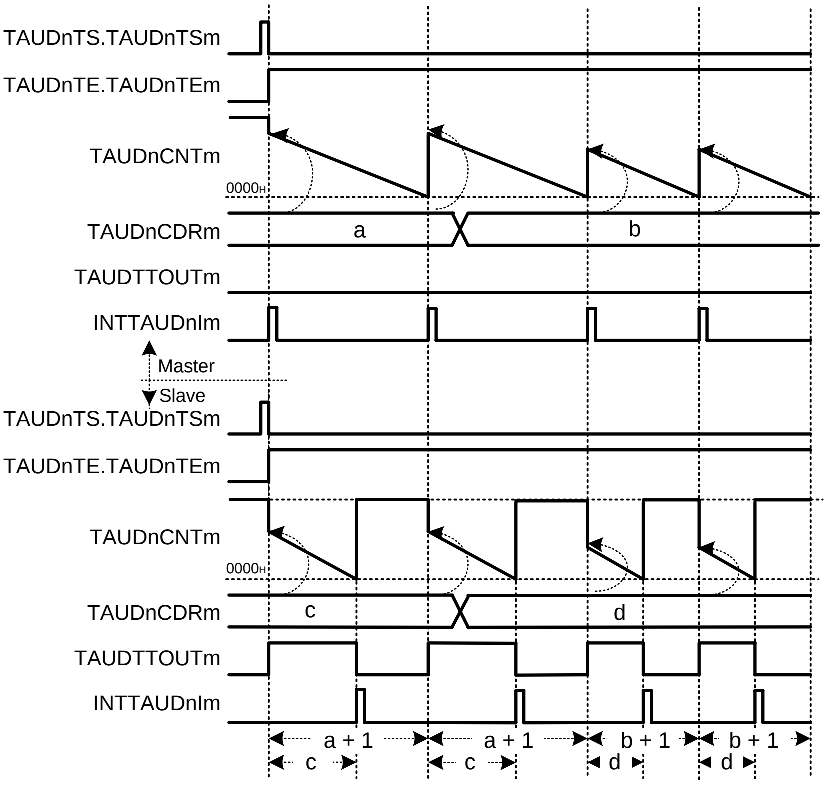


Figure 32.91 General Timing Diagram of PWM Output Function

NOTES

1. The interval between the start of the count and an interrupt being generated is the value of corresponding TAUDnCDRm + 1.
2. TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

32.15.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.146 Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.147 Contents of the TAUDnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

The channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.148 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires a channel higher than the master channel that operates with **Section 32.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  TAUDnCDRm settings for this channel are as follows:
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × interrupt count) – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

If TAUDnCDRm (slave) setting > TAUDnCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

32.15.1.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.149 Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.150 Contents of the TAUDnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 32.151 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.152 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.1.6 Operating Procedure for PWM Output Function

Table 32.153 Operating Procedure for PWM Output Function

Restart Operation

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.1.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.1.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.
During operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: ● INTTAUDnIm (master) is generated. ● TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. ● TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. ● TAUDTTOUTm (slave) is set to the active level. If TAUDnCNTm (slave) reaches 0000_H: ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.15.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

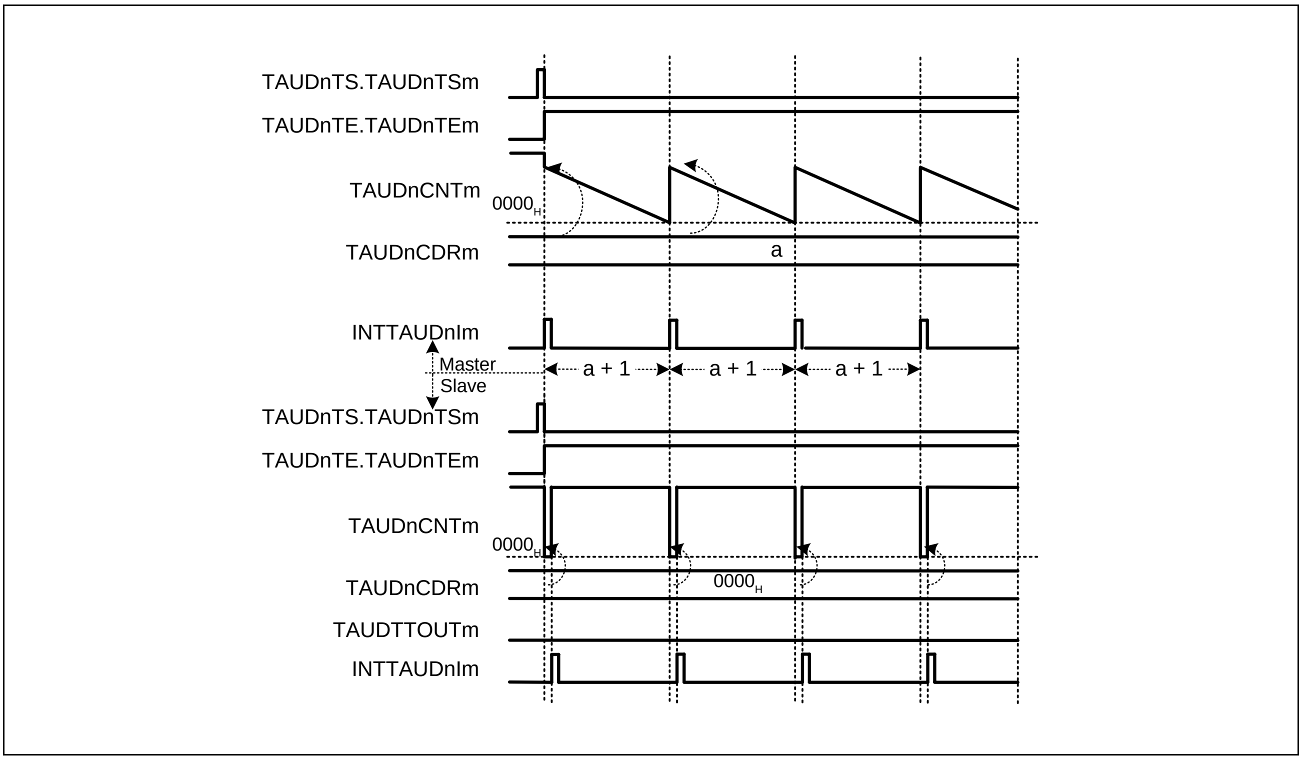


Figure 32.92 TAUDnCDRm (Slave) = 0000_H,
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is loaded into TAUDnCNTm (slave). As a result, a slave channel interrupt (INTTAUDnIm) is generated at the same time and TAUDTTOUTm remains inactive.
- TAUDnCDRm value is loaded into TAUDnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

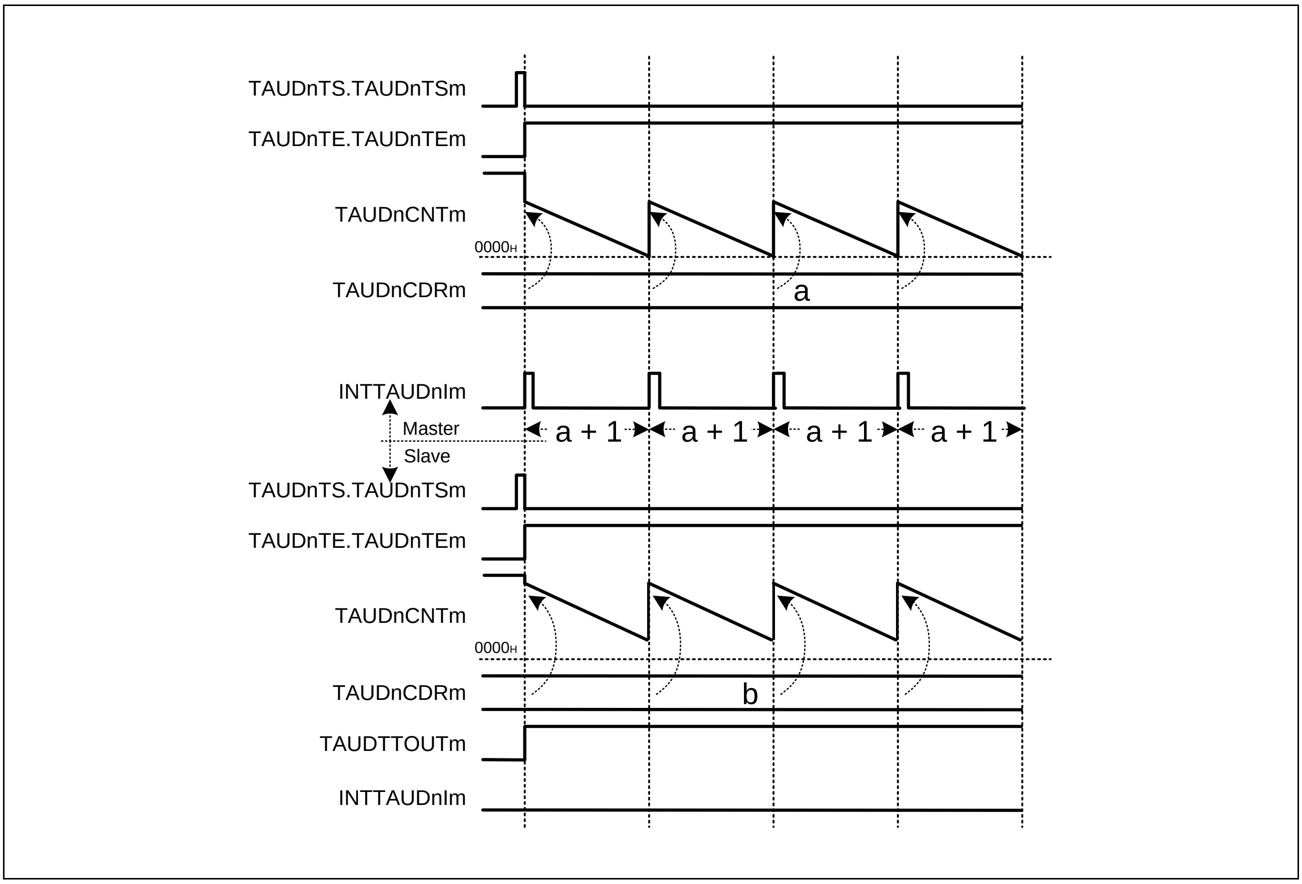


Figure 32.93 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the slave channel counter does not reach 0000_H and consequently, no interrupt occurs. TAUDTTOUTm remains active.

(3) Operation Stop and Restart

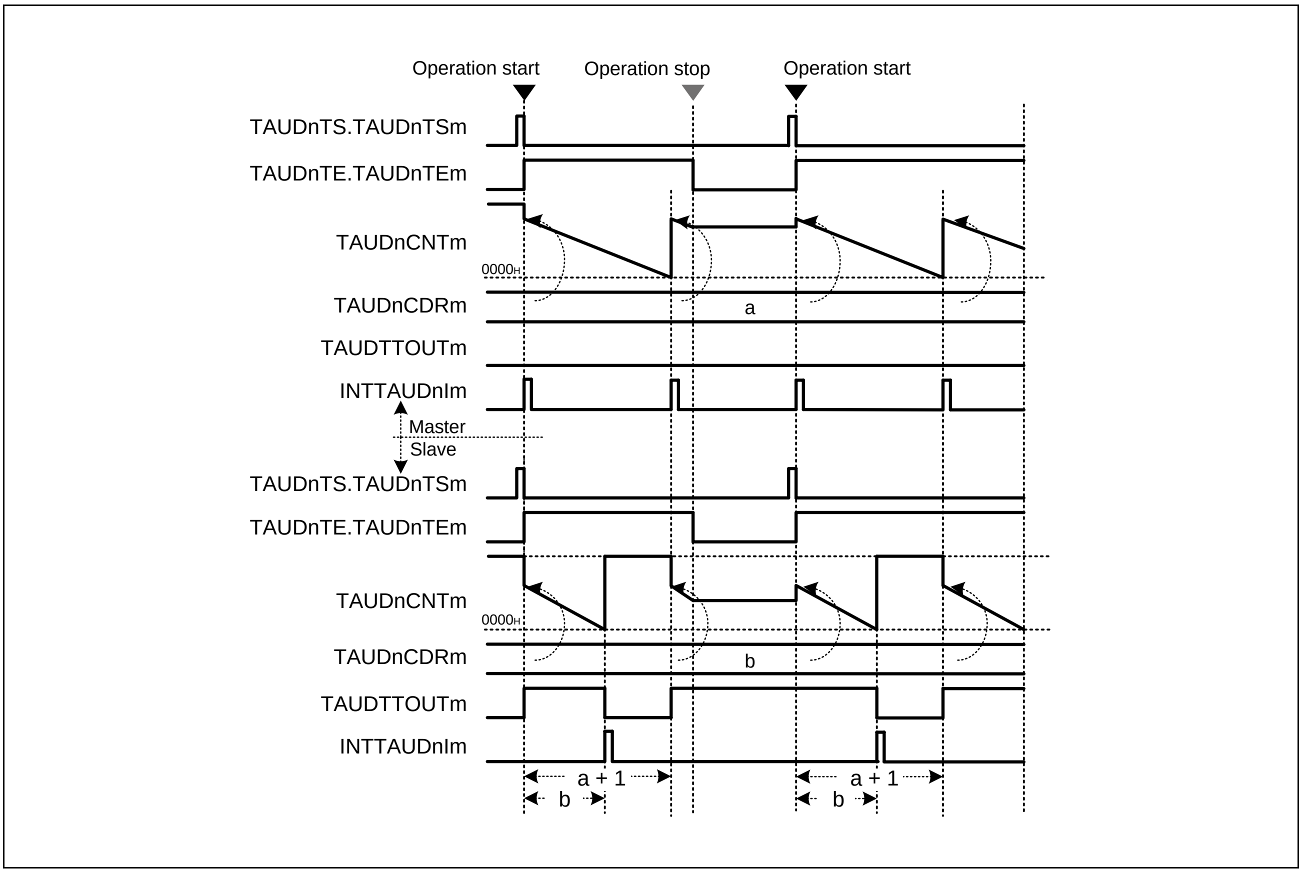


Figure 32.94 Operation Stop and Restart
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm of master and slave channels to 1. TAUDnCDRm values of the master and slave channels are loaded to TAUDnCNTm and start to count down from these values.

32.15.2 One-Shot Pulse Output Function

32.15.2.1 Overview

Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See **Table 32.154, Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**.)
- The operating mode for slave channels should be set to pulse one-count mode. (See **Table 32.157, Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**.)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channel should be set to independent channel output mode 2. (See **Section 32.7, Channel Output Modes**.)
- TAUDTTINm (master) has to be detected while TAUDnCNTm (master) and TAUDnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDTTINm (slave).

Functional description

The counters are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1 for the master and slave channels. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  When the next valid TAUDTTINm input edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm. The counter starts to count down from this value. If TAUDnCMORm.TAUDnMD0 = 0, a trigger (TAUDTTINm) which is detected within the delay time is ignored.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next valid TAUDTTINm input edge.
- Slave channel:
  INTTAUDnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter stops at 0000_H and waits for the next INTTAUDnIm of master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Setting TAUDnTS.TAUDnTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

Conditions

- If TAUDnCMORn.TAUDnMD0 of master channel is set to 0, TAUDTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See **Section 32.6, Simultaneous Rewrite**.

32.15.2.2 Equations

Delay from trigger input to pulse output

= (TAUDnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

32.15.2.3 Block Diagram and General Timing Diagram

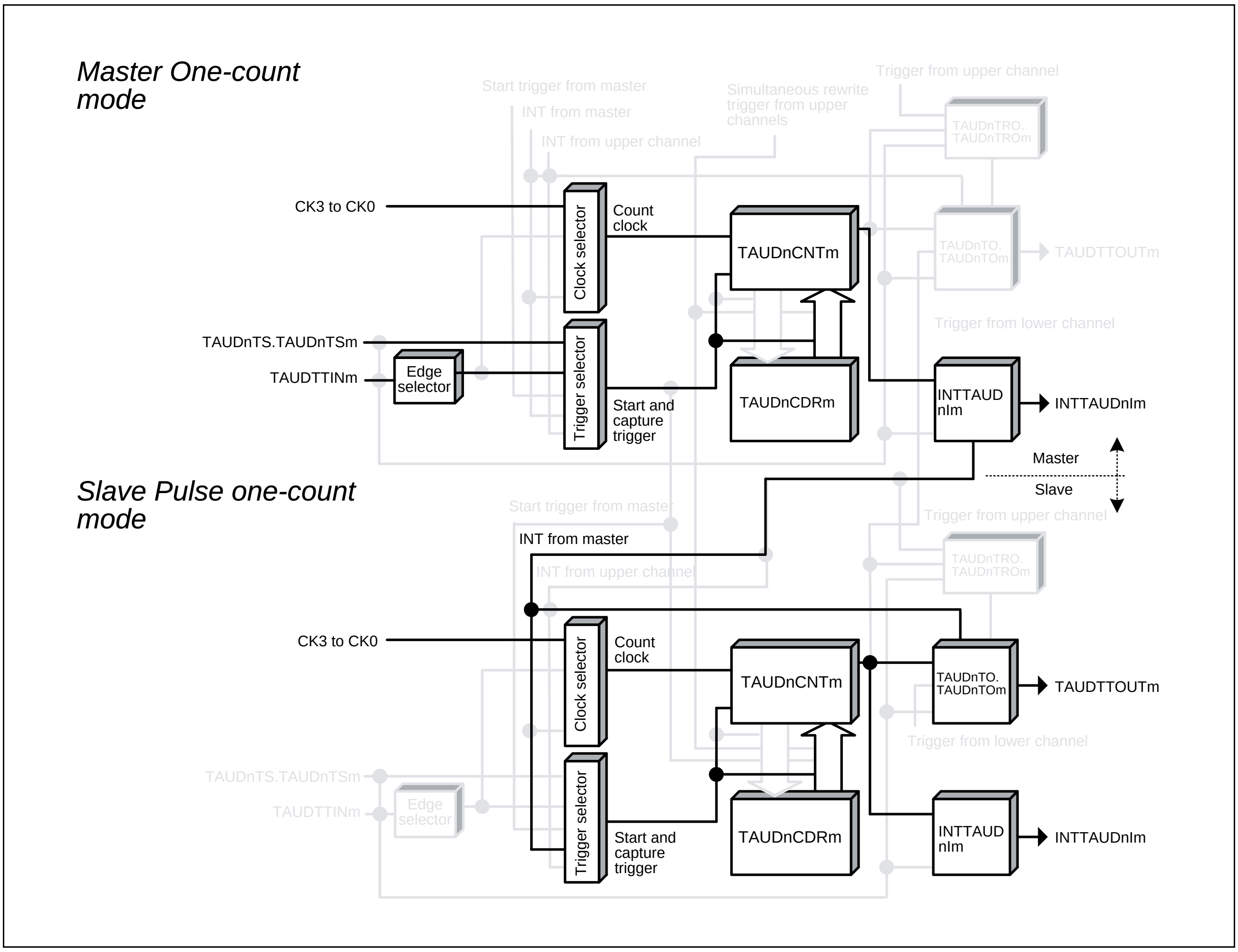


Figure 32.95 Block Diagram of One-Shot Pulse Output Function

The following settings apply to the general timing diagram.

- Start trigger detection is disabled during counting (TAUDnCMORm.TAUDnMD0 = 0).
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

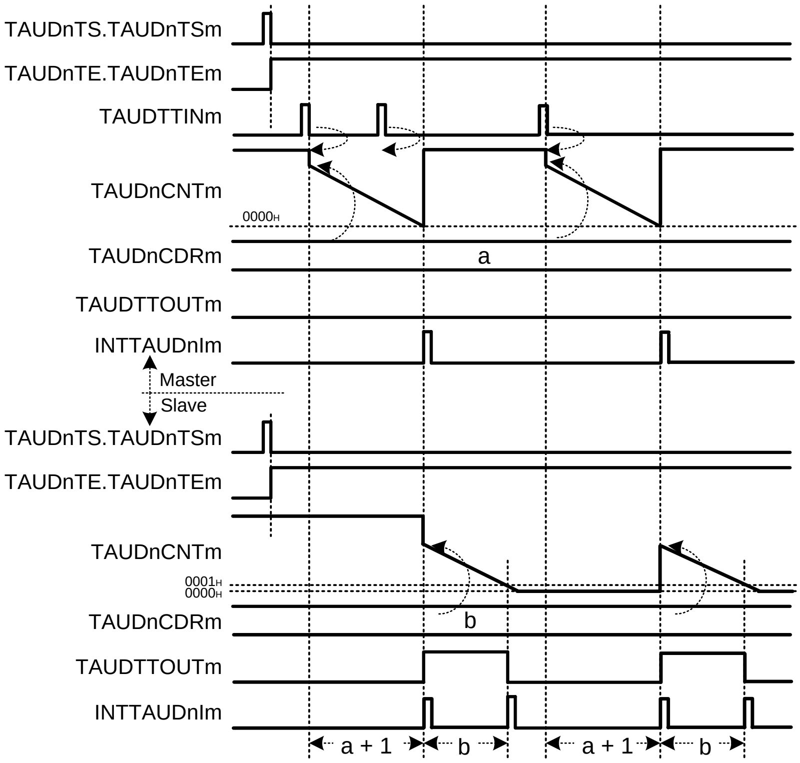


Figure 32.96 General Timing Diagram of One-Shot Pulse Output Function

32.15.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.154 Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. The value of the MD0 bits of the master and slave channels must be identical.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.155 Contents of the TAUDnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.156 Simultaneous Rewrite Settings for the Master Channel of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.2.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.157 Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables start trigger detection while counting. The MD0 bits of the master and slave channels must be identical.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.158 Contents of the TAUDnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Slave Channel

Table 32.159 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.160 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 32.161 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.2.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 32.15.2.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUDTTINm input.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	When valid TAUDTTINm input edge is detected, TAUDnCDRm value of master channel is loaded into TAUDnCNTm to perform counting down. When the counter reaches 0000$_H$: ● INTTAUDnIm (master) is generated. ● TAUDnCNTm (master) is reset to FFFF$_H$ and waits for the next valid TAUDTTINm input edge. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is set to the active level. When TAUDnCNTm (slave) reaches 0001$_H$: ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

32.15.2.7 Specific Timing Diagrams

(1) TAUDnCDRm (master) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

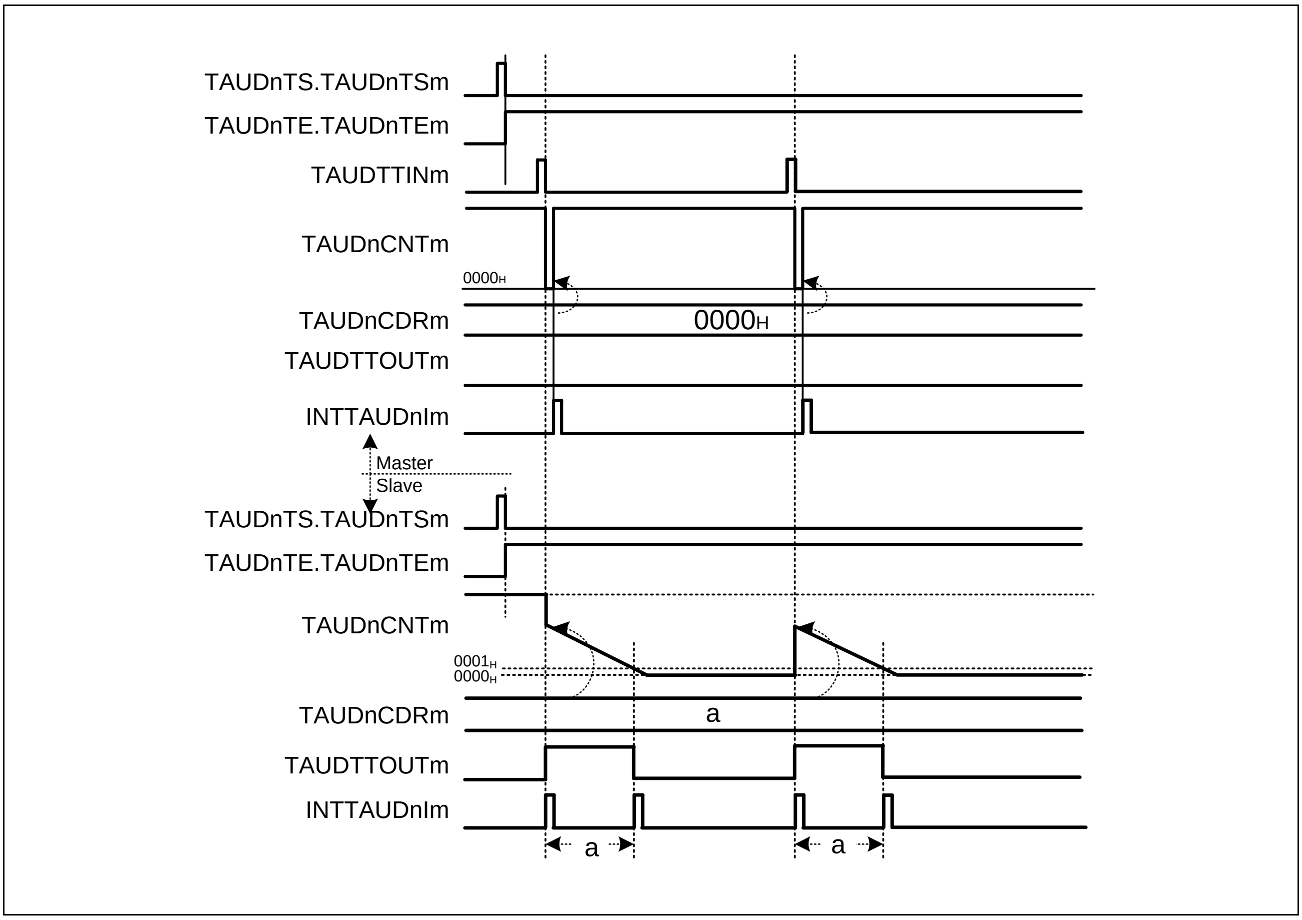


Figure 32.97 TAUDnCDRm (Master) = 0000_H

- When a valid TAUDTTINm input edge is detected, the value 0000_H is written to TAUDnCNTm (master). The counter is set to 0000_H for one count and returns to $FFFF_H$.
  Thus the slave channel starts to count down one count clock later than TAUDTTINm (master).

(2) TAUDnCDRm (slave) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

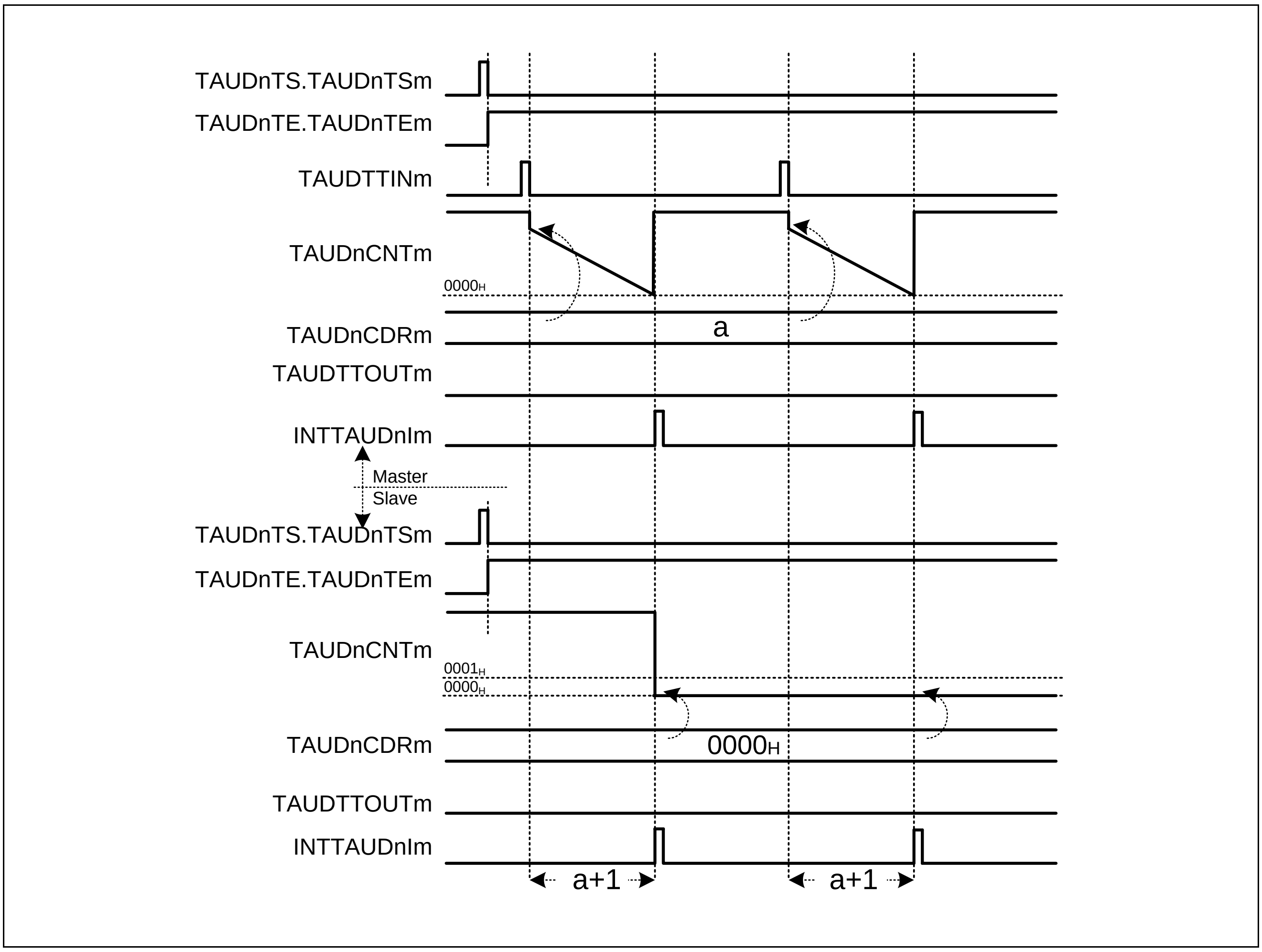


Figure 32.98 TAUDnCDRm (Slave) = 0000_H

- TAUDTTOUTm remains inactive, because the pulse width is zero.

(3) TAUDnCMORm.TAUDnMD0 = 1

The following settings apply to this diagram.

- Enables start trigger detection while counting. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

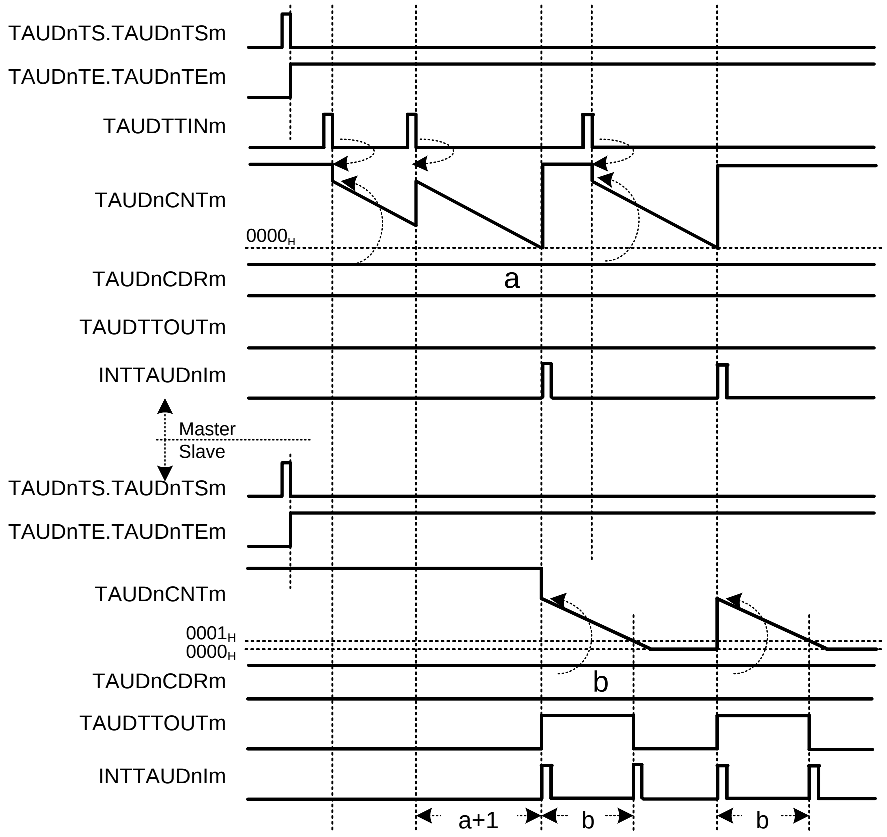


Figure 32.99 TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDTTINm input edge is detected while the counter of the master channel counts down, TAUDnCNTm reloads the value of TAUDnCDRm. The counter restarts to count down.
  This means the delay is extended by the value of TAUDnCNTm at the time when a valid TAUDTTINm input edge is detected.

(4) Restarting the Master Channel while the Slave Channel is Counting

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

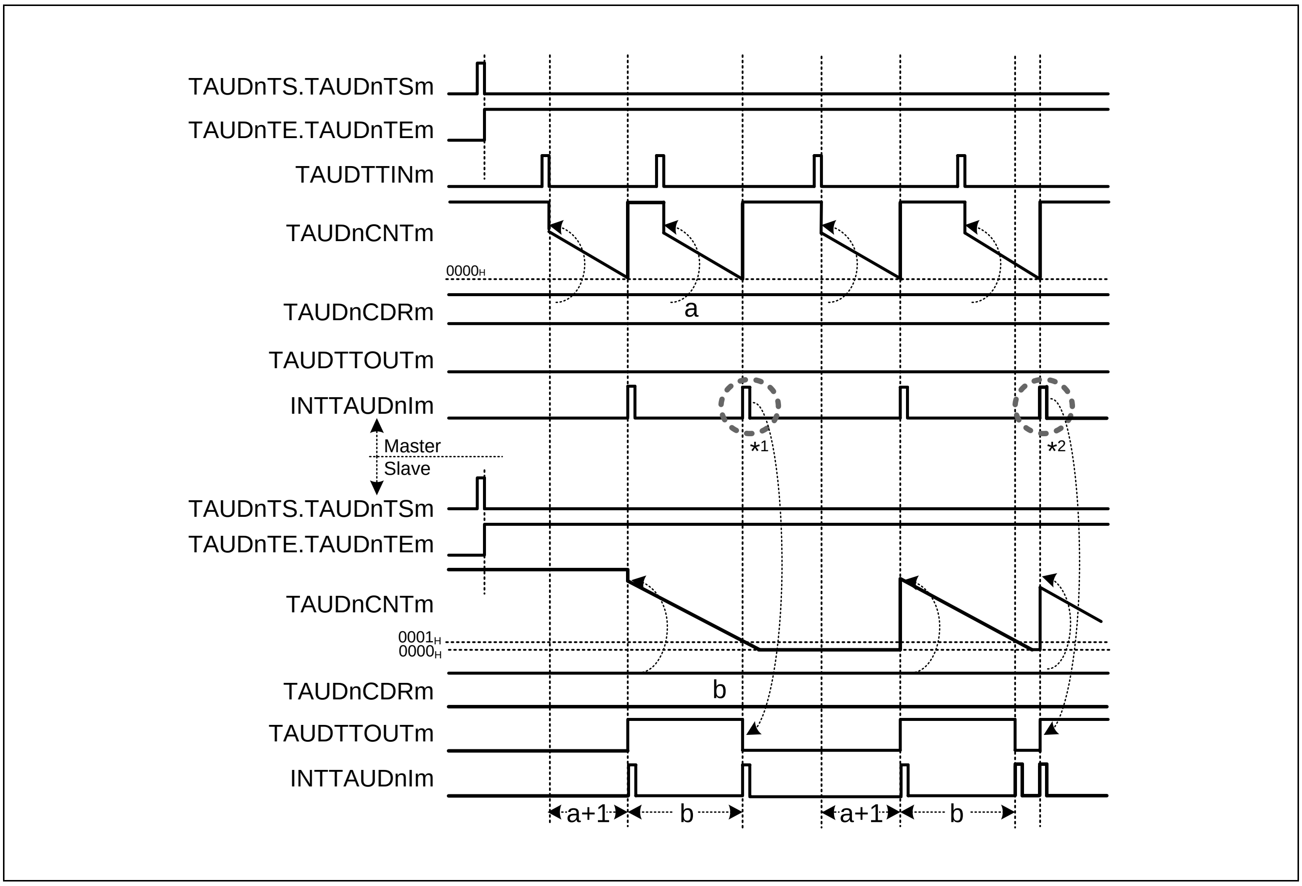


Figure 32.100 TAUDTTINm input interval ≤ Delay Time + Pulse Width +1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached*[1], the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUDnCDRm (slave) is reloaded. An interrupt is generated and TAUDTTOUTm toggles. If TAUDnCNTm (master) has started to count down while the TAUDnCNTm (slave) is still counting*[2], TAUDTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

32.15.3 Trigger Start PWM Output Function

32.15.3.1 Overview

Summary

This function generates a PWM output using a master and a slave channel. It enables the pulse cycle (frequency) and the duty of the TAUDTTOUTm to be set. The pulse cycle is specified using the master channel. The duty is specified using the slave channel. The Trigger Start PWM Output Function is identical to PWM Output Function except that the master channel of this function can be reset by a valid TAUDTTINm input edge.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode (see **Table 32.162, Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function**).
- The operation mode of the slave channel must be set to One-Count Mode (see **Table 32.165, Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function**).
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1 (see **Section 32.6, Simultaneous Rewrite**).

Functional description

The counters (master and slave) are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of TAUDnCDRm is loaded to TAUDnCNTm, and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel, and a PWM output is realized by setting and resetting TAUDTTOUTm (slave).

- Master channel:
  The current value of TAUDnCDRm is loaded to the counter (TAUDnCNTm), INTTAUDnIm is generated and the counter starts to count down from this value.
  When the counter reaches 0000_H and the pulse cycle time has elapsed, INTTAUDnIm is generated and the counters (master and slave) load the current TAUDnCDRm values.
  If a valid TAUDTTINm input edge is detected, the counter of the master channel loads the current TAUDnCDRm value, restarts counting down and generates an interrupt.
- Slave channel:
  When the slave detects an interrupt from the master channel, it starts to count down from the current value of TAUDnCDRm. The TAUDTTOUTm signal is set to the active level.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 32.6, Simultaneous Rewrite**.

32.15.3.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave) / (TAUDnCDRm (master) + 1)] × 100

- Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
- Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

32.15.3.3 Block Diagram and General Timing Diagram

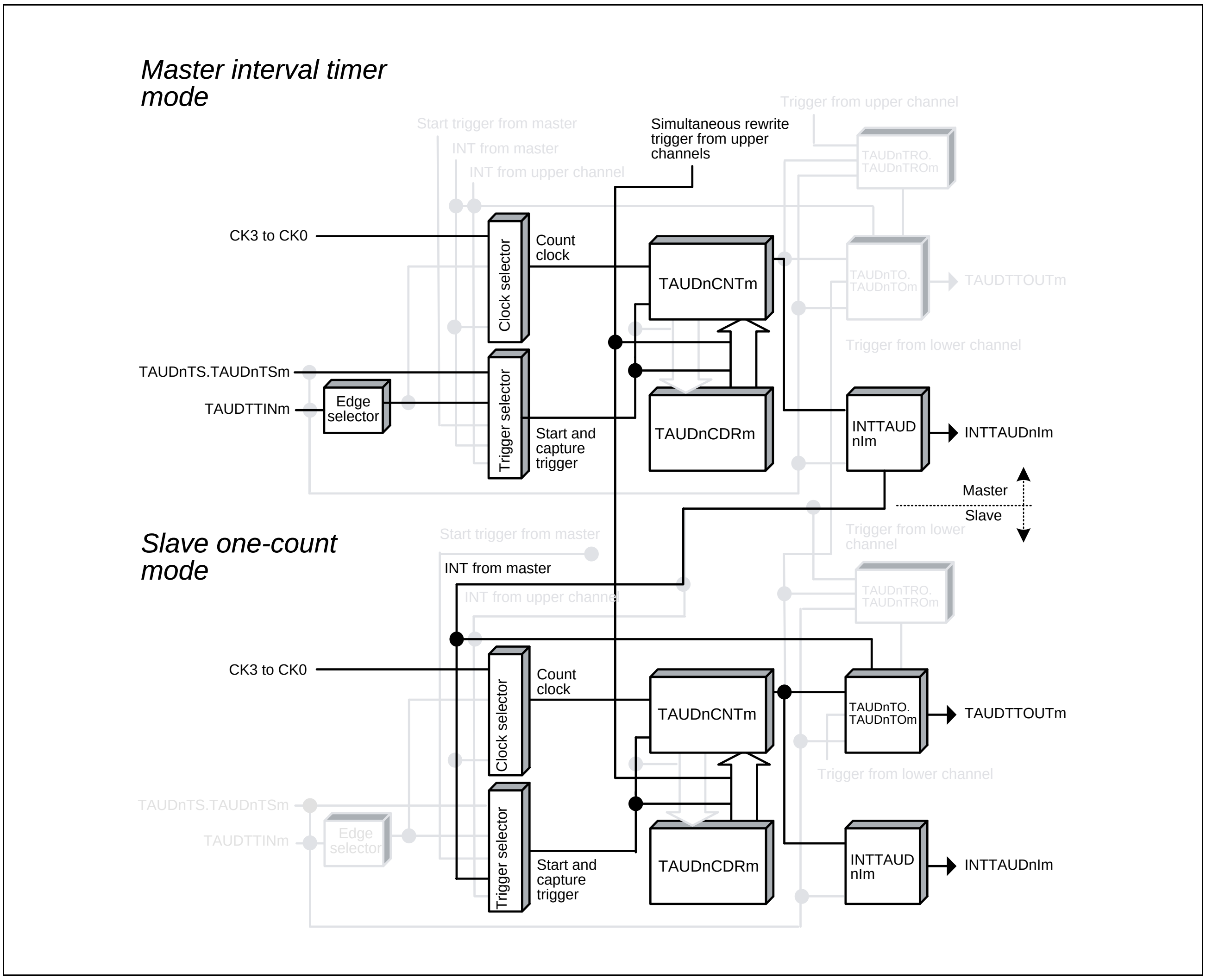


Figure 32.101 Block Diagram for Trigger Start PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)
- Positive logic (TAUDnTOL.TAUDnTOLm (slave) = 0)

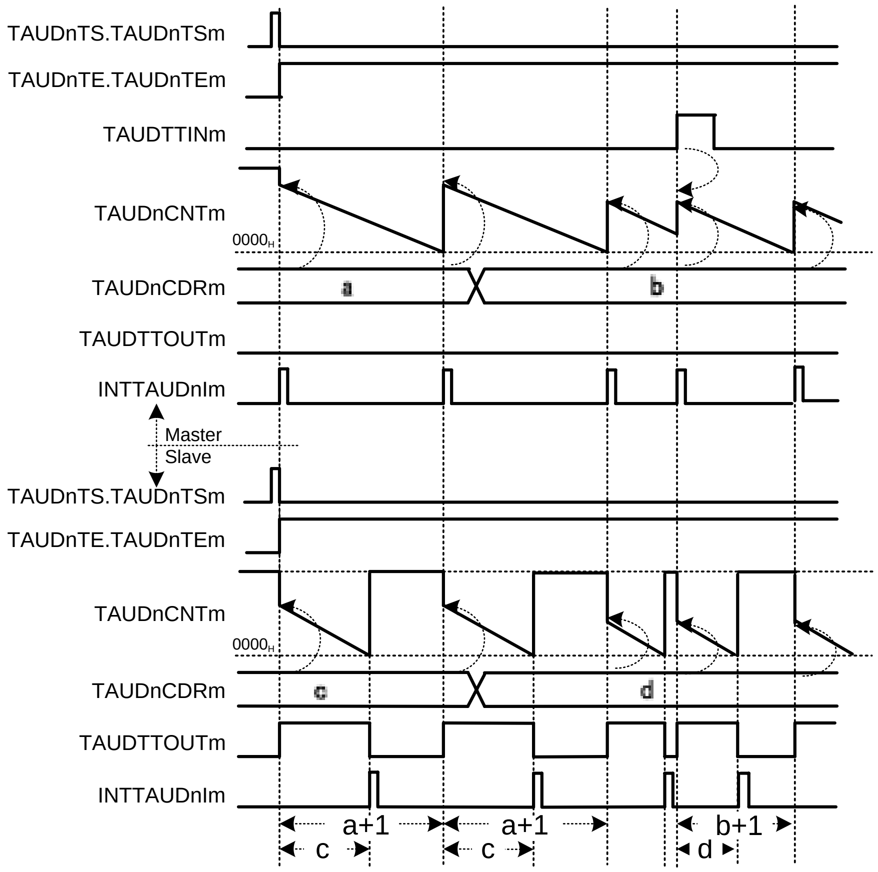


Figure 32.102 General Timing Diagram for Trigger Start PWM Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

32.15.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.162 Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.163 Contents of the TAUDnCMURm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

The channel output mode is not used by this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.164 Simultaneous Rewrite Settings for the Master Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.3.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.165 Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid. The value of the TAUDnMD[0] bit of the master and slave channels must be identical.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.166 Contents of the TAUDnCMURm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Slave Channel

Table 32.167 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.168 Simultaneous Rewrite Settings for the Slave Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.3.6 Operating Procedure for Trigger Start PWM Output Function

Table 32.169 Operating Procedure for Trigger Start PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.3.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.3.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: ● INTTAUDnIm (master) is generated. ● TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. ● TAUDnCNTm (slave) loads the TAUDnCDRm value and starts to count down ● TAUDTTOUTm (slave) is set When TAUDnCNTm of the slave = 0000_H: ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops. If a TAUDTTINm input is detected on the master channel while TAUDnCNTm of the master channel is counting down: ● TAUDnCNTm (master and slave) loads the TAUDnCDRm value and counts down ● INTTAUDnIm (master) is generated. ● TAUDTTOUTm (slave) is set to the active level.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

32.15.3.7 Specific Timing Diagrams

(1) Duty cycle = 0%

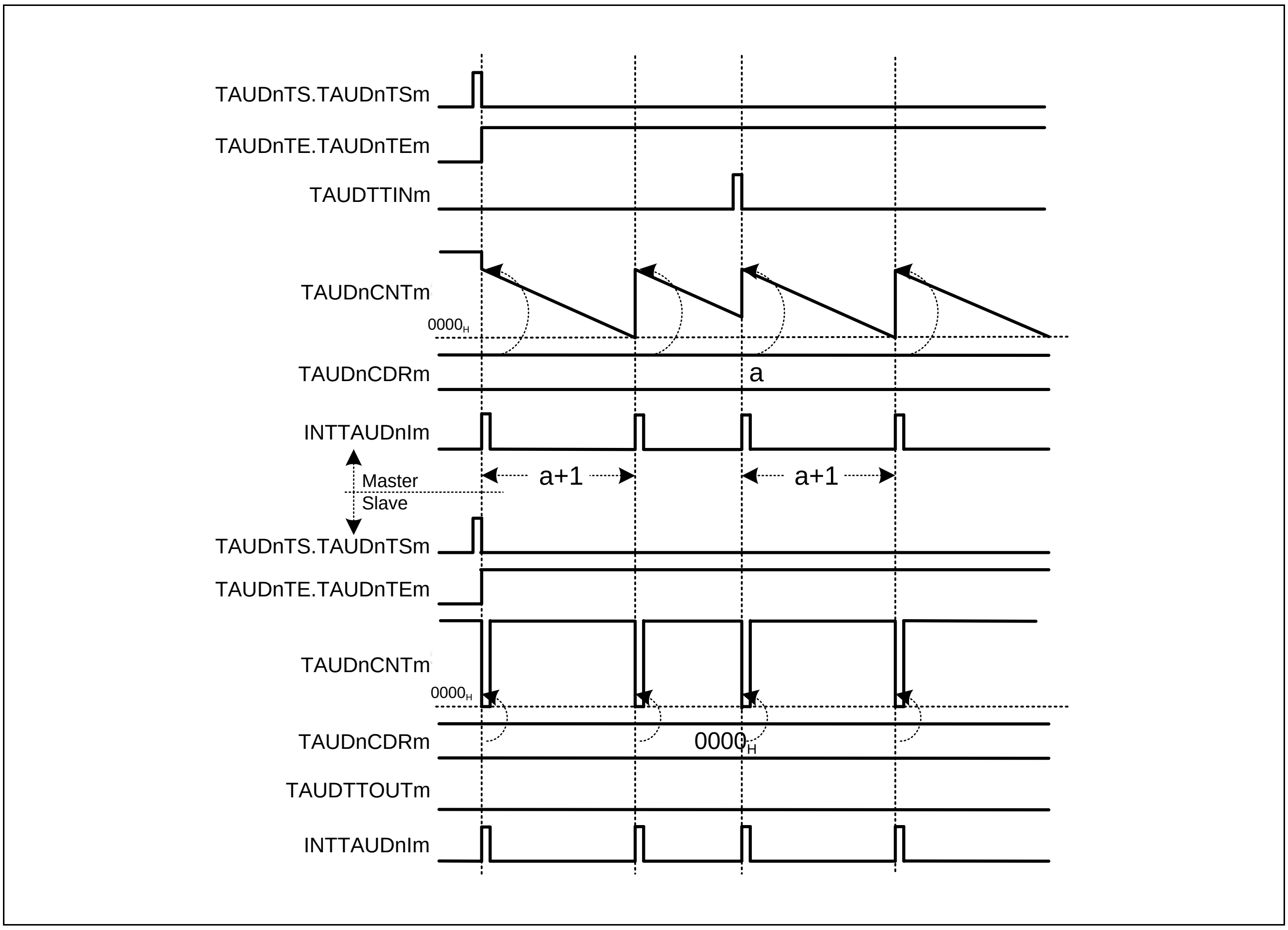


Figure 32.103 TAUDnCDRm (Slave) = 0000$_H$,
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00$_B$)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000$_H$ is written to TAUDnCNTm (slave). Therefore, TAUDnCNTm (slave) cannot start to count and TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(2) Duty cycle = 100%

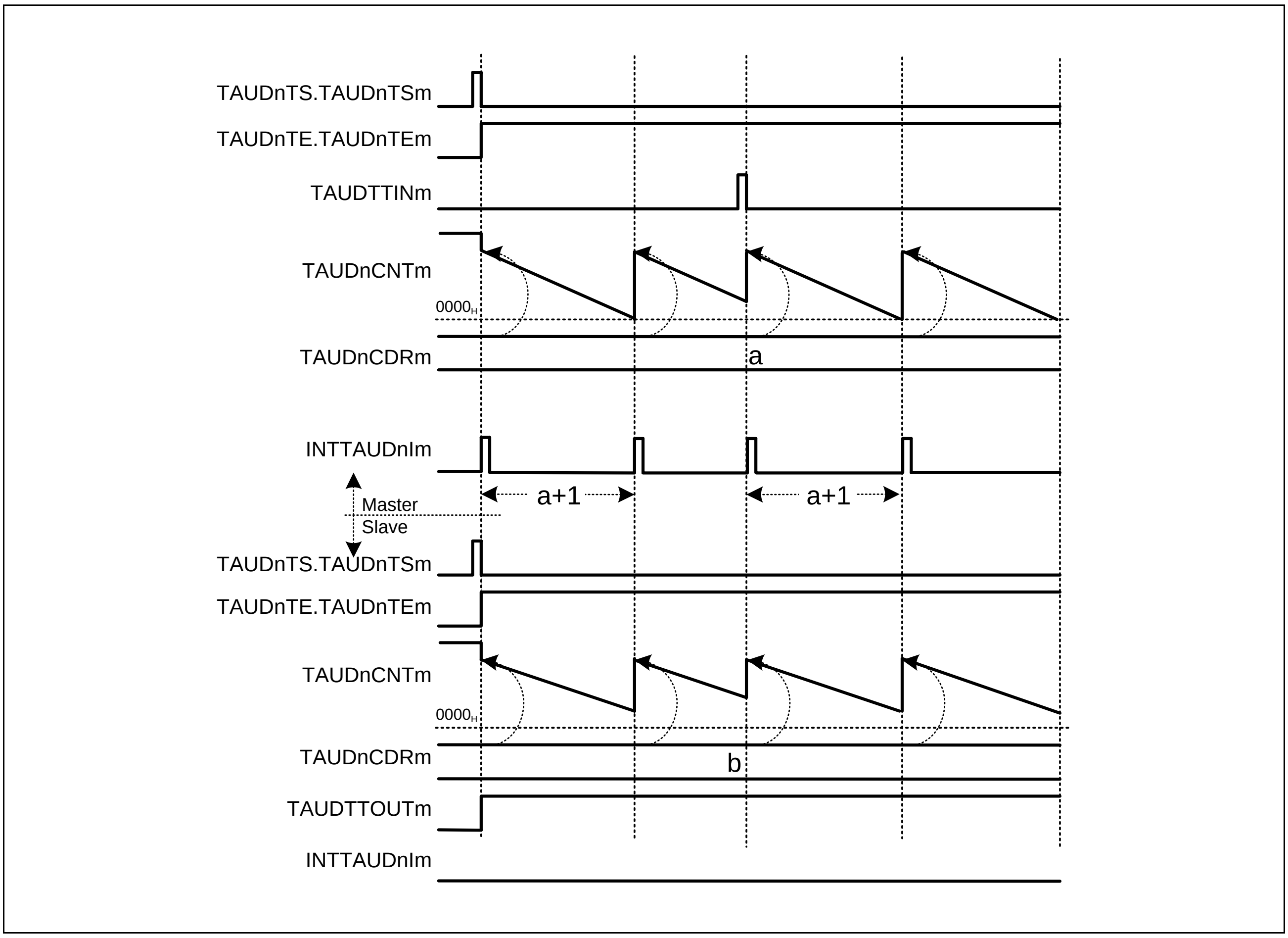


Figure 32.104 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1,
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Falling Edge Detection (TAUDnCMURm.TIS[1:0] = 00_B)

- If the value TAUDnCDRm (slave) is higher than the value TAUDnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts.
  The TAUDTTOUTm remains at active state.
  The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(3) TAUDTTINm Detection and Active Slave Counter

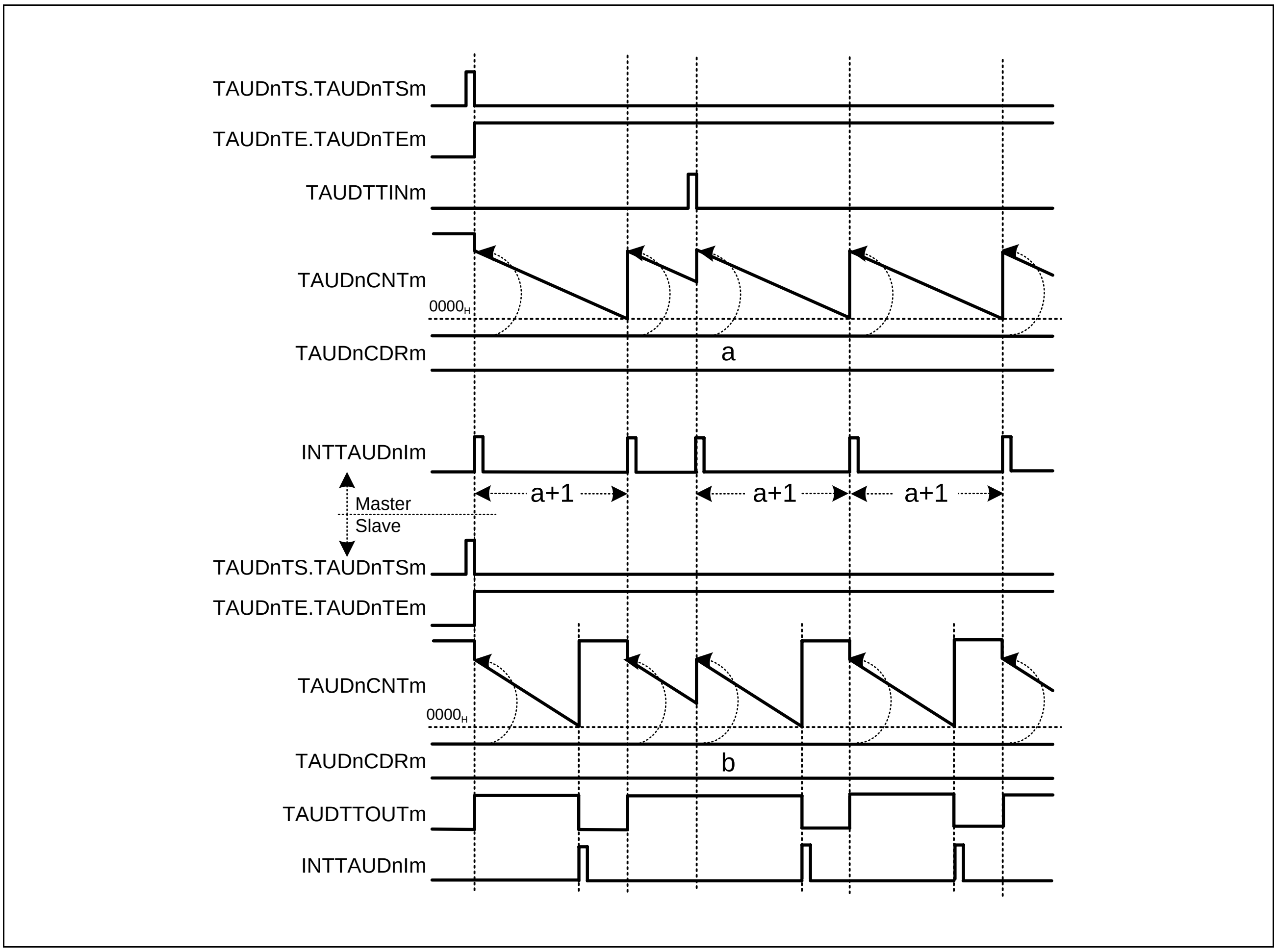


Figure 32.105 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- If TAUDnCNTm (slave) reloads the value TAUDnCDRm (slave) while it is still counting down, TAUDTTOUTm remains unchanged and extends the duty.
  The duty does not correspond to the value of the slave’s data register.

32.15.4 Delay Pulse Output Function

32.15.4.1 Overview

Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified on slave channel 2.

Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 32.170, Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function**.)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See **Table 32.173, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function** and **Table 32.177, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function**.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 32.180, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function**.)
- TAUDTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See **Section 32.7, Channel Output Modes**.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 32.7, Channel Output Modes**.)

Functional description

The counters of the channel group are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel.
  When the counter value of master channel reaches 0000_H and pulse cycle time has elapsed, INTTAUDnIm is generated. The TAUDnCDRm value is reloaded into the counter to perform counting down.
- Slave channels 1 and 2:
  Slave channels 1 and 2 start to count down from the current TAUDnCDRm value when detecting an interrupt from the master channel. TAUDTTOUTm signal (slave 1) is set.
- Slave channel 1:
  When the counter of slave channel 1 reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and

TAUDTTOUTm signal is reset. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.

  - Slave channel 2:
    When the counter of slave channel 2 reaches 0000_H and delay time has elapsed, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.
    Generating INTTAUDnIm (slave channel 2) triggers the counter of slave channel 3.

- Slave channel 3:
  When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUDnCDRm. INTTAUDnIm is generated and the TAUDTTOUTm signal (slave channel 3) is set.
  When the counter of slave channel 3 reaches 0001_H, INTTAUDnIm is generated and the TAUDTTOUTm signal is reset.
  The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 32.6, Simultaneous Rewrite**.

32.15.4.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDnCDRm (slave 1)) × count clock cycle

Delay width = (TAUDnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:
$0000_H \leq$ TAUDnCDRm (slave 2) < TAUDnCDRm (master)

NOTES

1. The waveform of TAUDTTOUTm (slave 3) becomes the waveform made by delaying the waveform of TAUDTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.
2. If INTTAUD0Im of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the waveform of TAUDTTOUTm (slave 3) is retained on the active level. In this case, TAUDTTOUTm (slave 3) cannot output the waveform generated by delaying the basic pulse of TAUDTTOUTm (slave 1).

32.15.4.3 Block Diagram and General Timing Diagram

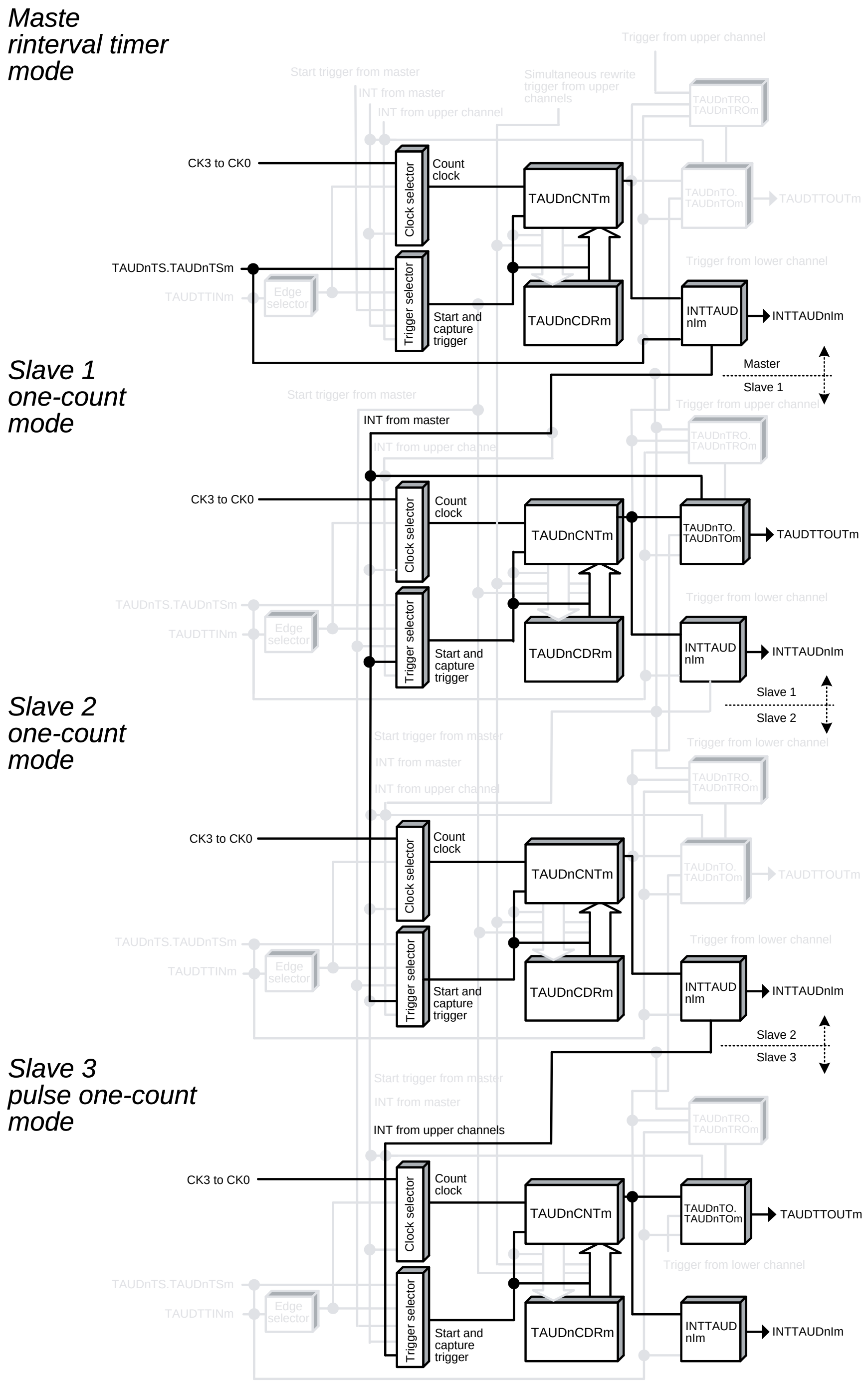


Figure 32.106 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDnTOL.TAUDnTOLm = 0)
- Slave channel 3: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

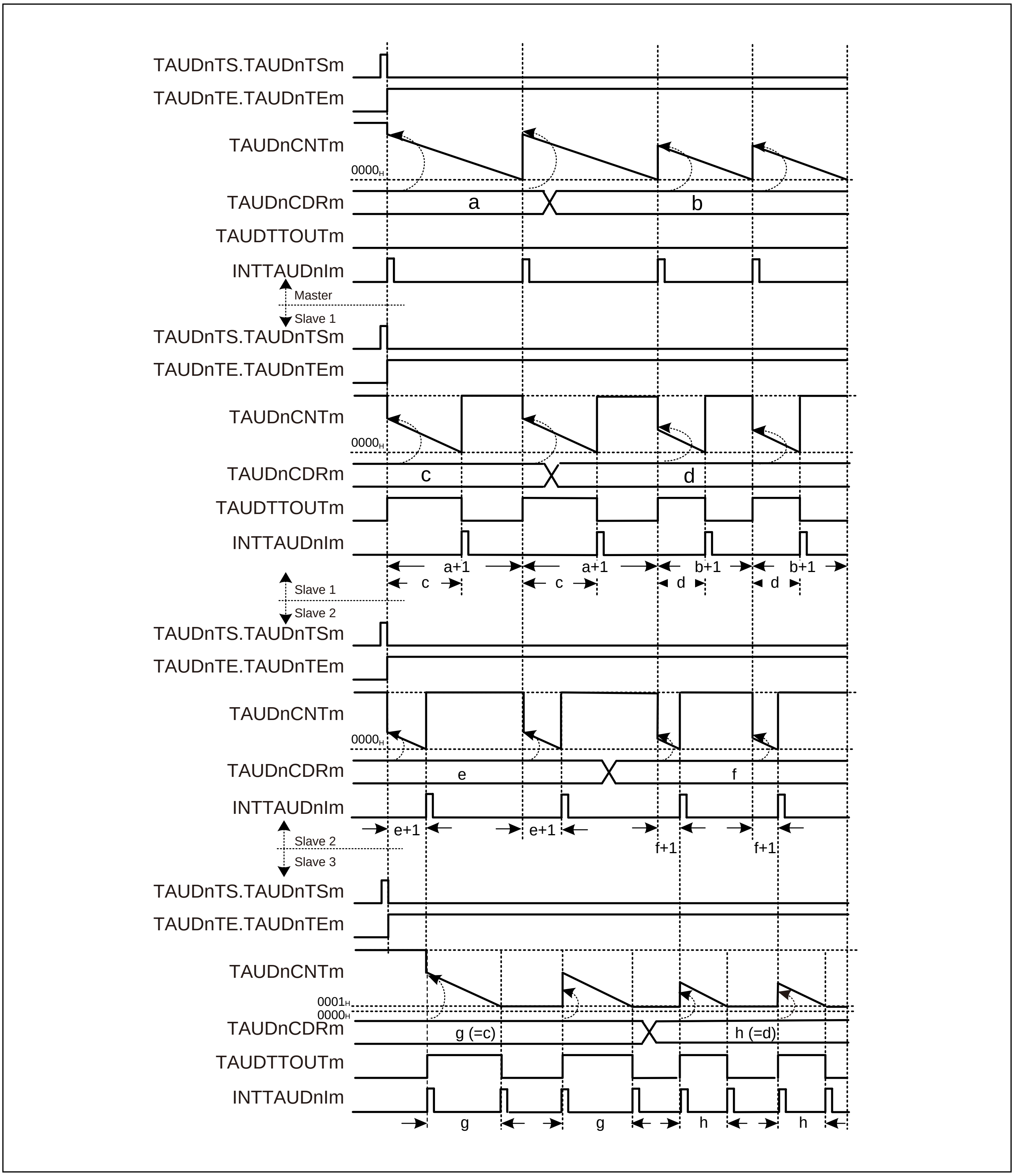


Figure 32.107 General Timing Diagram of Delay Pulse Output Function

NOTE

TAUDTTOUTm of slave channel 1 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

32.15.4.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.170 Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.171 Contents of the TAUDnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used for the master channel with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.172 Simultaneous Rewrite Settings for the Master Channel of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.4.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.173 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.174 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 1

Table 32.175 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.176 Simultaneous Rewrite Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.4.6 Register Settings for Slave Channel 2

(1) TAUDnCMORm for Slave Channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.177 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.178 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 2

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for Slave Channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.179 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.4.7 Register Settings for Slave Channel 3

(1) TAUDnCMORm for Slave Channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.180 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	101: INTTAUDnIm of upper channel (m - 1) is a start trigger regardless of master setting.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.181 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 3

Table 32.182 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.183 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.4.8 Operating Procedure for Delay Pulse Output Function

Table 32.184 Operating Procedure for Delay Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.4.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.4.5, Register Settings for Slave Channel 1**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.4.6, Register Settings for Slave Channel 2**. Slave channel 3: Set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.4.7, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.

Table 32.184 Operating Procedure for Delay Pulse Output Function

Restart Operation

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave channel 1) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 1 and 2 is loaded to TAUDnCNTm and count down. When the counter of master channel reaches 0000_H: ● INTTAUDnIm (master) is generated. ● TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1/2) to count down. ● TAUDTTOUTm (slave 1) is set. When TAUDnCNTm (slave 1) reaches 0000_H: ● INTTAUDnIm (slave 1) is generated. ● TAUDTTOUTm (slave 1) is reset. When TAUDnCNTm (slave 2) reaches 0000_H: ● INTTAUDnIm (slave 2) is generated. ● INTTAUDnIm (slave 3) is generated. ● TAUDTTOUTm (slave 3) is set. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave 3) to count down operation. When TAUDnCNTm (slave 3) reaches 0001_H: ● INTTAUDnIm (slave 3) is generated. ● TAUDTTOUTm (slave 3) is reset.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.15.4.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 32.108, Duty Cycle (Slave 3) = 100%**:

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = $000B_H$
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = $000B_H$

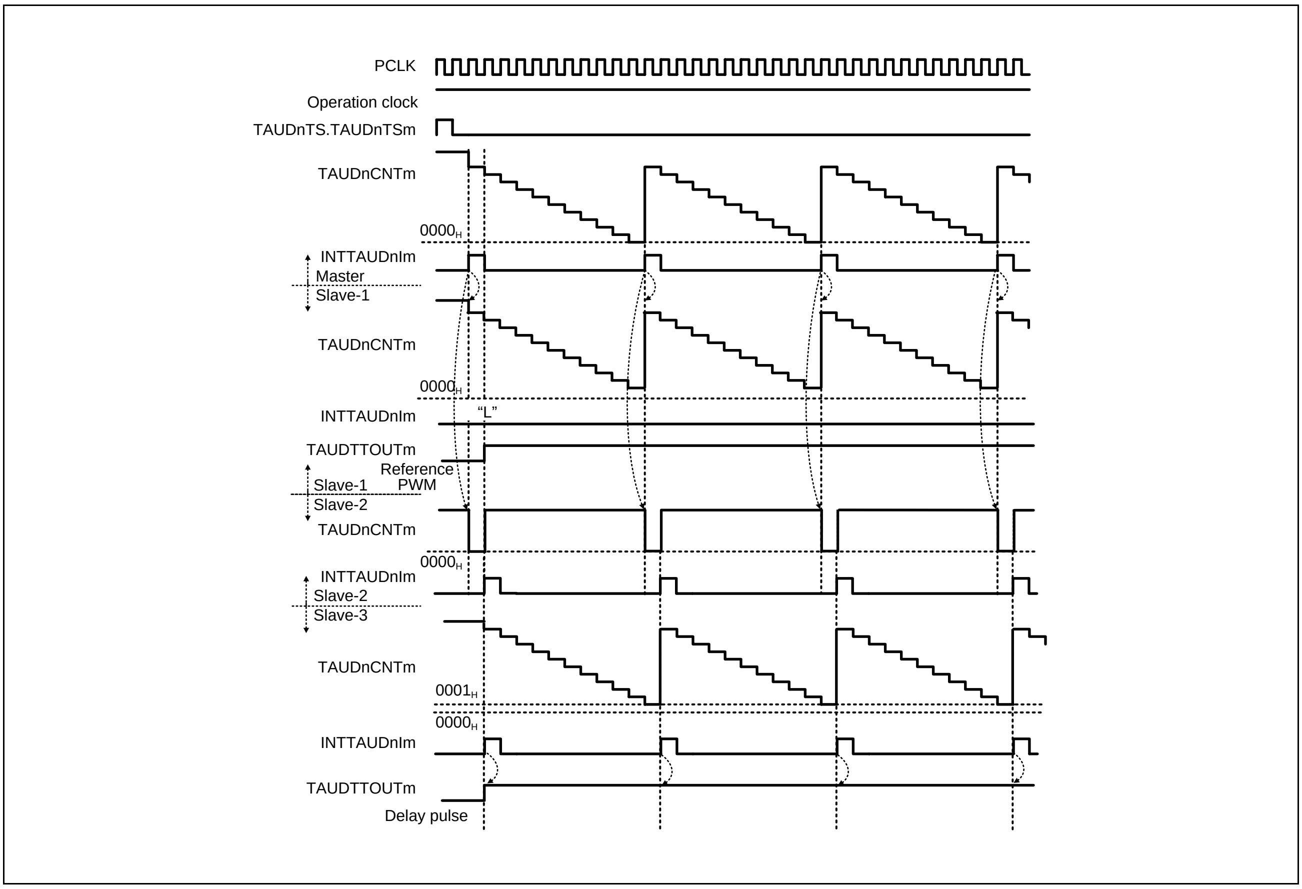


Figure 32.108 Duty Cycle (Slave 3) = 100%

- If the value of TAUDnCDRm (slaves 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of the slave channel 1 cannot reach 0000_H and cannot generate interrupts. TAUDTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUDTTOUTm (slave 1) = TAUDTTOUTm (slave 3)

The following values apply to **Figure 32.109, TAUDTTOUTm (Slave 1) = TAUDTTOUTm (Slave 3)**.

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = 0005_H
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = 0005_H

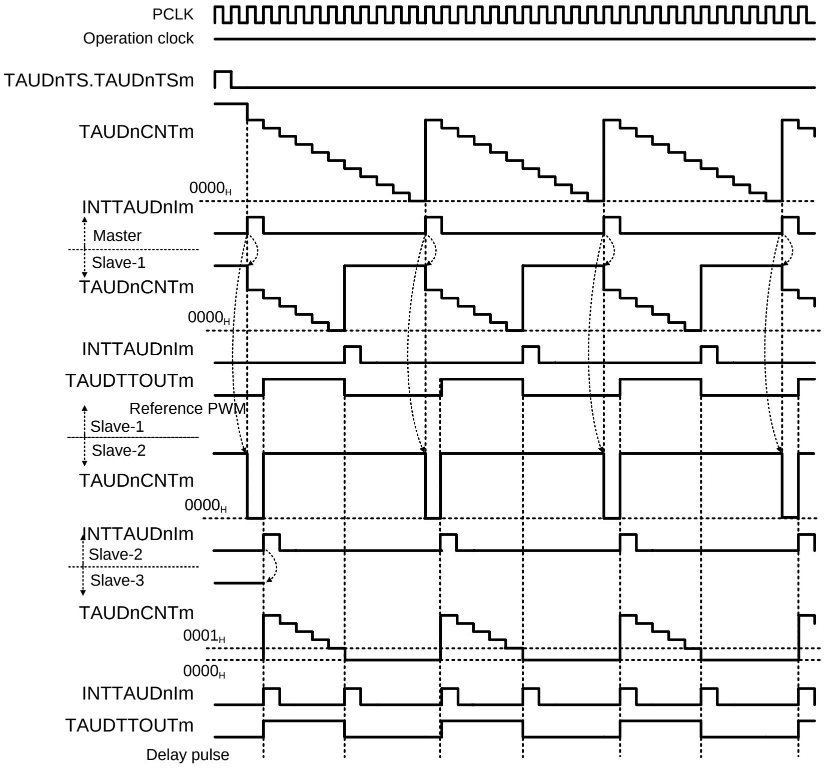


Figure 32.109 TAUDTTOUTm (Slave 1) = TAUDTTOUTm (Slave 3)

- If TAUDnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

32.15.5 Offset Trigger Output Function

32.15.5.1 Overview

Summary

This function generates a PWM output using a master channel and a slave channel, enabling the pulse width (duration) of the TAUDTTOUTm to be set. The pulse cycle is set by detecting a valid input edge of master channel. The pulse width is specified on the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See **Table 32.185, Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function**.)
- The operating mode for slave channels should be set to one-count mode. (See **Table 32.188, Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function**.)
- The output mode for slave channels should be set to synchronous channel output mode 1. (See **Section 32.7, Channel Output Modes**.)
- TAUDTTOUTm is not used with the master channel of this function.

Functional description

The counter can be enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The master channel counter (TAUDnCNTm) starts to count up from 0000_H.

- Master channel:
  When a valid TAUDTTINm input edge is detected, the current value of the counter (TAUDnCNTm) is loaded into the data register of master channel (TAUDnCDRm). INTTAUDnIm is generated and the counter restarts to count up from 0000_H.
- Slave channel:
  If INTTAUDnIm is generated on the master channel, the TAUDTTOUTm (slave) signal is set and the counter of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts to count down from this value.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

32.15.5.2 Equations

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)] × 100

- Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
- Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

32.15.5.3 Block Diagram and General Timing Diagram

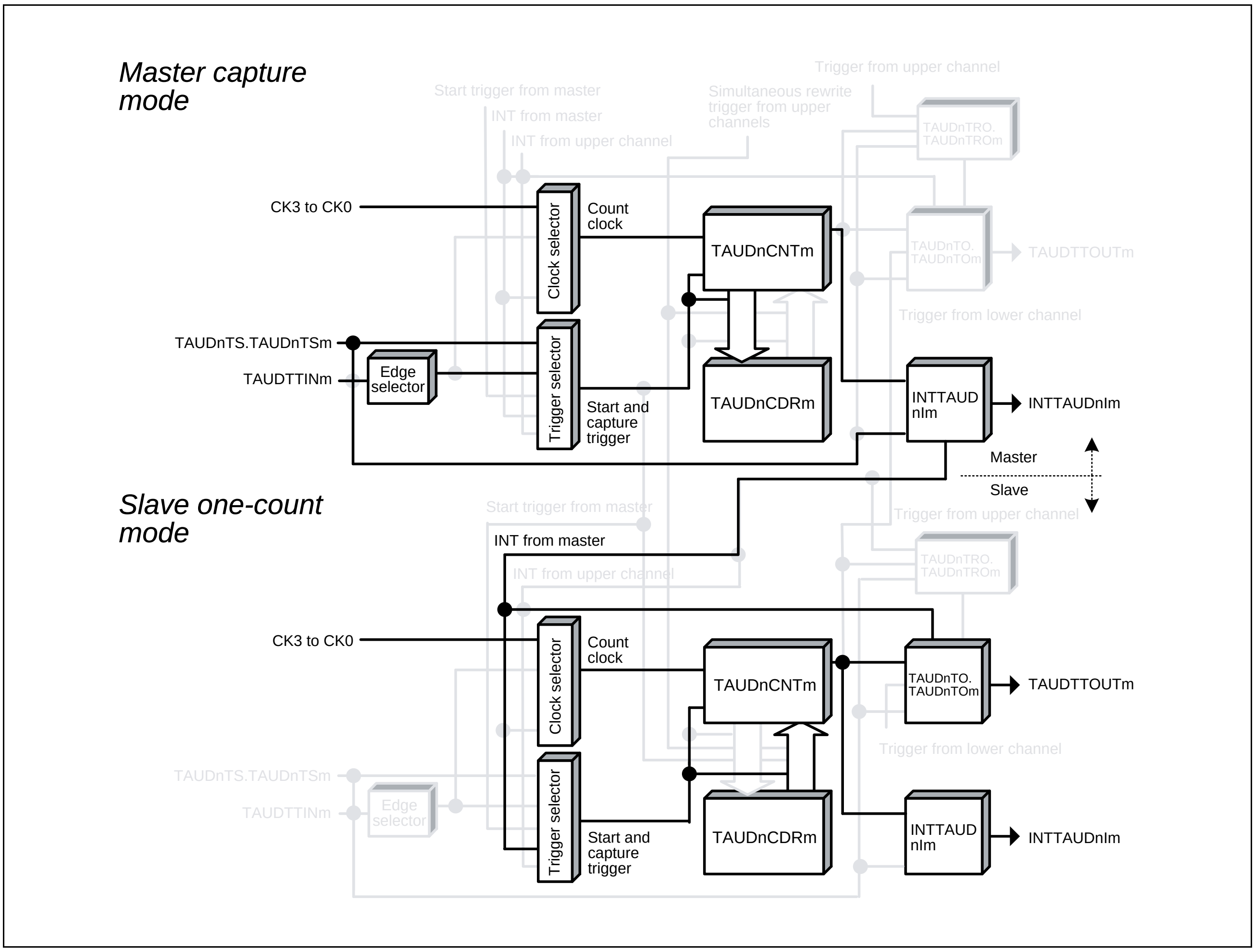


Figure 32.110 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

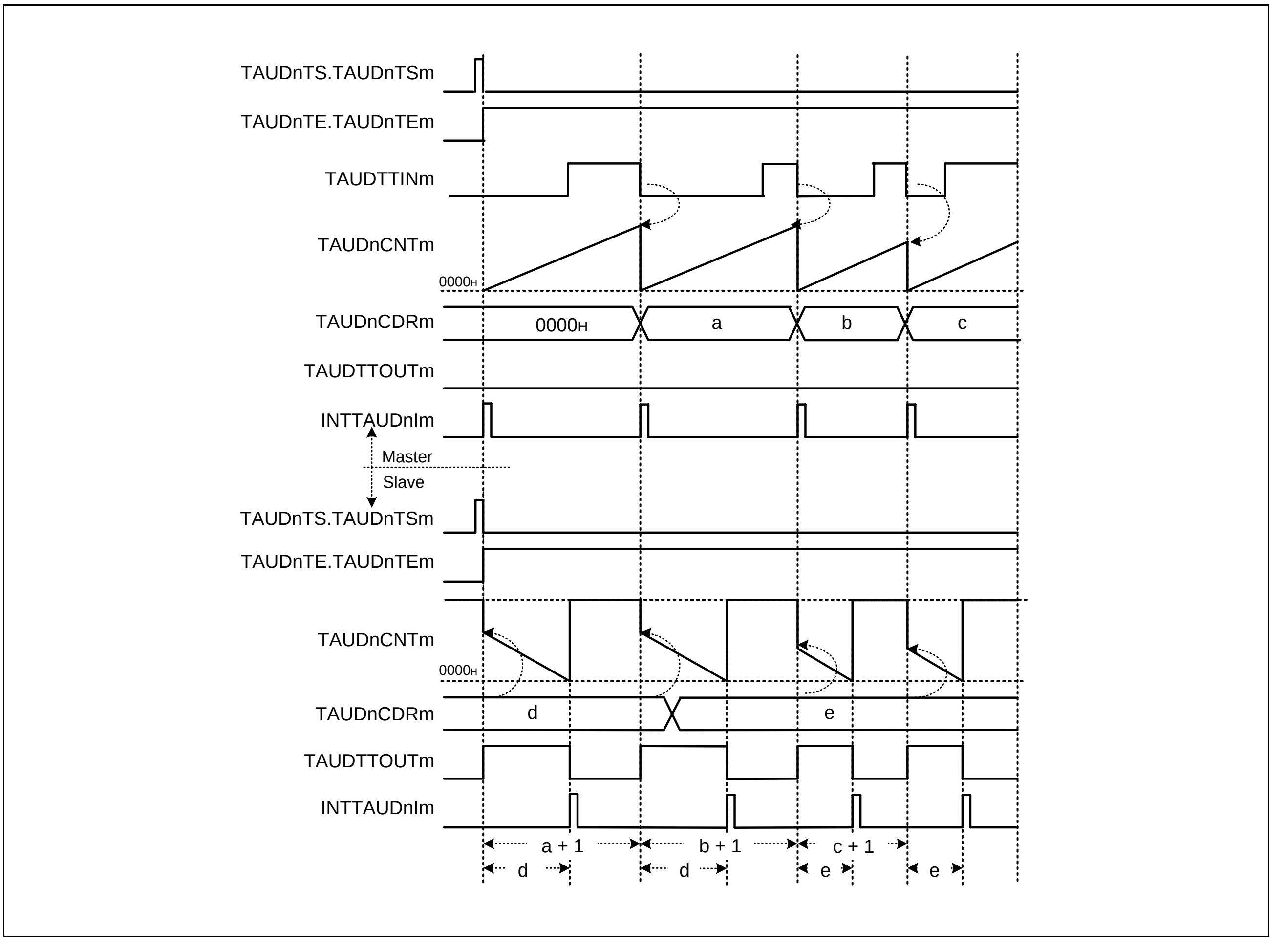


Figure 32.111 General Timing Diagram of Offset Trigger Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

32.15.5.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.185 Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	11: Capture register is updated upon detection of a valid TAUDTTINm input edge or when a counter overflow occurs: – Detection of valid TAUDTTINm input edge: The counter value is written into TAUDnCDRm. – Occurrence of overflow: FFFF$_H$ is written into TAUDnCDRm. A valid TAUDTTINm input edge to be detected next is ignored. TAUDnCSRm.TAUDnOVF is set when a counter overflow occurs, and cleared by setting TAUDnCSCm.TAUDnCLOV = 1.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.186 Contents of the TAUDnCMURm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

Table 32.187 Simultaneous Rewrite Settings for the Master Channel of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.15.5.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.188 Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.189 Contents of the TAUDnCMURm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 32.190 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

Table 32.191 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

32.15.5.6 Operating Procedure for Offset Trigger Output Function

Table 32.192 Operating Procedure for Offset Trigger Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.5.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 32.15.5.5, Register Settings for Slave Channels**. The TAUDnCDRm register of master channel functions as a capture register. Set the value of TAUDnCDRm register of slave channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start: ● TAUDnCNTm (master) counts up. ● TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCSCm.TAUDnCLOV can be set to 1. TAUDnCDRm of slave channel can be changed after the generation of INTTAUDnIm (master). TAUDnCNTm and TAUDnCSRm can be read at any time.	When TAUDnCNTm of the slave = 0000$_H$: ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is reset, and the counter of slave channel stops. When TAUDTTINm input edge is detected on the master channel: ● INTTAUDnIm (master) is generated. ● TAUDnCNTm (master) is reset to 0000$_H$ and then continues count operation subsequently. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. ● TAUDTTOUTm (slave) is set.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

32.15.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

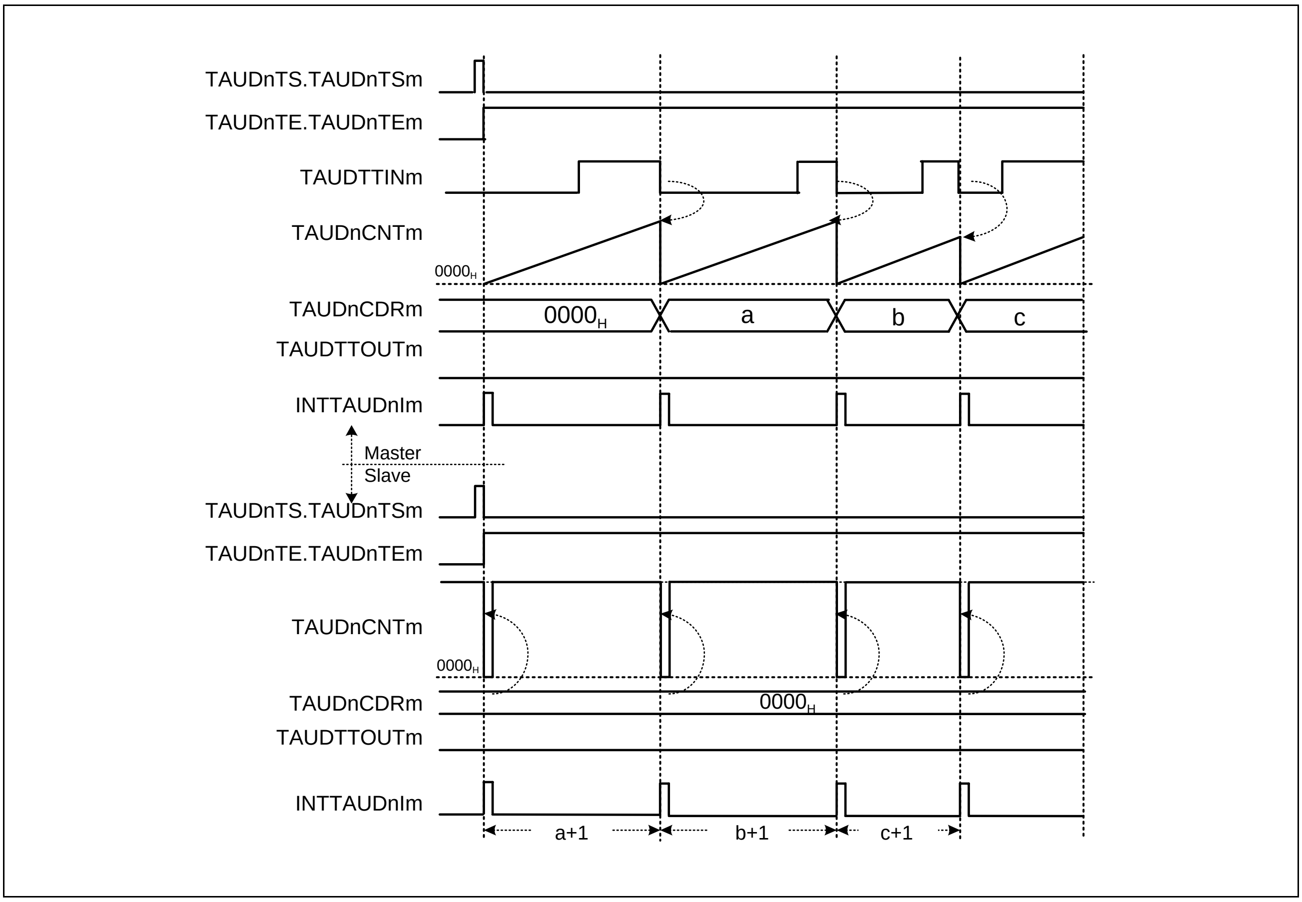


Figure 32.112 TAUDnCDRm (Slave) = 0000_H

- When TAUDnCDRm (slave) = 0000_H, 0000_H is written to TAUDnCNTm every time the master channel generates an interrupt (INTTAUDnIm), and TAUDnCNTm cannot start to count. The TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The slave and the master channels generate interrupts in the same cycle.

(2) Duty cycle = 100%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

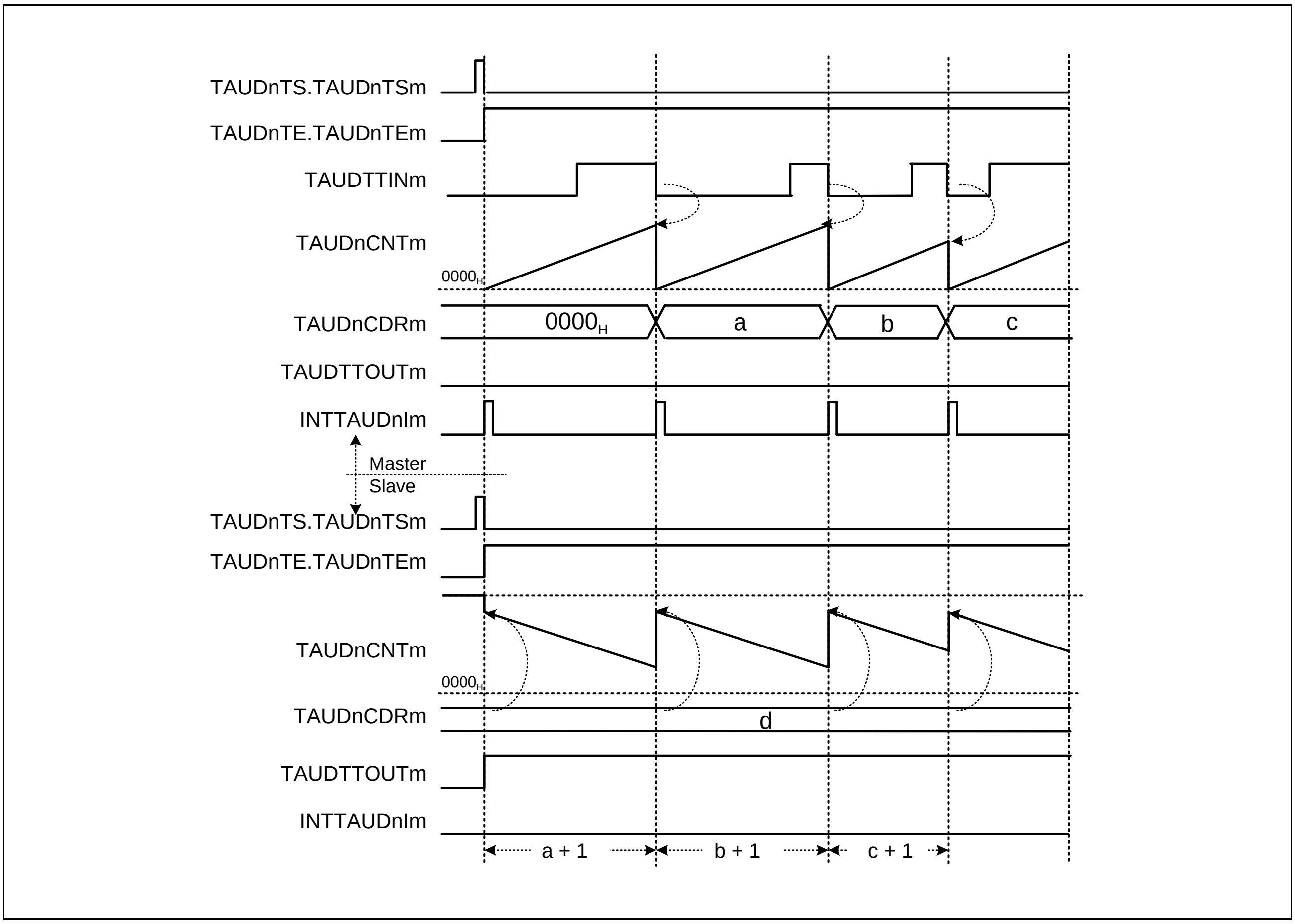


Figure 32.113 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If the value TAUDnCDRm (slave) is higher than the interval of valid input edges, the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUDTTOUTm remains at active state.

32.15.6 A/D Conversion Trigger Output Function Type 1

32.15.6.1 Overview

Summary

This function is identical to **Section 32.15.1, PWM Output Function**, except that TAUDTTOUTm is not output.

This function is enabled by setting the channel output mode for the slave to independent channel output mode controlled by software.

32.15.6.2 Block Diagram and General Timing Diagram

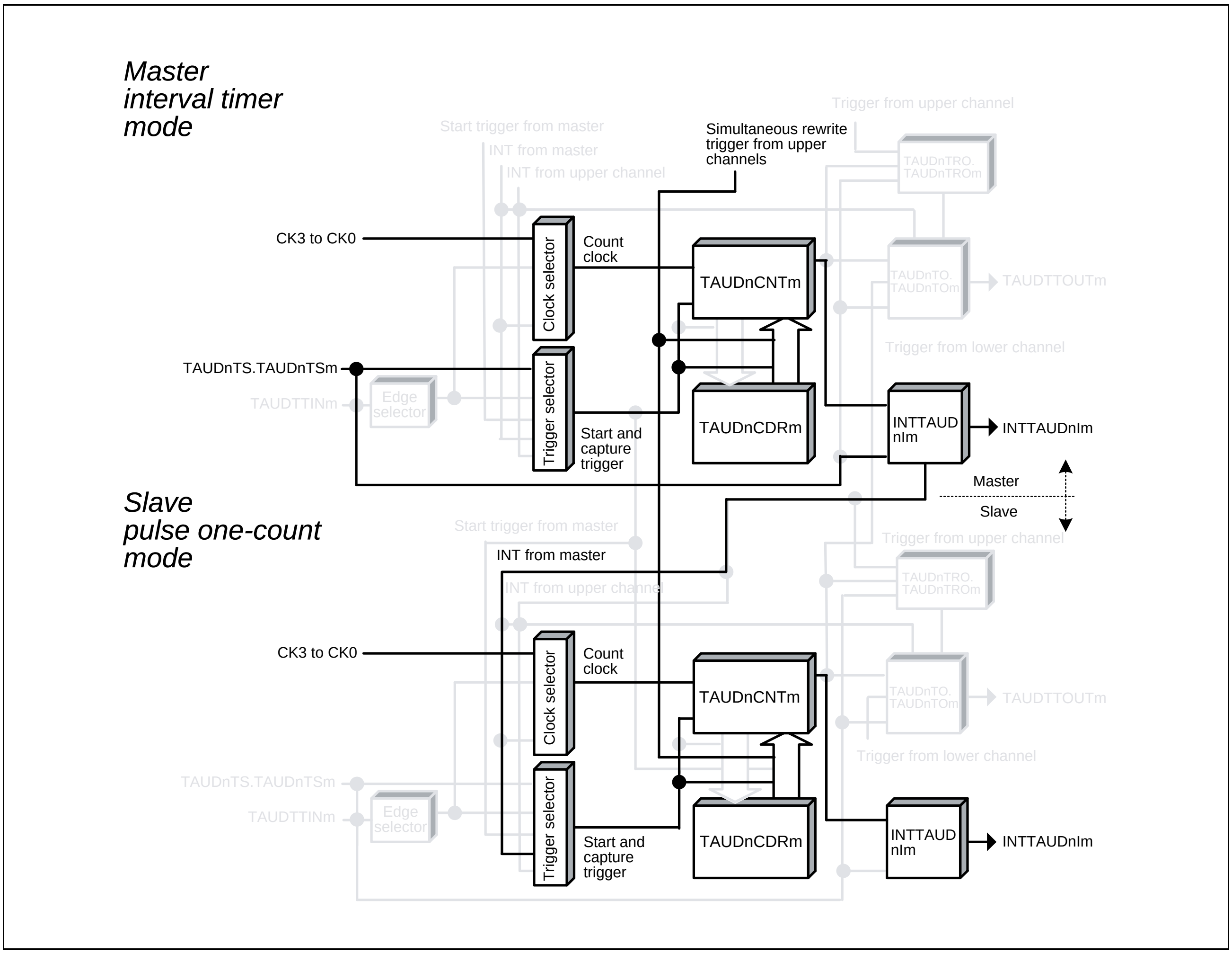


Figure 32.114 Block Diagram of A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

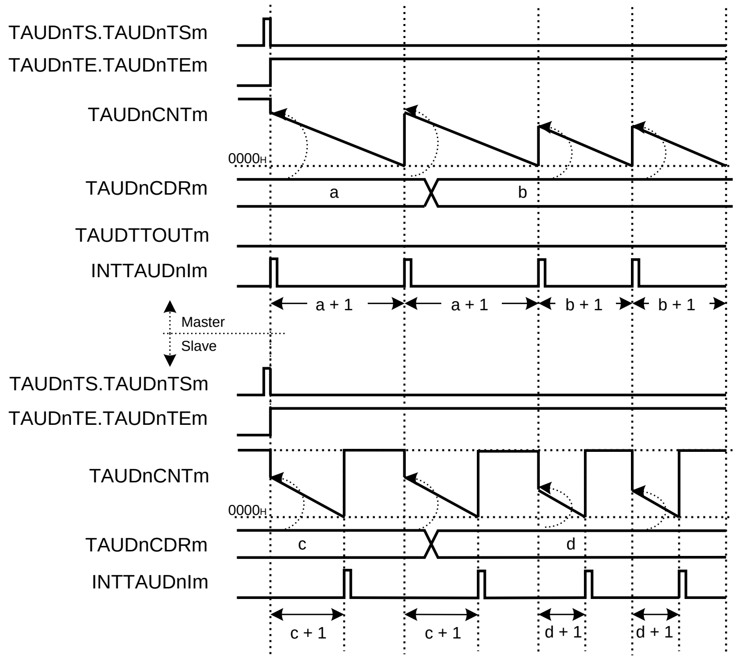


Figure 32.115 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

32.15.7 Triangle PWM Output Function

32.15.7.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels respectively.

The master channel generates a carrier cycle. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slave counter.

Prerequisites

- Two channels
- The operating mode for the master channels should be set to interval timer mode. (See **Table 32.193, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function**.)
- The operating mode for slave channels should be set to count-up/-down mode. (See **Table 32.197, Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function**.)
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 32.7, Channel Output Modes**.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 32.7, Channel Output Modes**.)
- The following settings allow the TAUDTTOUTm signal to be at high level during the down status of a carrier cycle.
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

Functional description

The counters are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for every channel. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current values of TAUDnCDRm (master and slave) are loaded into TAUDnCNTm (master and slave) and the counters start counting down from these values. When the TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and TAUDTTOUTm signal of master toggles.

- Master channel:
  When the counter of master channel reaches 0000_H (pulse cycle time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and counts down.
- Slave channel:
  If INTTAUDnIm is generated on the master channel, the counter of the slave channel is triggered.
  - If the slave counter is counting down, the count direction changes.
  - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts to count down.

When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUDnIm is generated and the TAUDTTOUTm (slave) signal is set/reset.
The counter continues count-up/-down and waits for the next INTTAUDnIm of the master channel.

Setting TAUDnTOL.TAUDnTOLm allows TAUDTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

Conditions

This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.

32.15.7.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle 100 [%] =
[(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave))/(TAUDnCDRm (master) + 1)] × 100

- Duty cycle = [%]
  TAUDnCDRm (slave) = 0000_H
- Duty cycle = 0%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

32.15.7.3 Block Diagram and General Timing Diagram

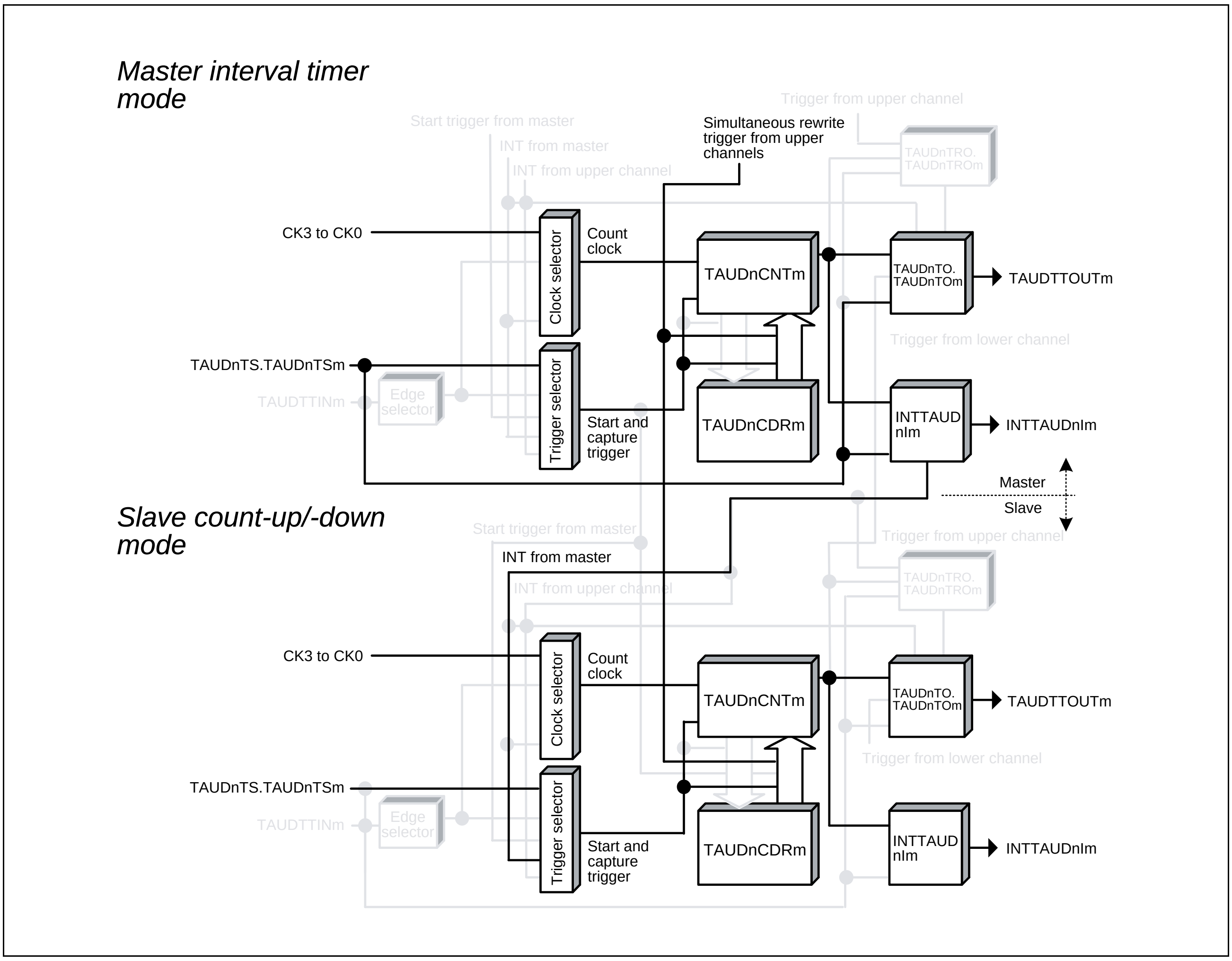


Figure 32.116 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

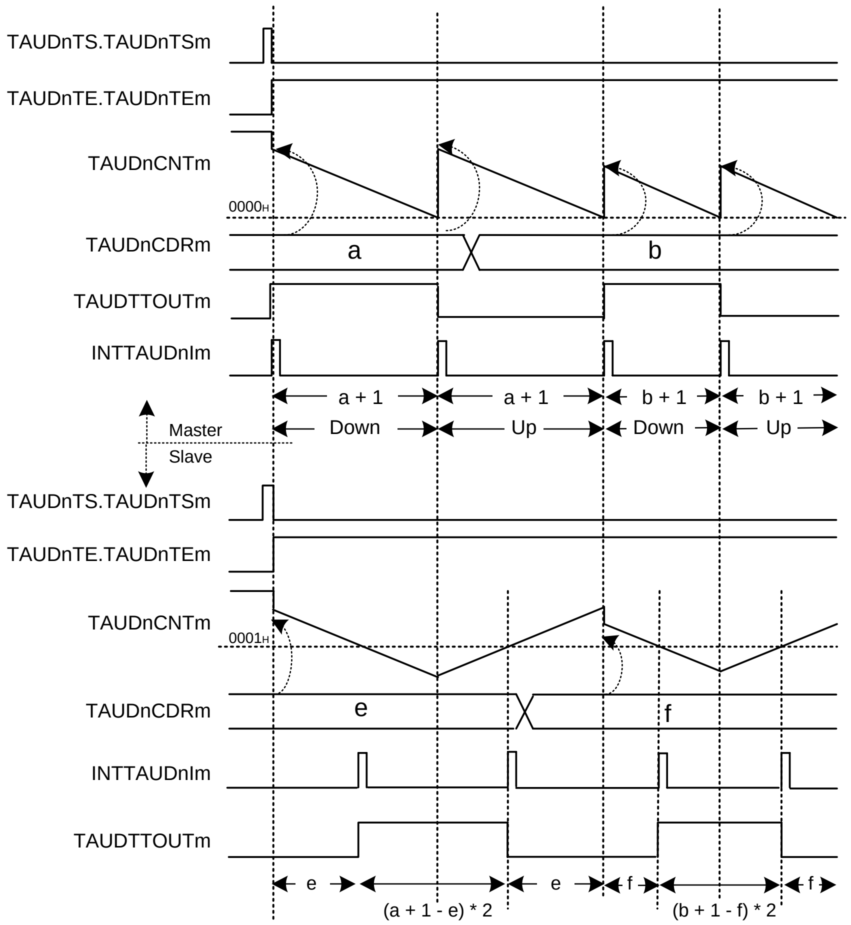


Figure 32.117 General Timing Diagram of Triangle PWM Output Function

32.15.7.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.193 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.194 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 32.195 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.196 Simultaneous Rewrite Settings for the Master Channel of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

32.15.7.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.197 Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.198 Contents of the TAUDnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 32.199 Control Bit Settings in Synchronous Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.200 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when the master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.7.6 Operating Procedure for Triangle PWM Output Function

Table 32.201 Operating Procedure for Triangle PWM Output Function

		Operation	TAUDn Status
	Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.7.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.7.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
	During Operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master and slave channels is loaded into TAUDnCNTm to count down. When the counter of master channel reaches 0000_H: ● INTTAUDnIm (master) is generated. ● TAUDTTOUTm (master) is toggled. ● TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave) or counting is started in opposite direction. When TAUDnCNTm of slave channel reaches 0001_H: ● INTTAUDnIm (slave) is generated. ● TAUDTTOUTm (slave) is set in the count-down status or reset in count-up status.
	Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.15.7.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 6_H

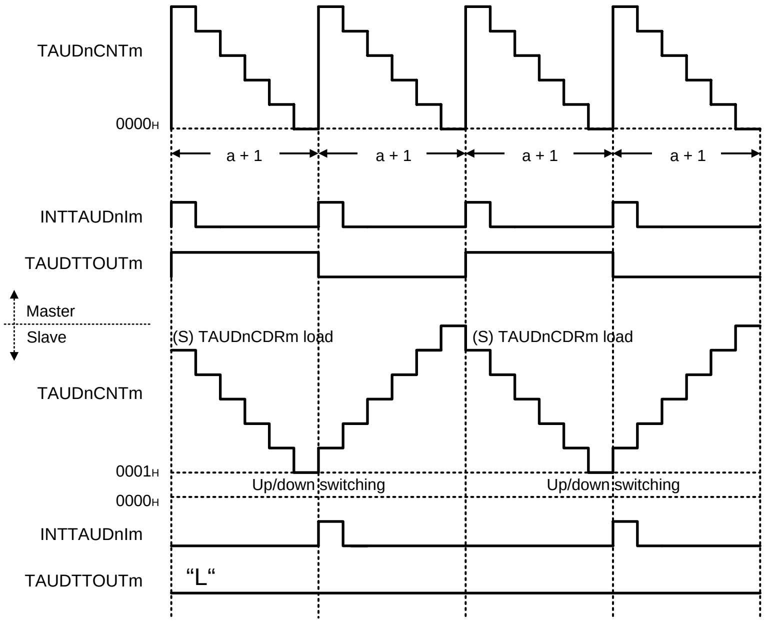


Figure 32.118 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value ≥ TAUDnCDRm (master) value +1, INTTAUDnIm of the slave channel is not generated while counting down. TAUDTTOUTm remains low because there is no set signal to be detected.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 0_H

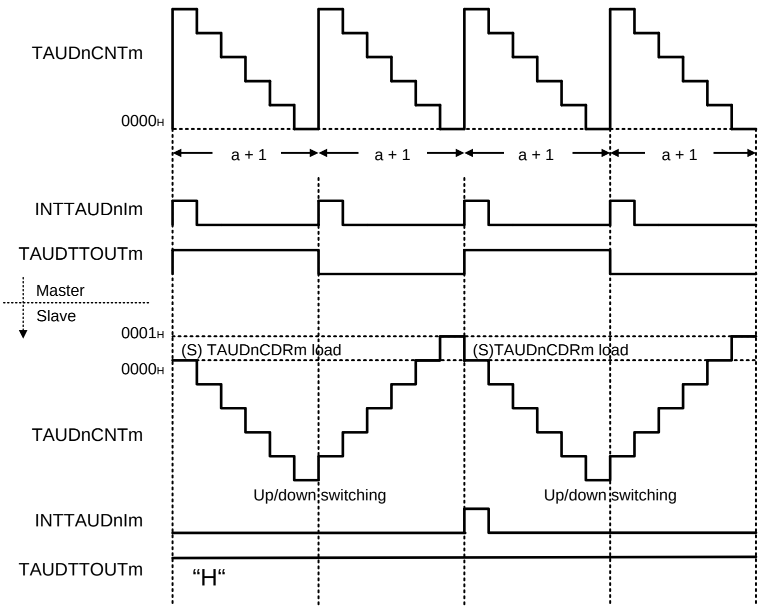


Figure 32.119 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave) = 0000_H, INTTAUDnIm of the slave channel is not generated while counting up. TAUDTTOUTm remains high because there is no reset signal to be detected.

32.15.8 Triangle PWM Output Function with Dead Time

32.15.8.1 Overview

Summary

This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals with dead time are output via TAUDTTOUTm of the slave channels 2 and 3, enabling the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on the master channel. The first pulse controls the down status of the slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUDTTOUTm of slave channels to be set/reset. Depending on the settings of TAUDnTDL.TAUDnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUDTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select even channel CH (a) and odd channel CH (a + 1).
- The operating mode for the master channel should be set to interval timer mode. (See **Table 32.203, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**)
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even channel (a), and slave channel 3 is an odd channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operating mode for slave channel 2 should be set to count-up/-down mode (See **Table 32.207, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time**). Slave channel 2 should be an even channel.
- The operating mode for slave channel 3 should be set to one-count mode (See **Table 32.211, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time**). Slave channel 3 should be an odd channel.
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 32.7, Channel Output Modes**)
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See **Section 32.7, Channel Output Modes**)
- The following settings make a TAUDTTOUTm signal at high level during the down status of the carrier cycle:
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

NOTE

The triangle PWM output function with dead time does not use slave channel 1. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counter starts by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUDTTOUTm signal is toggled.

- Master channel:
  When the counter of the master channel reaches 0000_H, an INTTAUDnIm is generated and the TAUDTTOUTm signal is toggled. The TAUDnCDRm value is reloaded to continue counting down.
- Slave channel 2:
  If INTTAUDnIm is generated on the master channel, the counter of slave channel 2 is triggered.
  - If the slave counter is counting down, the counting direction changes.
  - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts counting down.

The counter continues to count down/up and waits for the next INTTAUDnIm of the master channel.
When the counter value of slave channel 2 reaches 0001_H, INTTAUDnIm is generated

- Slave channel 3:
  If INTTAUDnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDnCDRm (slave 3) is loaded into TAUDnCNTm (slave 3) and the counter starts to count down from the TAUDnCDRm value.
  When the counter reaches 0000_H, INTTAUDnIm occurs. The counter returns to $FFFF_H$ and waits for the next INTTAUDnIm of slave channel 2.

As described in **Table 32.202, Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (right after occurrence of an interrupt or after dead time has elapsed) depends on the TAUDnTDL.TAUDnTDLm setting of the corresponding channel.

The setting of TAUDnTOL.TAUDnTOLm also determines whether a high level signal (TAUDnTOL.TAUDnTOLm = 0) or a low level signal (TAUDnTOL.TAUDnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

TAUDTTOUTm can be 100% output by setting the TAUDnCDRm value of slave channel 2 to 0000_H.

Conditions

This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.

TAUDnTOL.TAUDnTOLm and TAUDnTDL.TAUDnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDnTDL.TAUDnTDLm.

Table 32.202 Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

TAUDnTDL. TAUDnTDLm	Count Direction of Slave Channel 2 upon Occurrence of Interrupt	TAUDTTOUTm Set/Reset Timing
0	Down	Set after elapse of dead time
	Up	Reset right after interrupt occurs
1	Down	Set right after interrupt occurs
	Up	Reset after elapse of dead time

32.15.8.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (normal phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

32.15.8.3 Block Diagram and General Timing Diagram

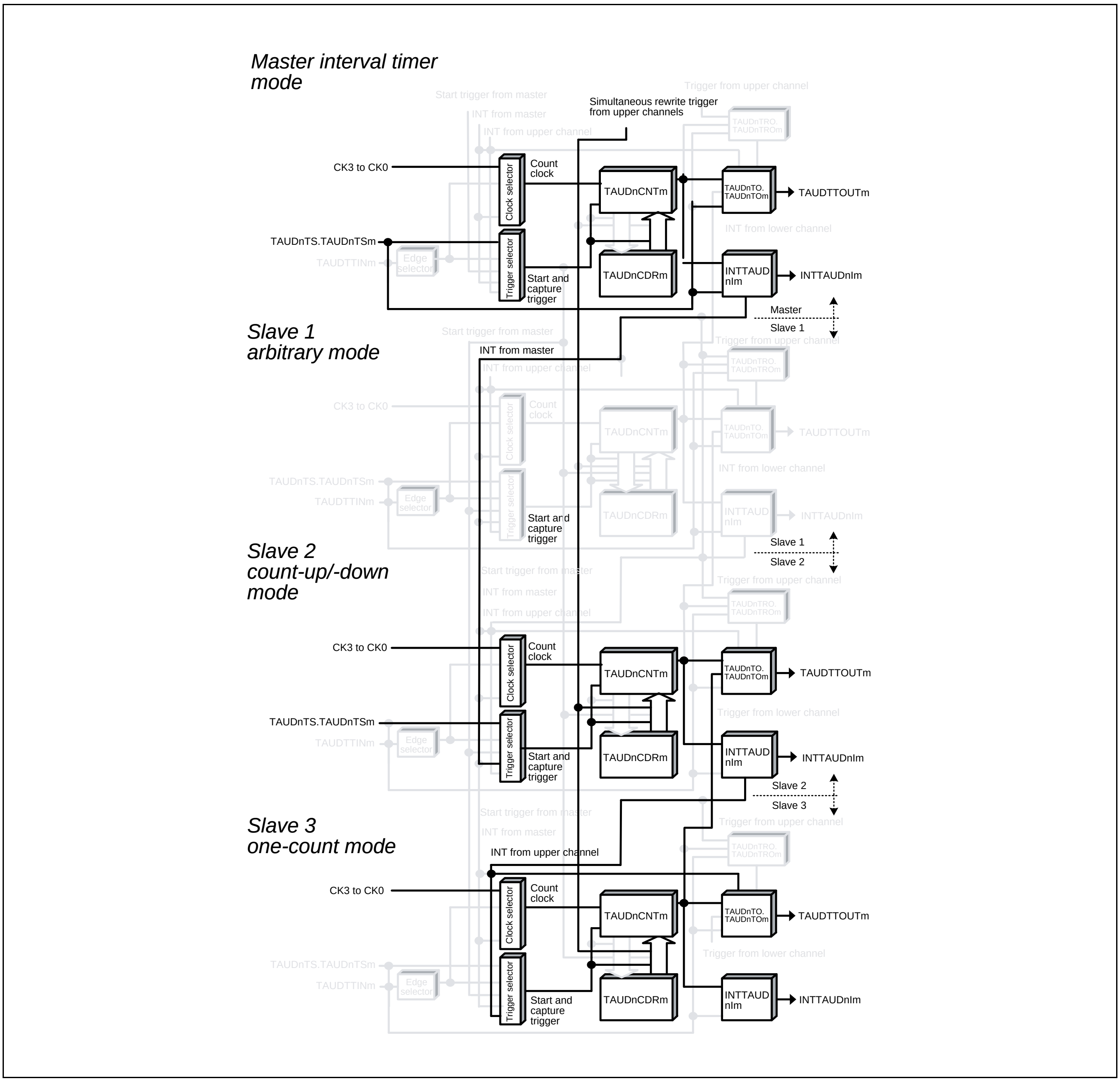


Figure 32.120 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Slave channel 2:
  - INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
  - TAUDnTDL.TAUDnTDLm = 0
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

- Slave channel 3:
  - Enables start trigger detection during counting (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnTDL.TAUDnTDLm = 1
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

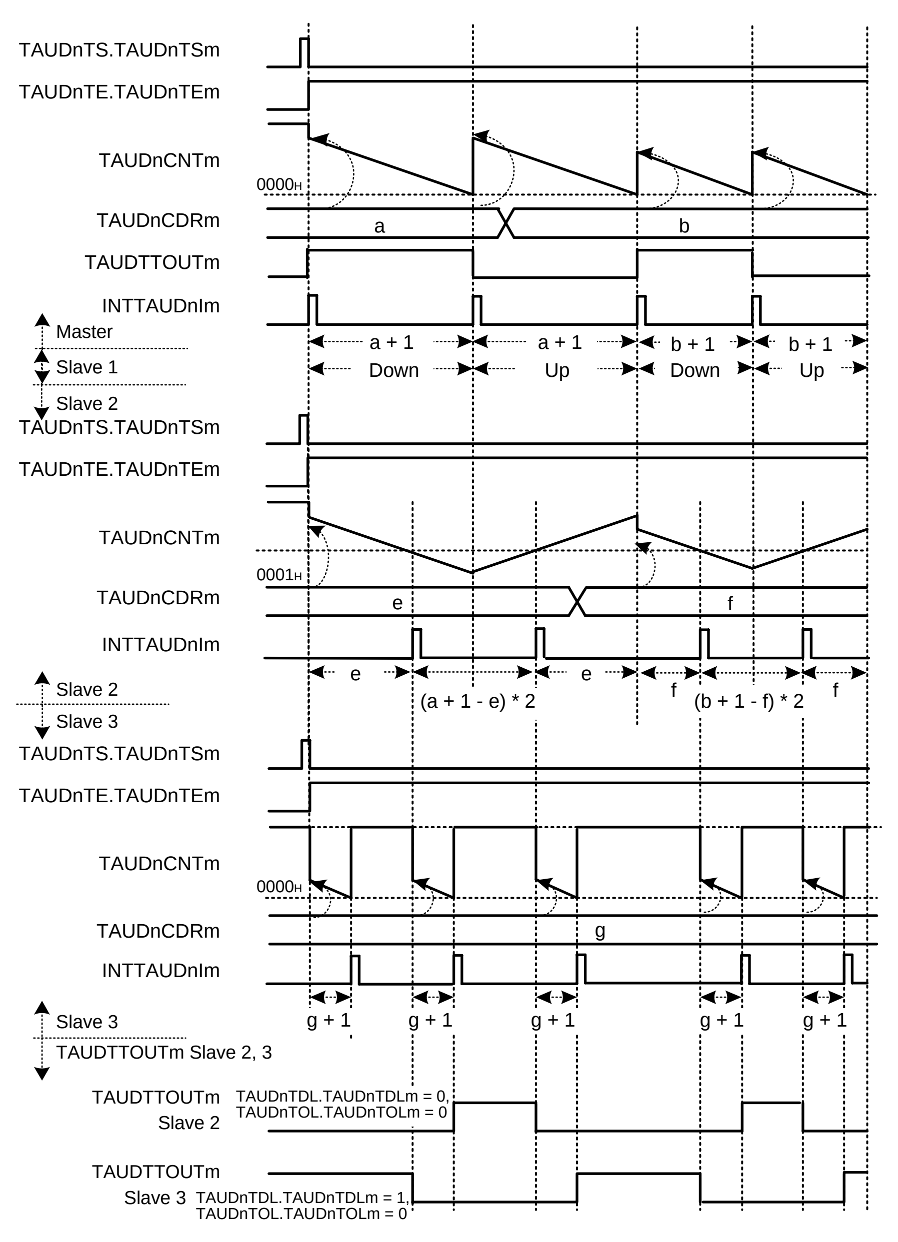


Figure 32.121 General Timing Diagram of Triangle PWM Output Function with Dead Time

32.15.8.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.203 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.204 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 32.205 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.206 Simultaneous Rewrite Setting for the Master Channel of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

32.15.8.5 Register Settings for Slave Channel 2

(1) TAUDnCMORm for Slave Channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.207 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for Slave Channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.208 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 2

Table 32.209 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous Rewrite for Slave Channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.210 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.8.6 Register Settings for Slave Channel 3

(1) TAUDnCMORm for Slave Channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.211 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time output signal of the TAUDTTOUTm generation unit
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.212 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 3

Table 32.213 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous Rewrite for Slave Channel 3

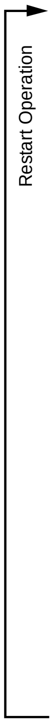
Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.214 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.8.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 32.215 Operating Procedure for Triangle PWM Output Function with Dead Time



	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.8.4, Register Settings for the Master Channel**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.8.5, Register Settings for Slave Channel 2**. Slave channel 3: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.8.6, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channel 2 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: ● INTTAUDnIm (master) is generated. ● TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. ● TAUDnCDRm value is reloaded into TAUDnCNTm (slave 2) or counting is started in opposite direction. When TAUDnCNTm of slave channel 2 reaches 0001_H: ● INTTAUDnIm (slave 2) is generated. ● TAUDnCDRm value of slave channel 3 is loaded into TAUDnCNTm perform counting down. When TAUDnCNTm of slave channel 3 reaches 0000_H: ● INTTAUDnIm is generated.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.15.8.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram in **Figure 32.122, TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

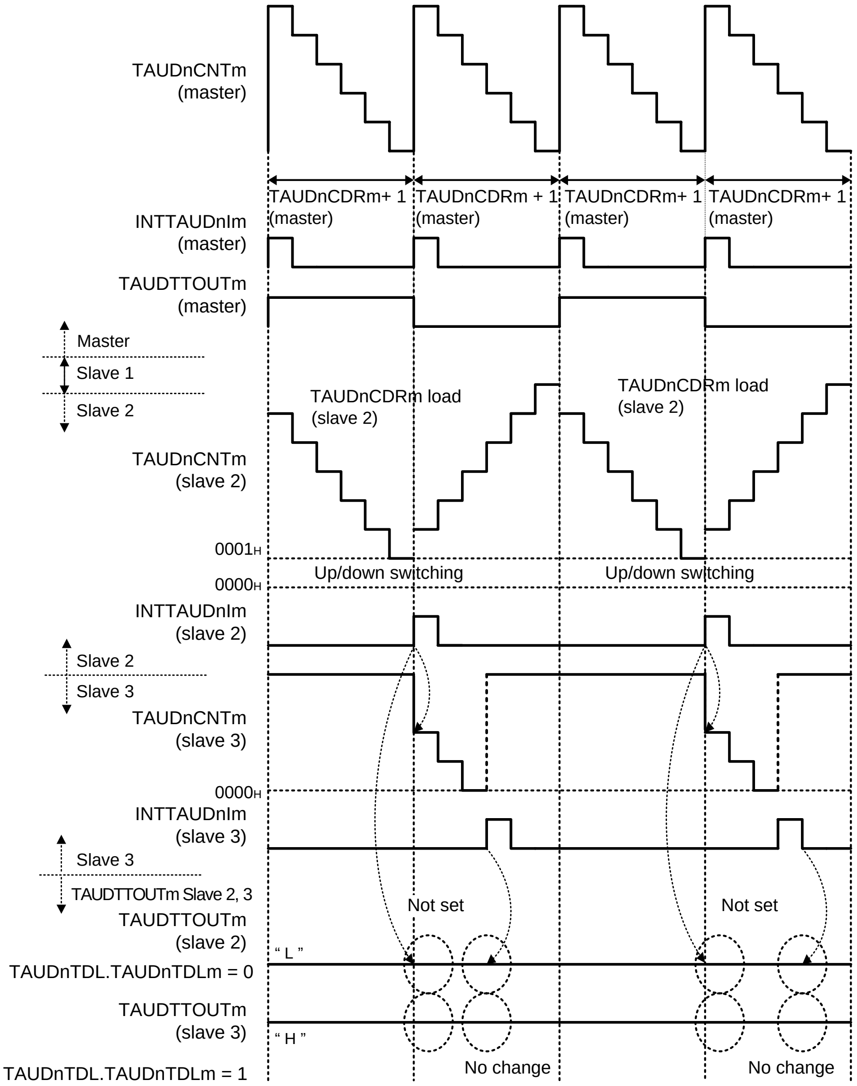


Figure 32.122 TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave 2) is greater than TAUDnCDRm (master), the counter of slave channel does not reach 0000_H while counting down. Therefore, TAUDTTOUTm signal is not set/reset and remains in the initial state. This signal becomes a reset signal because an interrupt occurs on slave channel 2 during count-up operation.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram in **Figure 32.123, TAUDnCDRm (Slave) = 0000H**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

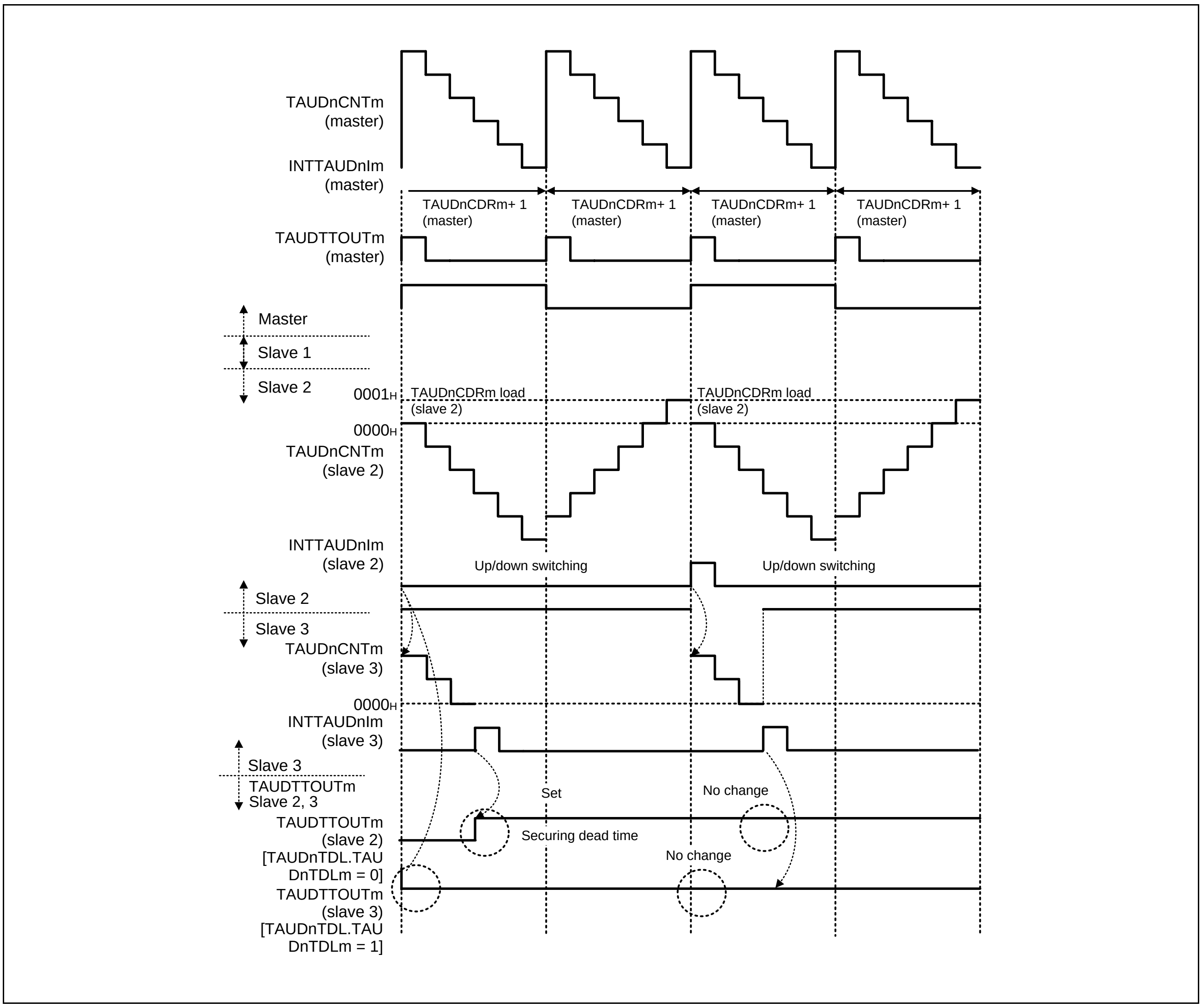


Figure 32.123 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave 2) = 0000_H, the slave channel counter does not reach 0001_H while counting up. Therefore, no INTTAUDnIm is generated during count-up operation.
  - The set conditions for a channel with TAUDnTDL.TAUDnTDLm = 0 are met after elapse of dead time. TAUDTTOUTm is left in a newly set state even if a set/reset is made because no reset conditions are satisfied on such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts. However, TAUDTTOUTm is left in an initial state on the slave channel with TAUDnTDL.TAUDnTDLm = 1 because no reset conditions are satisfied on that channel.

32.15.9 A/D Conversion Trigger Output Function Type 2

32.15.9.1 Overview

Summary

This function is identical to **Section 32.15.7, Triangle PWM Output Function**, except that TAUDTTOUTm is not output.

This function is enabled by setting channel output mode for the slave to independent channel output mode controlled by software.

32.15.9.2 Block Diagram and General Timing Diagram

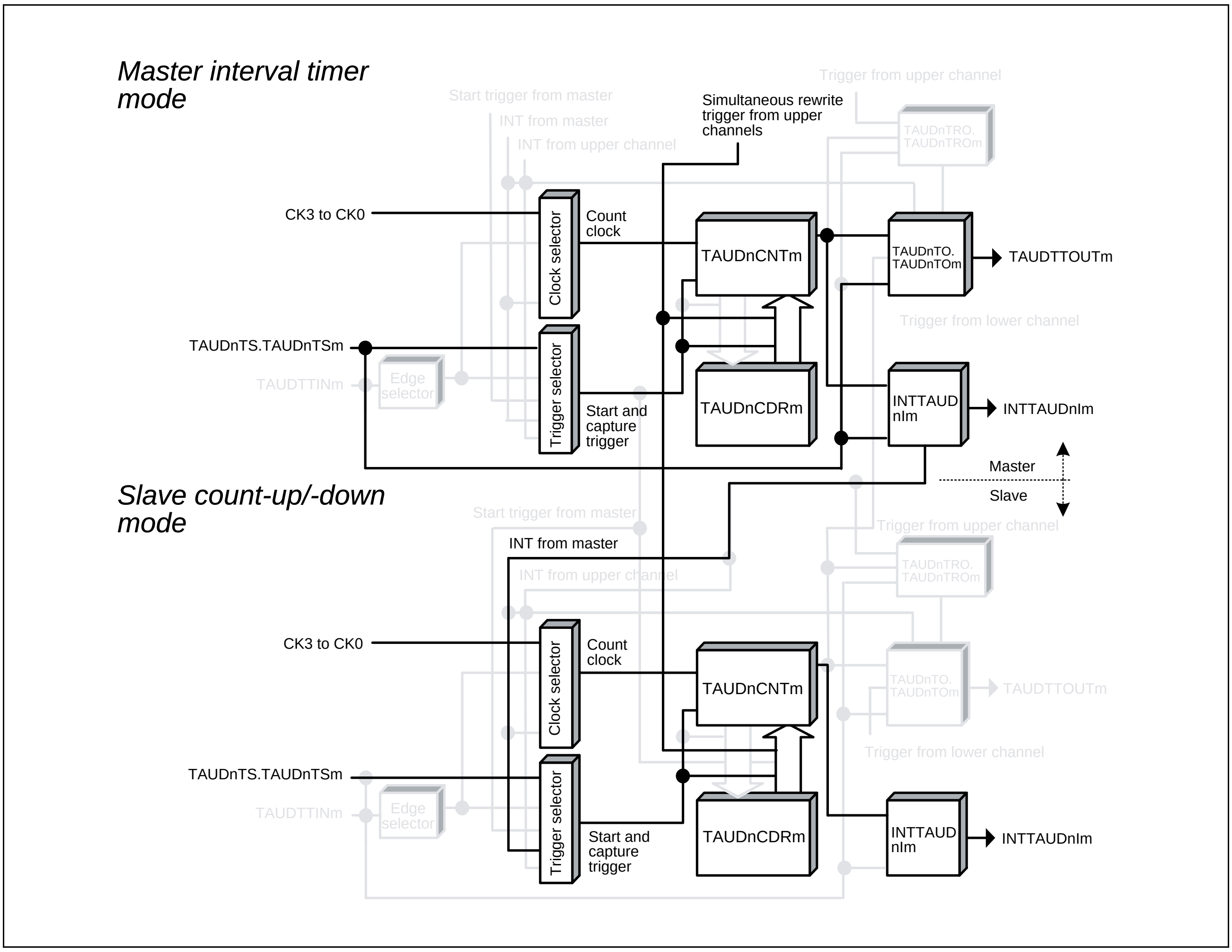


Figure 32.124 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

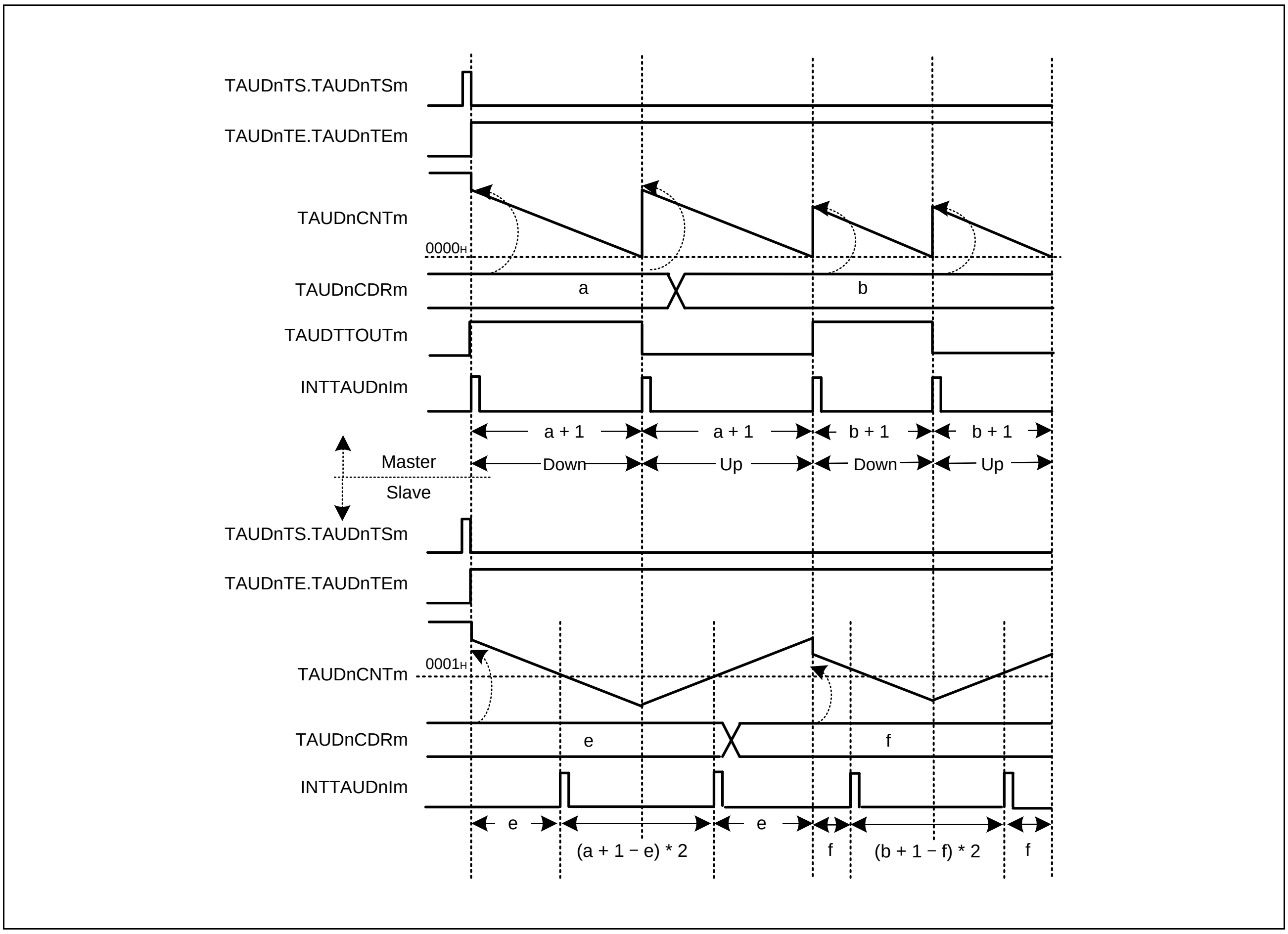


Figure 32.125 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

32.15.10 Interrupt Request Signals Culling Function

32.15.10.1 Overview

Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The interrupt request signals culling function is a sub function of the following functions:

- PWM Output Function (See **Section 32.15.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 32.15.7, Triangle PWM Output Function**)
- Triangle PWM Output Function with Dead Time (See **Section 32.15.8, Triangle PWM Output Function with Dead Time**)

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 32.216, Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function**)
- The operation mode of the slave channel must be set to Event Count Mode. (See **Table 32.219, Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function**)
- This function does not use TAUDTTOUTm.

Functional description

The counters (master and slave) are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for both channels. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDnCDRm) are written to the counter (TAUDnCNTm).

- Master channel:
  When the counter of the master channel reaches 0000_H, INTTAUDnIm is generated and TAUDnCDRm value is reloaded to TAUDnCNTm.
- Slave channel:
  Every time the master channel generates an INTTAUDnIm, the counter of the slave channel decrements by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUDnCNTm (slave) to reload the value of TAUDnCDRm, and an INTTAUDnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm of master and slave channels stops but retains its value.

Conditions

This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.

32.15.10.2 Equations

Interrupt division operator = TAUDnCDRm (slave channel)

- One INTTAUDnIm is generated for the INTTAUDnIm count of the master channel defined by TAUDnCDRm (slave channel) + 1.

32.15.10.3 Block Diagram and General Timing Diagram

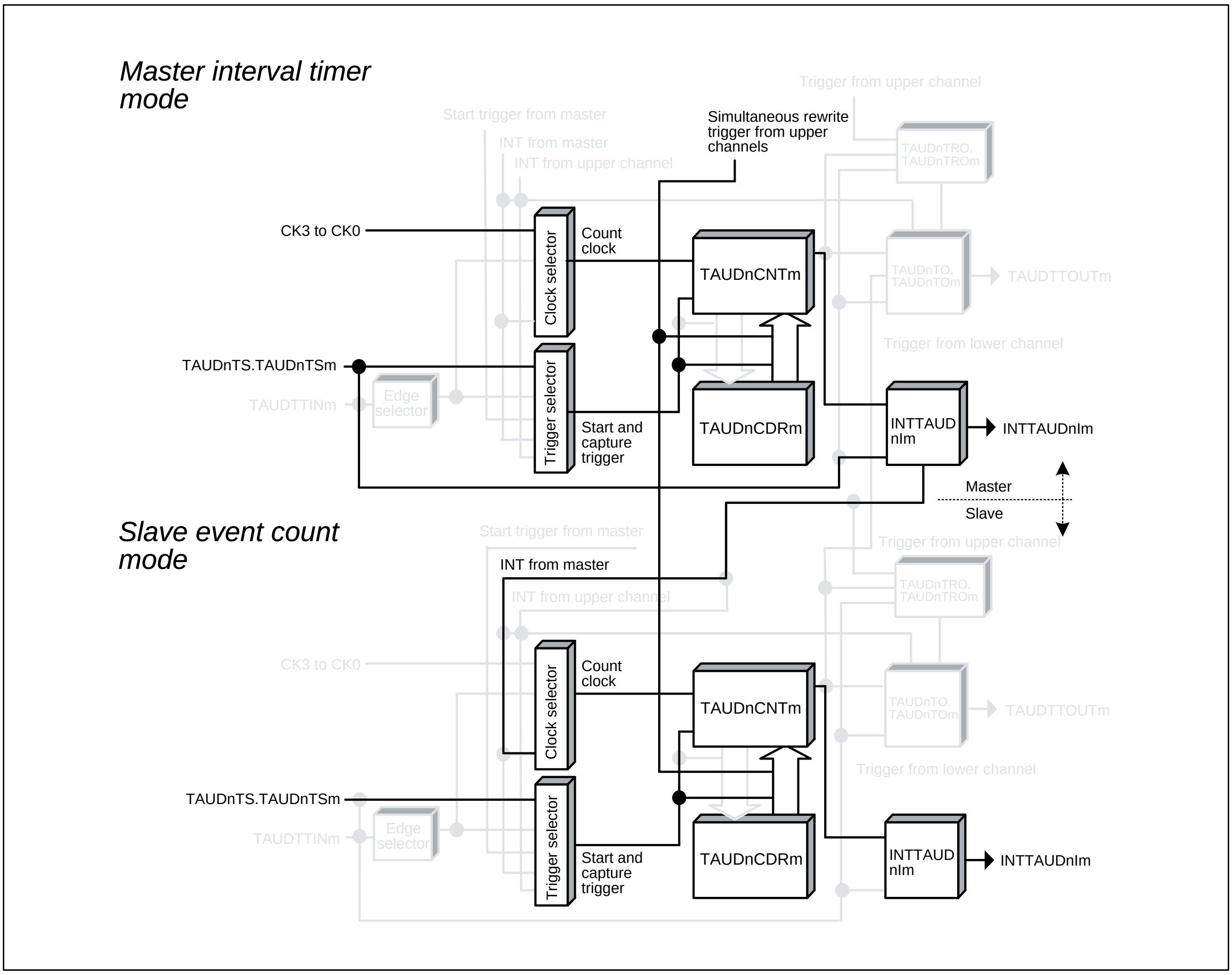


Figure 32.126 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

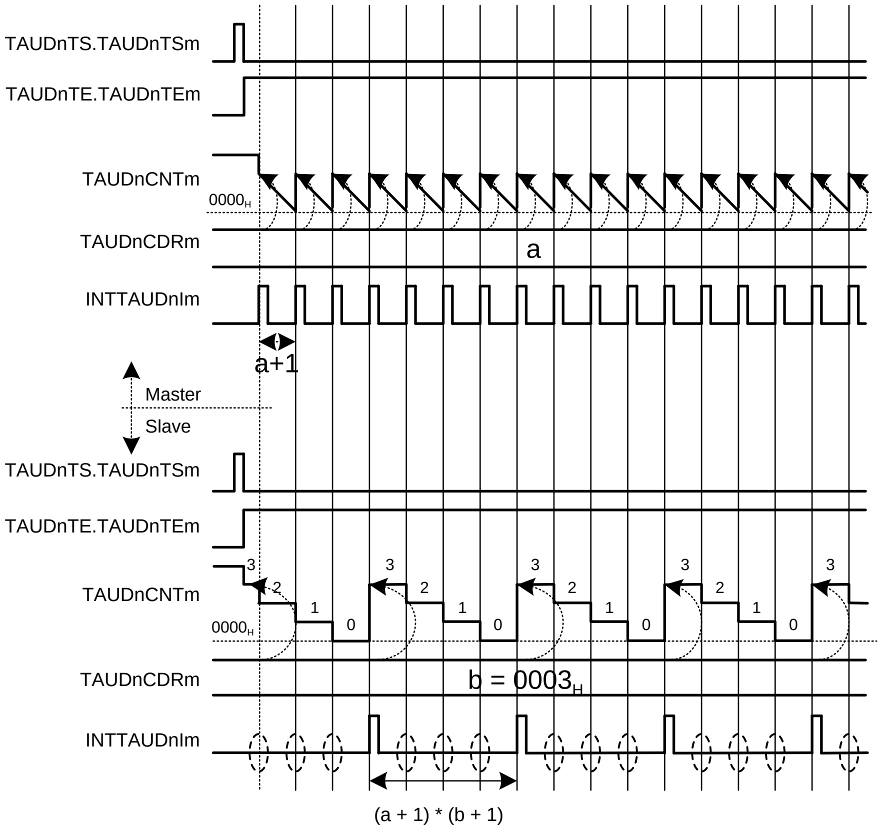


Figure 32.127 General Timing Diagram of Interrupt Request Signals Culling Function

32.15.10.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.216 Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.217 Contents of the TAUDnCMURm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.218 Simultaneous Rewrite Settings for the Master Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.10.5 Register Settings for the Slave Channel

(1) TAUDnCMORm for the Slave Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.219 Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for the Slave Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.220 Contents of the TAUDnCMURm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Slave Channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Slave Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.221 Simultaneous Rewrite Settings for the Slave Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

32.15.10.6 Operating Procedure for Interrupt Request Signals Culling Function

Table 32.222 Operating Procedure for Interrupt Request Signals Culling Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.10.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.15.10.5, Register Settings for the Slave Channel**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation. • TAUDnCNTm of slave channels counts down each time INTTAUDnIm of master channel is detected. When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • The TAUDnCDRm value is loaded in TAUDnCNTm (slave) and count operation continues.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

32.15.10.7 Specific Timing Diagrams

(1) Interrupt count (master) = interrupt count (slave)

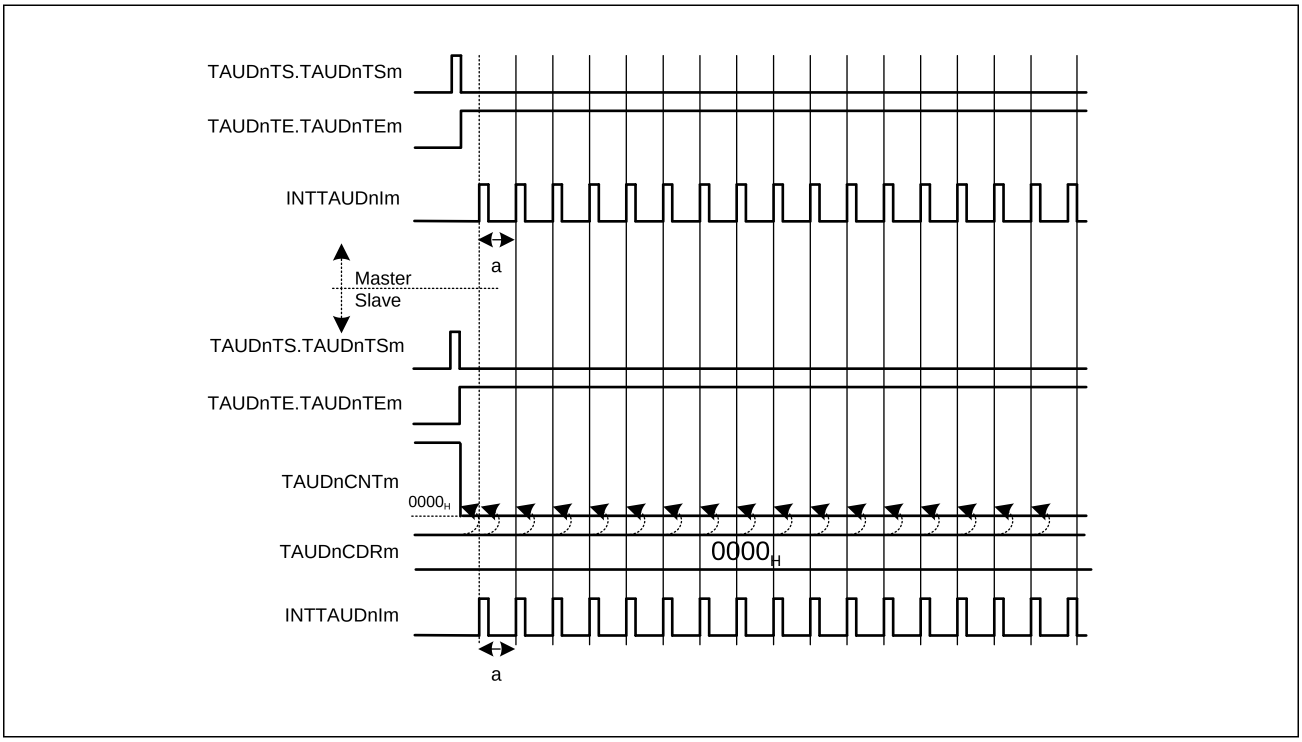


Figure 32.128 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm = 0000_H, the TAUDnCDRm value of the slave channel is loaded into TAUDnCNTm each time INTTAUDnIm of master channel is detected. In other words, TAUDnCNTm is always 0000_H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

32.16 Synchronous Non-Complementary and Complementary Modulation Output Functions

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and seven slave channels.

32.16.1 Non-Complementary Modulation Output Function Type 1

32.16.1.1 Overview

Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUDTTOUTm depending on the values of the real-time output bits (TAUDnTRO.TAUDnTROm) and the modulation output enable bits (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 32.224, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1**).
- The operating mode for slave channels 1 to 7 should be set to one-count mode (See **Table 32.227, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1**, and **Table 32.230, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1**).
- TAUDTTOUTm is not used with the master channel of this function.
- TAUDTTOUTm of slave channel 1 is not used with this function, but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 32.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output (See **Section 32.7, Channel Output Modes**).
- TAUDnCDRm of slave channel 1 should be set to 0000_H.

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.

- Slave channel 1:
  Slave channel 1 is set as a channel that triggers real-time output (TAUDnTRC.TAUDnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDnCDRm is fixed to 0000_H), the value of real-time output bit (TAUDnTRO.TAUDnTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to the TAUDTTOUTm output. After that, the counter returns to $FFFF_H$ and waits for the next interrupt of master channel.
- Slave channel 2:
  Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in **Table 32.223, TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**, a signal output from TAUDTTOUTm depends on the value of the real-time output bit (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 32.223 TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.
- TAUDnCDRm value of slave channel 1 should be set to 0000_H so that a real-time output is triggered at the same time with PWM generation on slave channels 2 to 7.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

32.16.1.2 Equations

Slave channels 2 to 7:

Pulse period = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

32.16.1.3 Block Diagram and General Timing Diagram

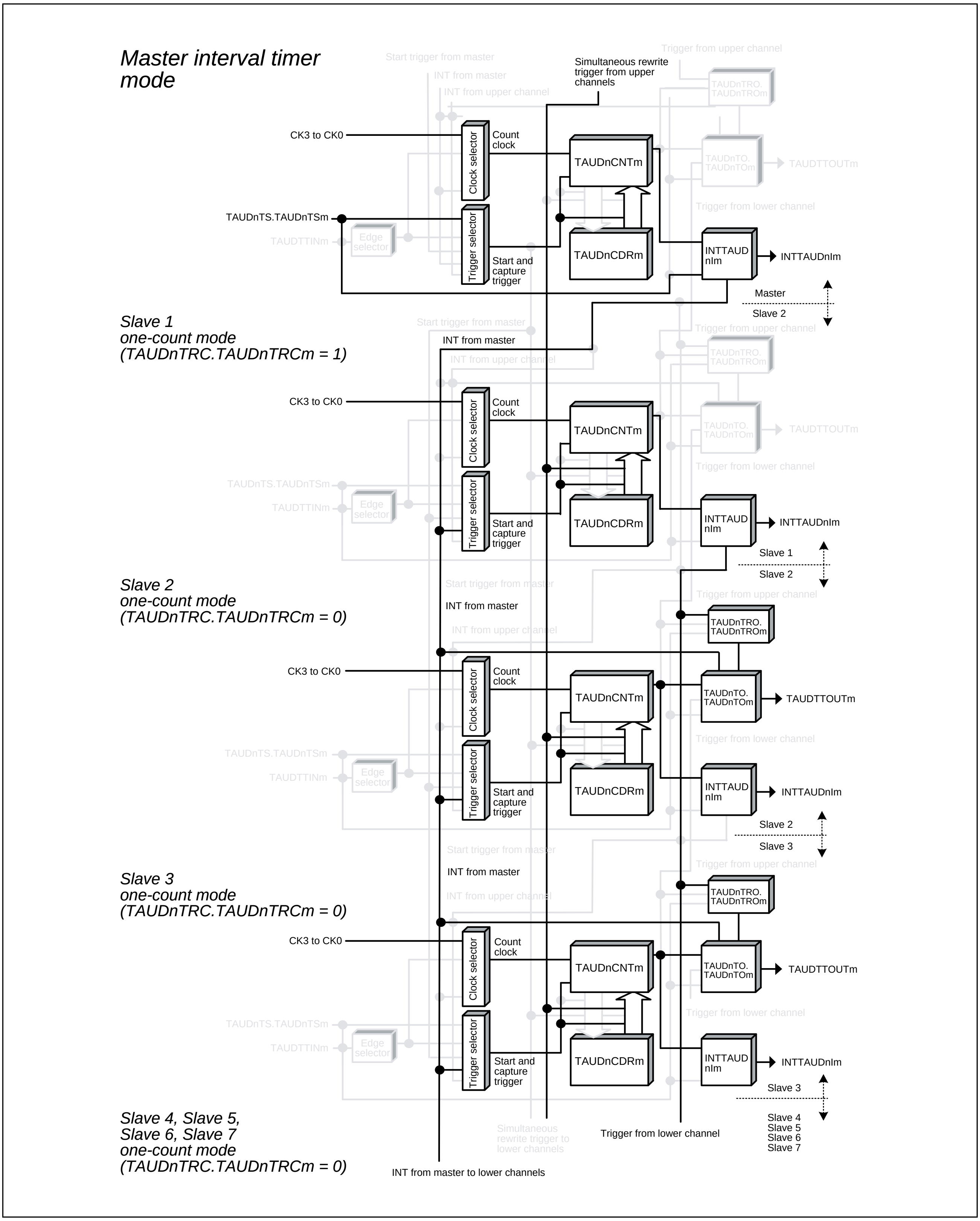


Figure 32.129 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

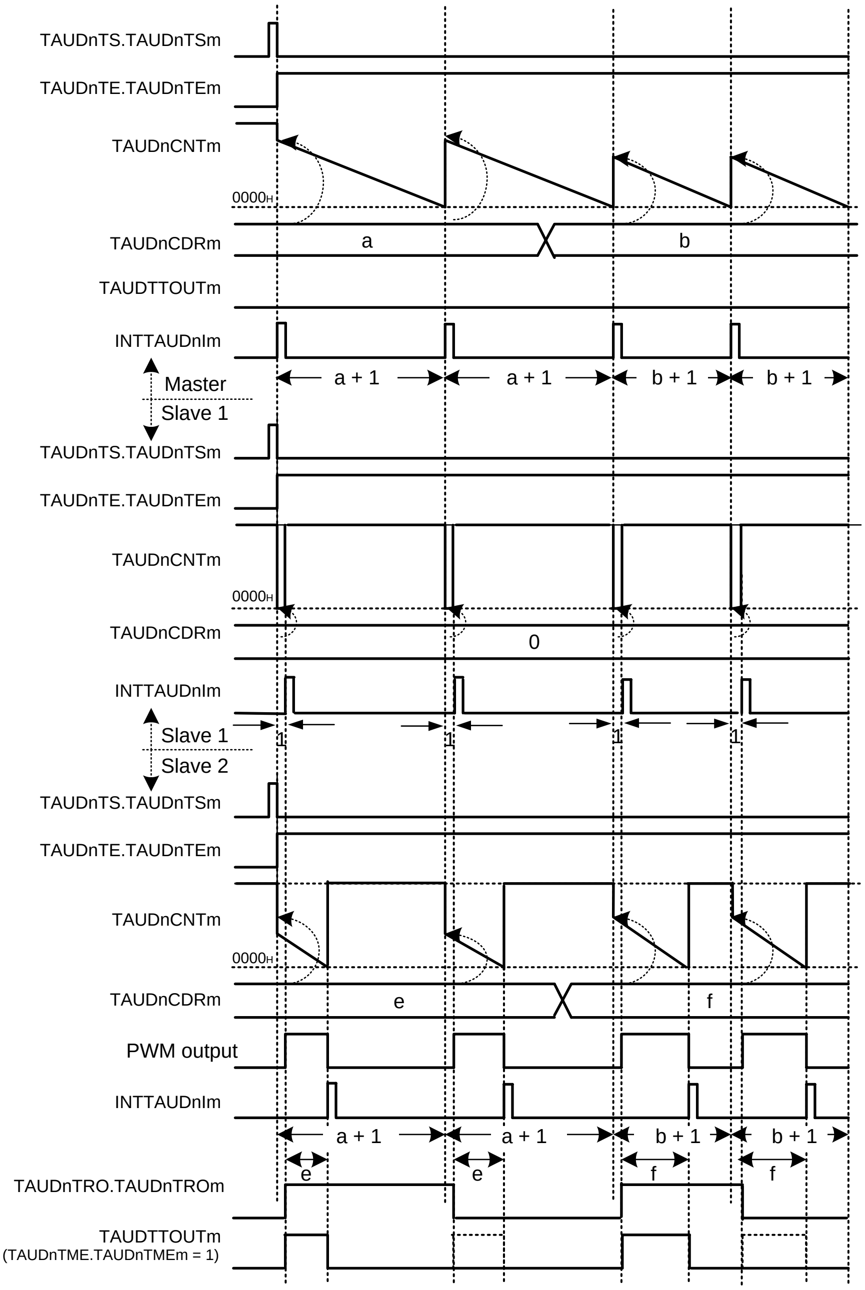


Figure 32.130 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

NOTE

TAUDTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

32.16.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.224 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.225 Contents of the TAUDnCMURm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.226 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires an upper channel higher than the master channel that operates with **Section 32.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions.

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  In addition, TAUDnCDRm settings for this channel are as follows.
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × Interrupt count) – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

32.16.1.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.227 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.228 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.229 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.1.6 Register Settings for Slave Channels 2 to 7

(1) TAUDnCMORm for Slave Channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.230 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.231 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels 2 to 7

Table 32.232 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous Rewrite of Slave Channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.233 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 32.234 Operating Procedure for Non-Complementary Modulation Output Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.1.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.1.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.1.6, Register Settings for Slave Channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle with TAUDnCDRm of master channel, 0000_H in TAUDnCDRm of slave channel 1, and duty width with TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 32.234 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Restart Operation

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: ● INTTAUDnIm is generated. ● TAUDnCDRm value of master channel is reloaded into TAUDnCNTm to continue counting down. ● PWM output signals of slave channels 2 to 7 are set. ● TAUDnCDRm value of slave channel 1 is reloaded into TAUDnCNTm to perform counting down. ● TAUDnCDRm value of slave channels 2 to 7 is reloaded into TAUDnCNTm to perform counting down. ● When the counter of slave channel 1 reaches 0000_H: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. ● When the counter of slave channels 2 to 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set. TAUDTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of a pair of slave channels.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.16.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

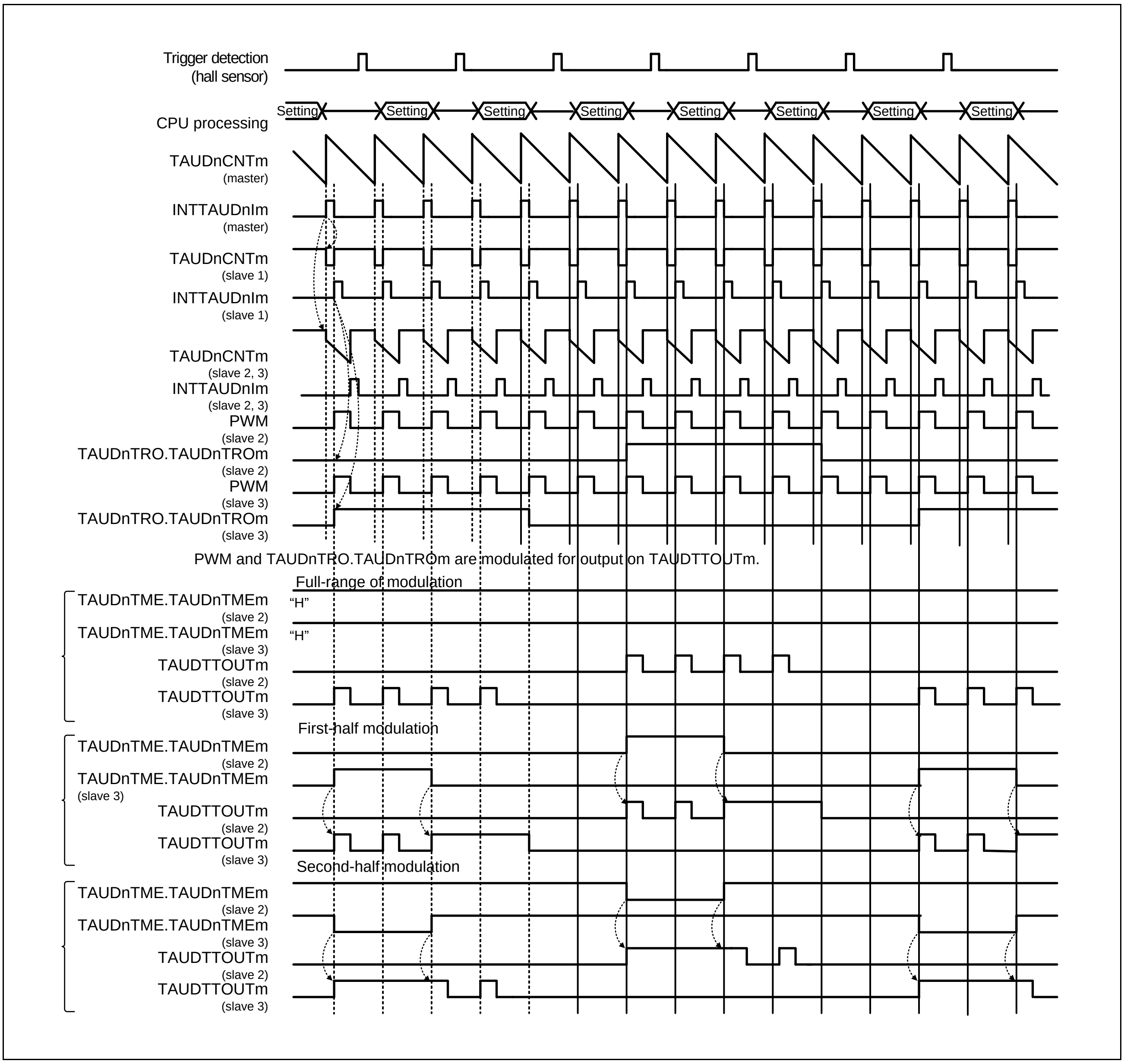


Figure 32.131 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUDTTOUTm.

A TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

32.16.2 Non-Complementary Modulation Output Function Type 2

32.16.2.1 Overview

Summary

This function outputs a triangular PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output enable bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 32.236, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 32.240, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channels 2 to 7 should be set to count-up/-down mode (See **Table 32.243, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2**).
- The output mode for the master channel should be set to independent channel output mode 1. (See **Section 32.7, Channel Output Modes**.)
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 32.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output (See **Section 32.7, Channel Output Modes**).

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm).

- Master channel:
  The counter of master channel starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.
- Slave channel 1:
  When slave channel 1 detects an interrupt from the master channel, the TAUDnCNTm value is decremented. When an interrupt from the master channel is detected (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, the TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.
  Since slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1), if an interrupt occurs on slave channel 1, the real-time output bit (TAUDnTRO.TAUDnTROm) of the channel which monitors an interrupt on the corresponding channel is reflected to the TAUDTTOUTm output.
- Slave channel 2:
  Once detecting an interrupt from the master channel, TAUDnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDnCDRm value is reloaded and then the counter starts to count down.
  If TAUDnCNTm = 0001_H, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operate like slave channel 2.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm) and a modulation output bit value (TAUDnTME.TAUDnTMEm) of the slave channel, as described in **Table 32.235, TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**.

This function cannot make a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel’s TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 32.235 TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

32.16.2.2 Equations

Slave channels 2 to 7:

Carrier cycle (down/up) = [TAUDnCDRm (master) + 1] × 2 × count clock cycle

Duty time = [TAUDnCDRm (master) + 1 - TAUDnCDRm (slave)] × 2 × count clock cycle

32.16.2.3 Block Diagram and General Timing Diagram

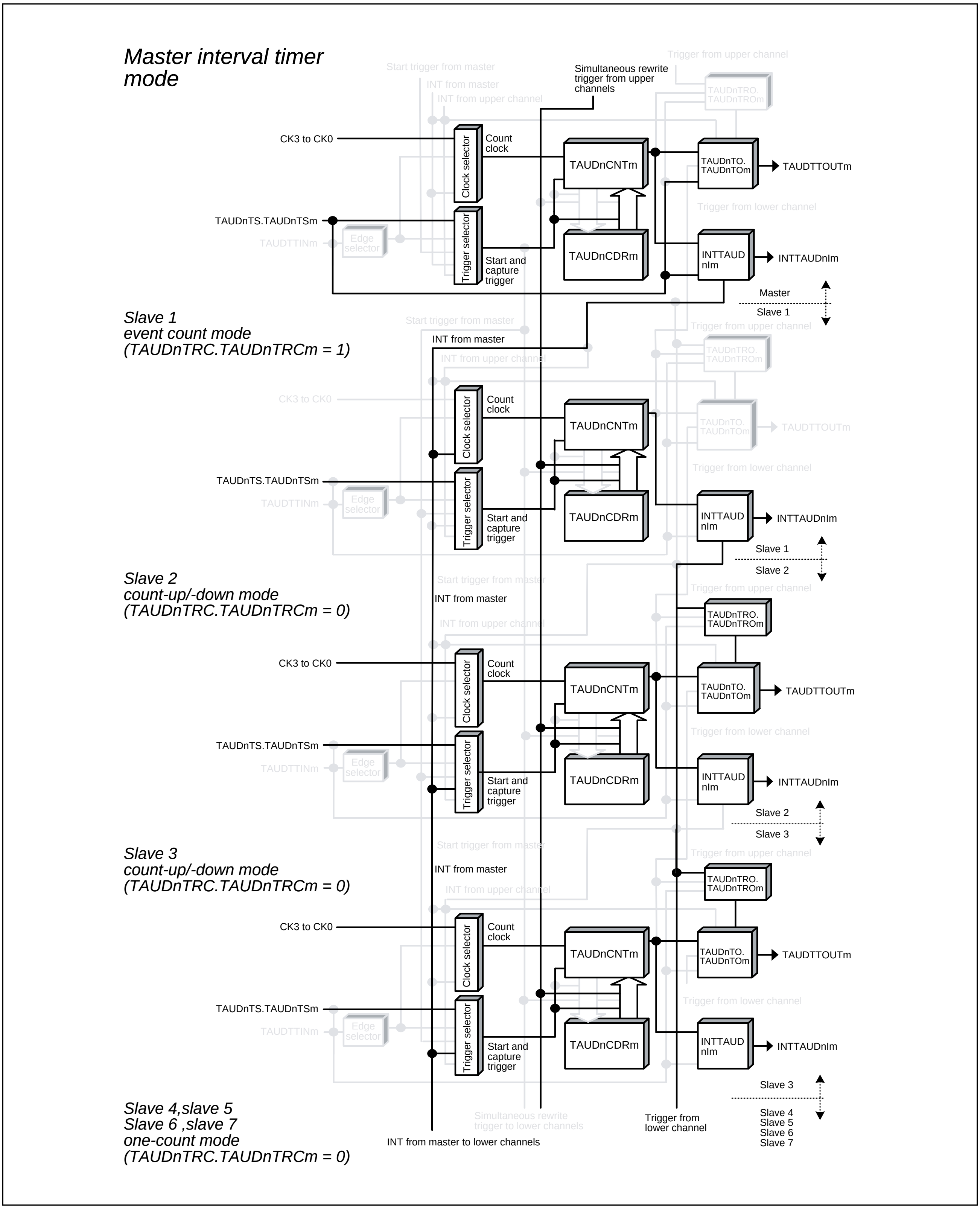


Figure 32.132 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

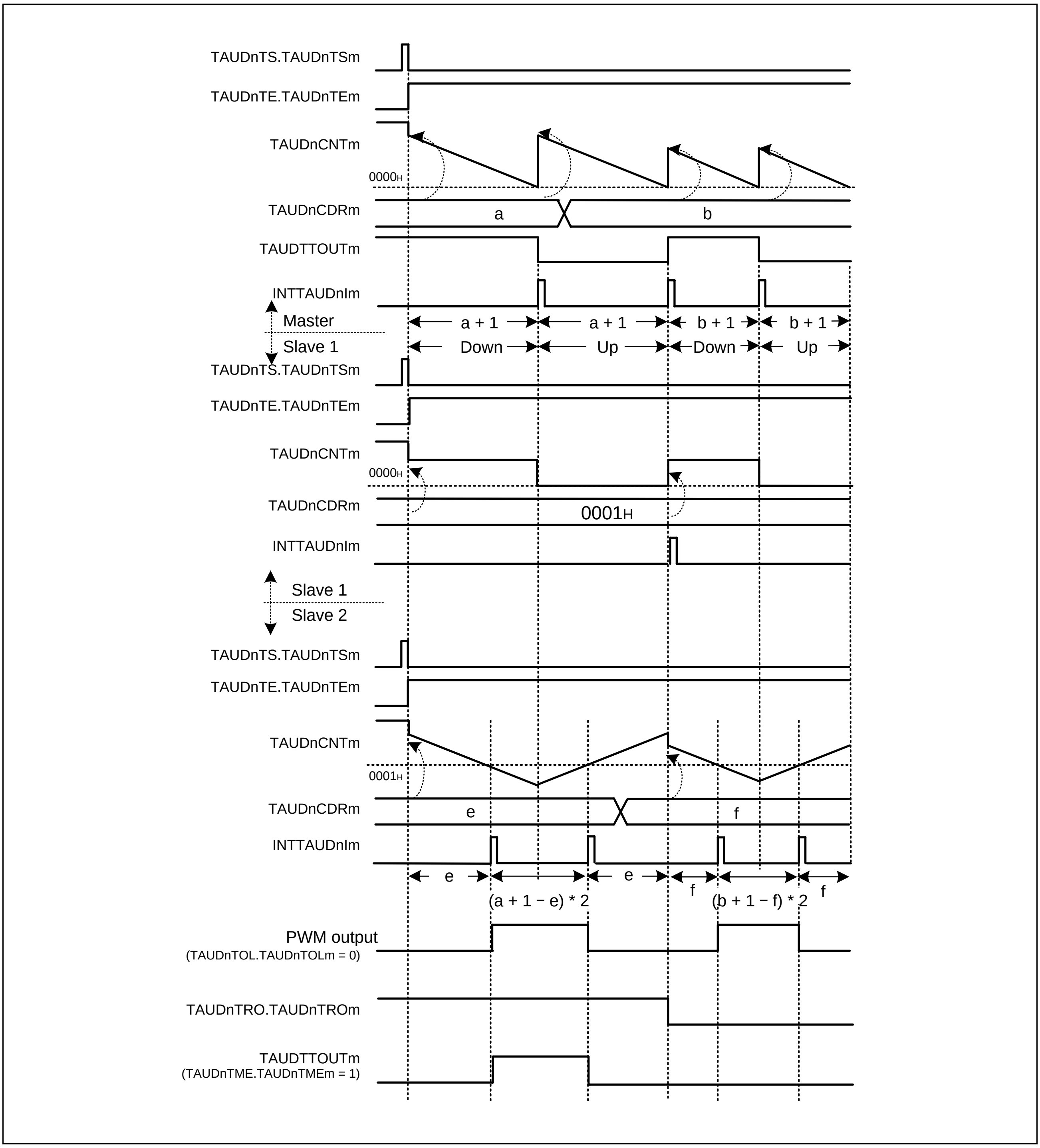


Figure 32.133 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

32.16.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.236 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.237 Contents of the TAUDnCMURm Register for the Master channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 32.238 Control Bit Settings for the Master Channel in Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (toggle mode with TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.239 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

32.16.2.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.240 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.241 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.242 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.2.6 Register settings for slave channels 2 to 7

(1) TAUDnCMORm for Slave Channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.243 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: The up/down output trigger signal of the master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.244 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output Mode for Slave Channels 2 to 7

Table 32.245 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: The upper channel generates the real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous Rewrite for Slave Channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.246 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 32.247 Operating Procedure for Non-Complementary Modulation Output Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.2.4, Register Settings for the Master** Channel. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.2.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.2.6, Register settings for slave channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set pulse cycle in TAUDnCDRm of master channel, and in TAUDnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates a real-time output trigger. Set duty width in TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 32.247 Operating Procedure for Non-Complementary Modulation Output Function Type 2

		Operation	TAUDn Status
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
	During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	The TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. The TAUDnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches 0000_H: ● INTTAUDnIm is generated. ● TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. ● The TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. ● TAUDnCNTm of slave channels 2 to 7 reloads the TAUDnCDRm value, or performs counting in opposite direction. ● At the same timing when the TAUDnCDRm value is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. ● When slave channel 1 detects an interrupt from the master channel for the (TAUDnCDRm + 1) times: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. ● When the counter of slave channels 2 to 7 reaches 0001_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set/reset.
	Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.16.2.8 Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

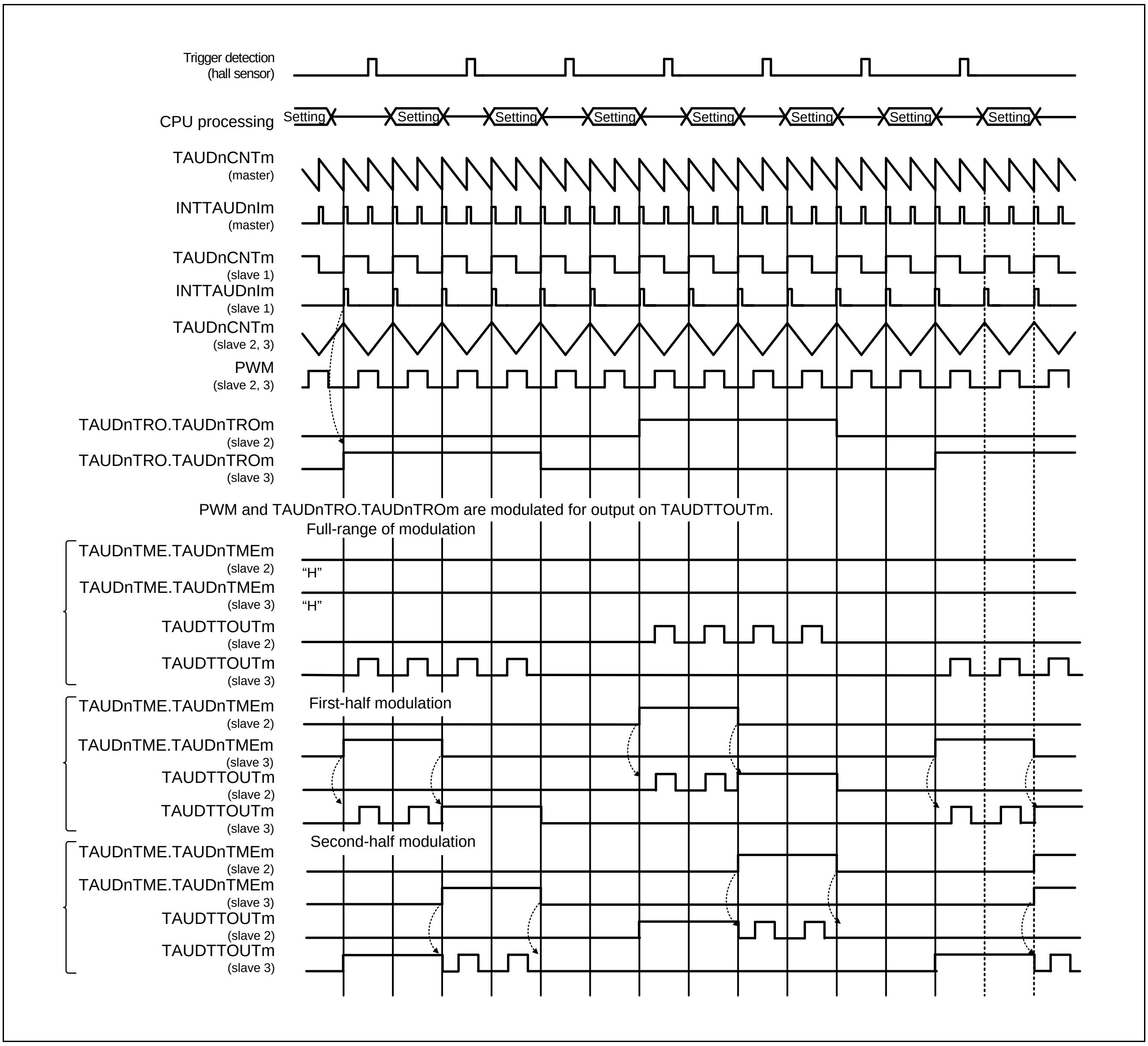


Figure 32.134 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

32.16.3 Complementary Modulation Output Function

32.16.3.1 Overview

Summary

This function outputs a triangle PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm with dead time added, depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels, and an output level bit value (TAUDnTDL.TAUDnTDLm). Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 32.249, Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 32.253, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See **Table 32.256, Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See **Table 32.260, Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function**).
  In addition, as the number of occurrences of an interrupt for slave channels 3, 5 and 7 within the carrier cycle is not uniquely determined, do not use the interrupt as an interrupt source.
- The output mode for master channels should be set to independent channel output mode 1 (See **Section 32.7, Channel Output Modes**).
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 32.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with complementary modulation output (See **Section 32.7, Channel Output Modes**).

Functional description

- Master channel:
  The counter of the master channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) of the master channel is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.
- Slave channel 1:
  When the counter reaches 0000_H, slave channel 1 waits for the next interrupt from the master channel. And the TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1) and INTTAUDnIm is generated.
  Slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1). The value of real-time output bit (TAUDnTRO.TAUDnTROm) of each channel is applied to the channel that detects the occurrence of an

interrupt on slave channel 1 by an interrupt. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.

- Slave channel 2:
  When the slave channel 2 counter reaches 0001_H, the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000_H, an interrupt occurs.
- Slave channels 2 and 3:
  The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.
- Slave channels 4 to 7:
  Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm), a modulation output bit value (TAUDnTME.TAUDnTMEm), and an output level bit value (TAUDnTDL.TAUDnTDLm) of the slave channel, as described in **Table 32.248, TAUDTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function**.

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit).

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 1 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm of one channel is set to 1, TAUDTTOUTm outputs the corresponding PWM of the channel.
  - If TAUDnTRO.TAUDnTROm of both channels is set to 0, TAUDTTOUTm of a pair outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 0 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm of the channel outputs a high-level signal.
  - If TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm of the channel outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic.

Table 32.248 TAUDTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

TAUDnTME.TAUDnTME02	TAUDnTME.TAUDnTME03	TAUDnTRO.TAUDnTRO02	TAUDnTRO.TAUDnTRO03	TAUDnTDL.TAUDnTDL02	TAUDnTDL.TAUDnTDL03	TAUDTTOUT2 Output	TAUDTTOUT3 Output
0	0	0	0	X	X	Low level	Low level
		0	1	1	0	Low level	High level
		1	0	0	1	High level	Low level
		1	1	X	X	Setting prohibited	Setting prohibited
1	1	0	0	X	X	Low level	Low level
		0	1	1	0	~PWM	PWM
		1	0	0	1	PWM	~PWM
		1	1	X	X	Setting prohibited	Setting prohibited

NOTES

1. In the above table, PWM indicates a positive PWM signal and ~PWM indicates an inverted PWM signal (positive logic). PWM and ~PWM are set by TAUDnTDL.TAUDnTDLm.
2. Any settings not listed above are prohibited.

- If TAUDnTME.TAUDnTMEm is continuously set to 1 while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the first half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the second half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, second-half modulation is applied.
- Whether dead time is added to a normal or reverse phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUDnTDL.TAUDnTDLm bit value.
  - If TAUDnTDL.TAUDnTDLm = 0, dead time is added to a normal phase PWM signal.
  - If TAUDnTDL.TAUDnTDLm = 1, dead time is added to a reverse phase PWM signal.
  - The operation defined by a TAUDnTDL.TAUDnTDLm bit value should be conducted by application software during operation. To modify TAUDnTDL.TAUDnTDLm, rewrite it during the period when TAUDnTRO.TAUDnTROm is 00_B.
- The TAUDnCDRm value of slave channel 1 should be set to the value to generate INTTAUDnIm of slave channel 1 at a carrier cycle (peak interrupt timing).
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- This function enables simultaneous rewrite. See **Section 32.6, Simultaneous Rewrite**.

32.16.3.2 Equations

Pulse period = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

For slave channels 2 and 3:

PWM signal width (positive phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2) × 2) – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2) × 2) + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

For slave channels 4 to 7:

Slave channels 4 and 6 are calculated in the same way as slave channel 2, whereas slave channels 5 and 7 are calculated as slave channel 3.

32.16.3.3 Block Diagram and General Timing Diagram

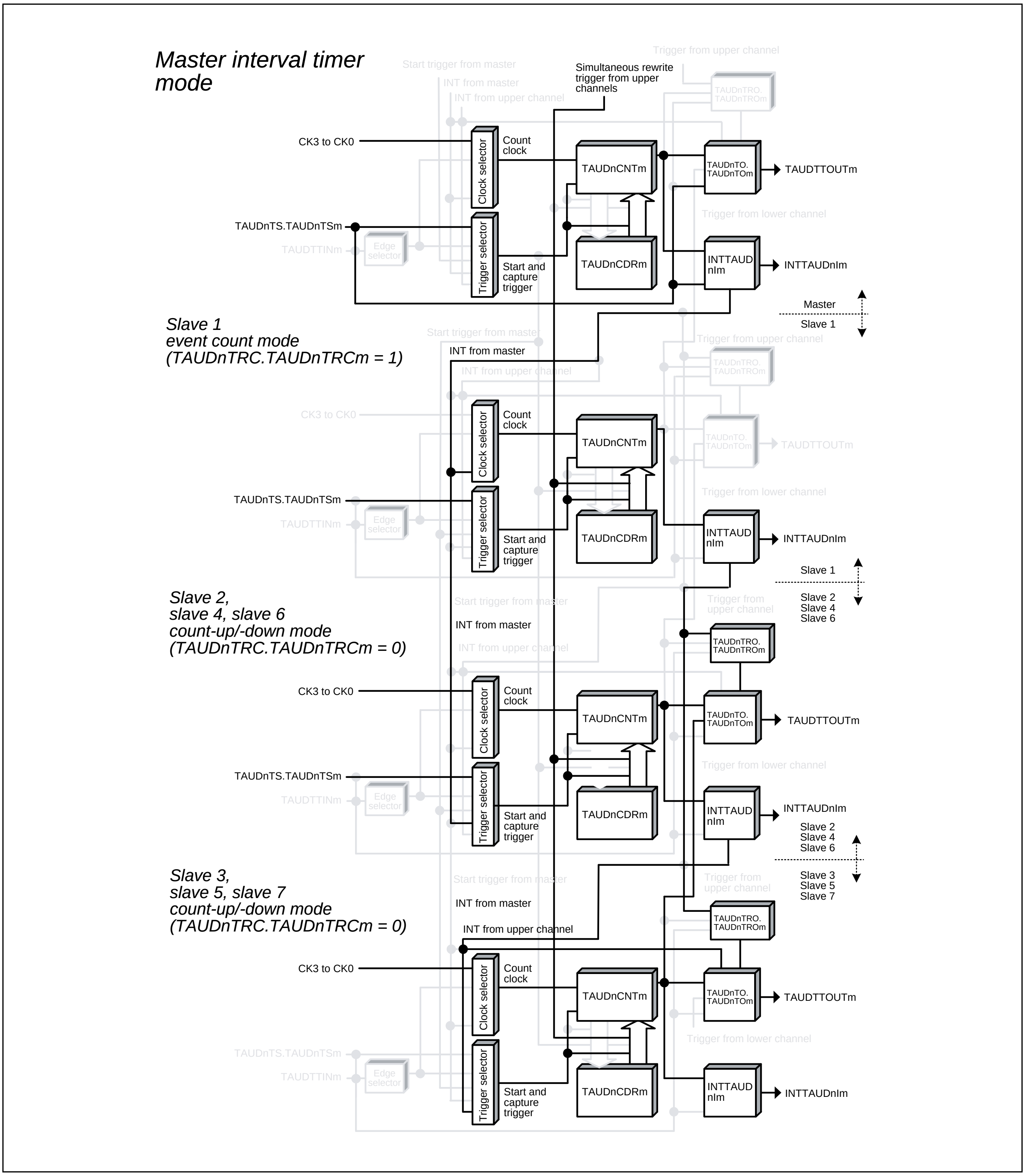


Figure 32.135 Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001_H
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

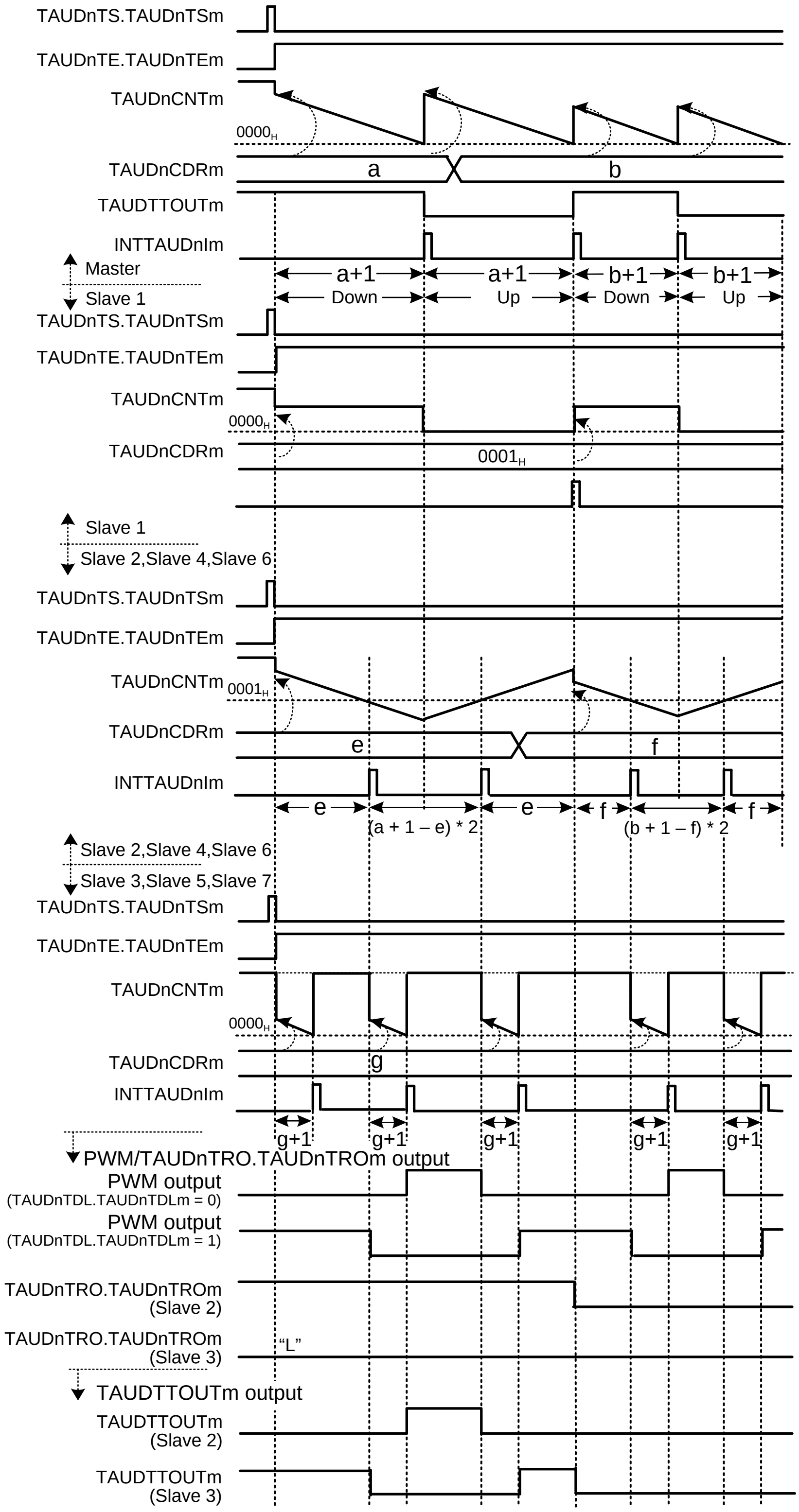


Figure 32.136 General Timing Diagram of Complementary Modulation Output Function

32.16.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.249 Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.250 Contents of the TAUDnCMURm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 32.251 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.252 Simultaneous Rewrite Settings for the Master Channel of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

32.16.3.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.253 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.254 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.255 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.3.6 Register settings for slave channels 2, 4, and 6

(1) TAUDnCMORm for Slave Channels 2, 4, and 6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.256 Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channels 2, 4, and 6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.257 Contents of the TAUDnCMURm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for Slave Channels 2, 4, and 6

Table 32.258 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous Rewrite for Slave Channels 2, 4, and 6

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.259 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.3.7 Register settings for slave channels 3, 5, and 7

(1) TAUDnCMORm for Slave Channels 3, 5, and 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 32.260 Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channels 3, 5, and 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.261 Contents of the TAUDnCMURm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output Mode for Slave Channels 3, 5, and 7

Table 32.262 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous Rewrite for Slave Channels 3, 5, and 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 32.263 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

32.16.3.8 Operating Procedure for Complementary Modulation Output Function

Table 32.264 Operating Procedure for Complementary Modulation Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.3.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.3.5, Register Settings for Slave Channel 1**. Slave channels 2, 4, and 6: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.3.6, Register settings for slave channels 2, 4, and 6**. Slave channels 3, 5, and 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 32.16.3.7, Register settings for slave channels 3, 5, and 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle using TAUDnCDRm of master channel, and an interrupt count of master channel to be ignored using TAUDnCDRm of slave channel 1. Also set a duty width in TAUDnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 32.264 Operating Procedure for Complementary Modulation Output Function

		Operation	TAUDn Status
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
	During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. TAUDnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000_H: ● INTTAUDnIm is generated. ● TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. ● TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for the next master channel interrupt. ● TAUDnCNTm of slave channels 2, 4, and 6 reloads the TAUDnCDRm value, or performs counting in opposite direction. ● At the same timing when the TAUDnCDRm value of slave channels 2, 4, and 6 is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. ● The counter of slave channel 1 waits for the next interrupt from the master channel when reaching 0000_H. When the interrupt is detected: – TAUDnCDRm value is reloaded into TAUDnCNTm and the counter waits for the next master channel interrupt. – INTTAUDnIm is generated. – TAUDnTRO.TAUDnTROm is changeable. ● When the counter of slave channels 2, 4, and 6 reaches 0001_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched). – TAUDnCDRm value of slave channels 3, 5, and 7 is loaded into TAUDnCNTm to perform counting down. ● When the counter of slave channels 3, 5, and 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched).
	Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

32.16.3.9 Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001_H
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

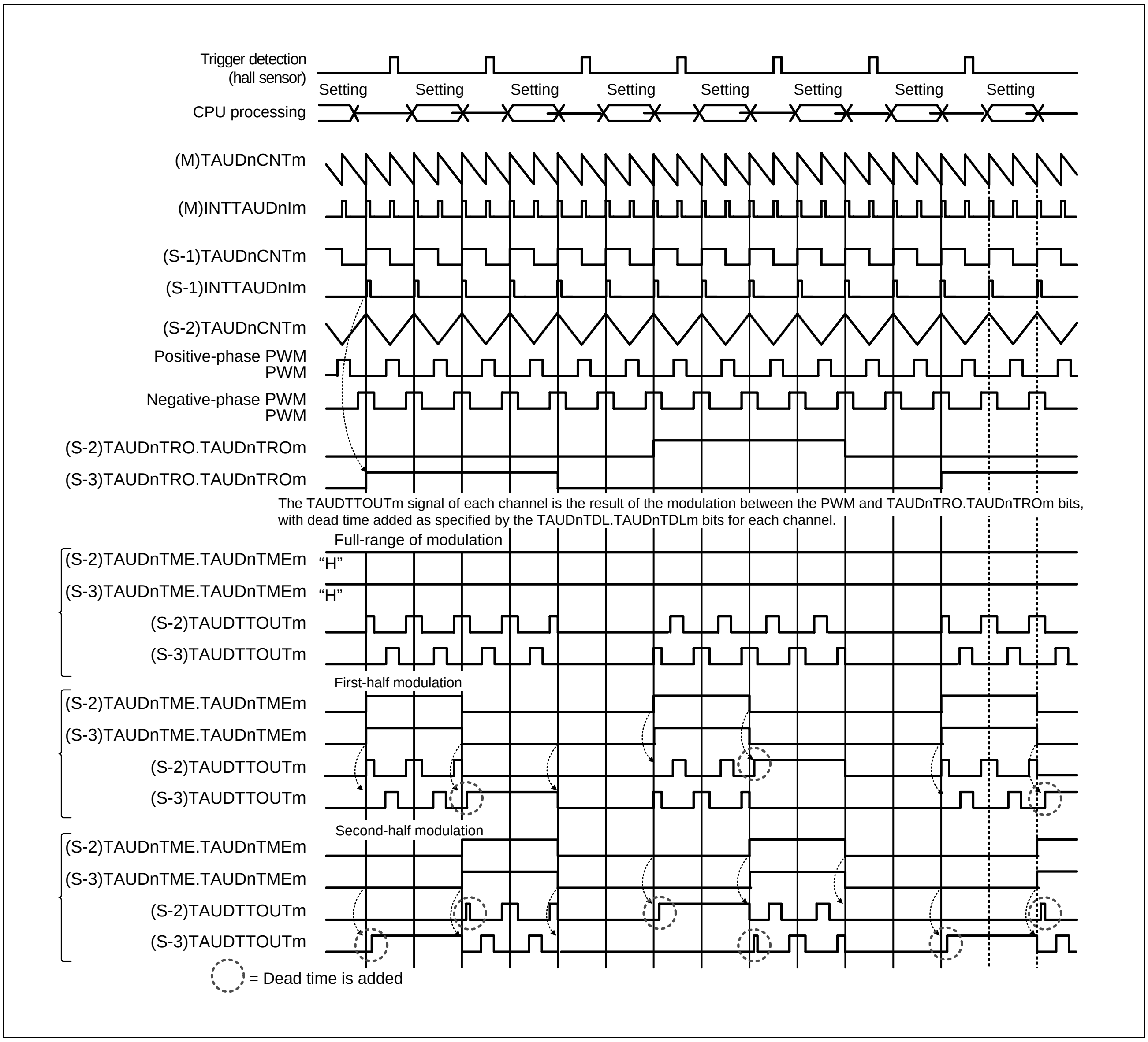


Figure 32.137 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDnTRO.TAUDnTROm bit value are output from slave channels 2 and 3.

TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm settings are reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

NOTE

Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, TAUDnTRO.TAUDnTROm, and TAUDnTDL.TAUDnTDLm can be changed.

Section 33 Timer Array Unit J (TAUJ)

This section contains a generic description of the timer array unit J (TAUJ).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of TAUJ.

33.1 Features of RH850/F1KH, RH850/F1KM TAUJ

33.1.1 Number of Units and Channels

This microcontroller has the following number of TAUJ units and channels.

Table 33.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	4	4	4
Name	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)

Table 33.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	4	4	4	4	4
Name	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)

Table 33.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	4	4	4	4
Name	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)	TAUJn (n = 0 to 3)

Table 33.4 TAUJn Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name TAUJn	Number of Channels per Unit	RH850/F1KH-D8 176 Pins (16 ch)	RH850/F1KH-D8 233 Pins (16 ch)	RH850/F1KH-D8 324 Pins (16 ch)
TAUJ0	4	✓	✓	✓
TAUJ1	4	✓	✓	✓
TAUJ2	4	✓	✓	✓
TAUJ3	4	✓	✓	✓

Table 33.5 TAUJn Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name TAUJn	Number of Channels per Unit	RH850/F1KM-S4 100 Pins (16 ch)	RH850/F1KM-S4 144 Pins (16 ch)	RH850/F1KM-S4 176 Pins (16 ch)	RH850/F1KM-S4 233 Pins (16 ch)	RH850/F1KM-S4 272 Pins (16 ch)
TAUJ0	4	✓	✓	✓	✓	✓
TAUJ1	4	✓	✓	✓	✓	✓
TAUJ2	4	✓	✓	✓	✓	✓
TAUJ3	4	✓	✓	✓	✓	✓

Table 33.6 TAUJn Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name TAUJn	Number of Channels per Unit	RH850/F1KM-S1 48 Pins (16 ch)	RH850/F1KM-S1 64 Pins (16 ch)	RH850/F1KM-S1 80 Pins (16 ch)	RH850/F1KM-S1 100 Pins (16 ch)
TAUJ0	4	✓	✓	✓	✓
TAUJ1	4	✓	✓	✓	✓
TAUJ2	4	✓	✓	✓	✓
TAUJ3	4	✓	✓	✓	✓

Table 33.7 Indices (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual TAUJ units are identified by the index “n”; for example, TAUJnTOM is the TAUJn channel output mode register.
m	The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index “m”, thus a certain channel is denoted as CHm (m = 0 to 3). The even numbered channels (m = 0, 2) are denoted as CHm_even. The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

33.1.2 Register Base Addresses

TAUJn base addresses are listed in the following table.

TAUJn register addresses are given as offsets from the base addresses.

Table 33.8 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<TAUJ0_base>	FFE5 0000_H
<TAUJ1_base>	FFE5 1000_H
<TAUJ2_base>	FFE5 0100_H
<TAUJ3_base>	FFE5 1100_H

33.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

Table 33.9 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
TAUJ0	PCLK	CKSCLK_ATAUJ	Timer count clock
	Register access clock	CPUCLK_L, CKSCLK_ATAUJ	Bus clock
TAUJ1	PCLK	CKSCLK_IPERI1	Timer count clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock
TAUJ2	PCLK	CKSCLK_ATAUJ	Timer count clock
	Register access clock	CPUCLK_L, CKSCLK_ATAUJ	Bus clock
TAUJ3	PCLK	CKSCLK_IPERI1	Timer count clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock

33.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

Table 33.10 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAUJ0			
INTTAUJ0I0	Channel 0 interrupt	80	21
INTTAUJ0I1	Channel 1 interrupt	81	80
INTTAUJ0I2	Channel 2 interrupt	82	81
INTTAUJ0I3	Channel 3 interrupt	83	22
TAUJ1			
INTTAUJ1I0	Channel 0 interrupt	168	46
INTTAUJ1I1	Channel 1 interrupt	169	100
INTTAUJ1I2	Channel 2 interrupt	170	47
INTTAUJ1I3	Channel 3 interrupt	171	101
TAUJ2			
INTTAUJ2I0	Channel 0 interrupt	277	103
INTTAUJ2I1	Channel 1 interrupt	278	104
INTTAUJ2I2	Channel 2 interrupt	279	105
INTTAUJ2I3	Channel 3 interrupt	280	106
TAUJ3			
INTTAUJ3I0	Channel 0 interrupt	281	107
INTTAUJ3I1	Channel 1 interrupt	282	108
INTTAUJ3I2	Channel 2 interrupt	283	109
INTTAUJ3I3	Channel 3 interrupt	284	110

33.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

Table 33.11 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
TAUJ0	All reset sources except the transition to DeepSTOP mode (AWORES)
TAUJ1	All reset sources (ISORES)
TAUJ2	All reset sources except the transition to DeepSTOP mode (AWORES)
TAUJ3	All reset sources (ISORES)

33.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 33.12 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
TAUJ0		
TAUJTTIN0, TAUJTTIN1	Channel 0, 1 input	TAUJ0I0, TAUJ0I1
TAUJTTIN2	Channel 2 input	TAUJ0I2 or RTCA0OUT*[1] or TAUJTTOUT0 (TAUJ1)*[1]
TAUJTTIN3	Channel 3 input	TAUJ0I3 or RTCA0OUT*[1] or TAUJTTOUT0 (TAUJ1)*[1]
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ0O0 to TAUJ0O3
TAUJ1		
TAUJTTIN0 to TAUJTTIN3	Channel 0 to 3 input	TAUJ1I0 to TAUJ1I3
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ1O0 to TAUJ1O3
TAUJ2		
TAUJTTIN0, TAUJTTIN1	Channel 0, 1 input	TAUJ2I0, TAUJ2I1
TAUJTTIN2	Channel 2 input	TAUJ2I2 or TAUJTTOUT0(TAUJ3)*[1]
TAUJTTIN3	Channel 3 input	TAUJ2I3 or TAUJTTOUT0(TAUJ3)*[1]
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ2O0 to TAUJ2O3
TAUJ3		
TAUJTTIN0 to TAUJTTIN3	Channel 0 to 3 input	TAUJ3I0 to TAUJ3I3
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ3O0 to TAUJ3O3

Note 1. For details, see **Section 33.1.8, TAUJ0 / TAUJ2 Input Selection**.

33.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

Table 33.13 Internal Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
TAUJnTSSTm*[1]	Simultaneous channel start trigger input	PIC

Note 1. n = 1 only. TAUJ0TSSTm is not connected to PIC.

33.1.8 TAUJ0 / TAUJ2 Input Selection

TAUJ0 Input Selection (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

The 1-Hz pulse output (RTCA0OUT) from RTCA0 and the output (TAUJTTOUT0) from TAUJ1 can be input to TAUJTTIN2 and TAUJTTIN3 as shown in the following figure.

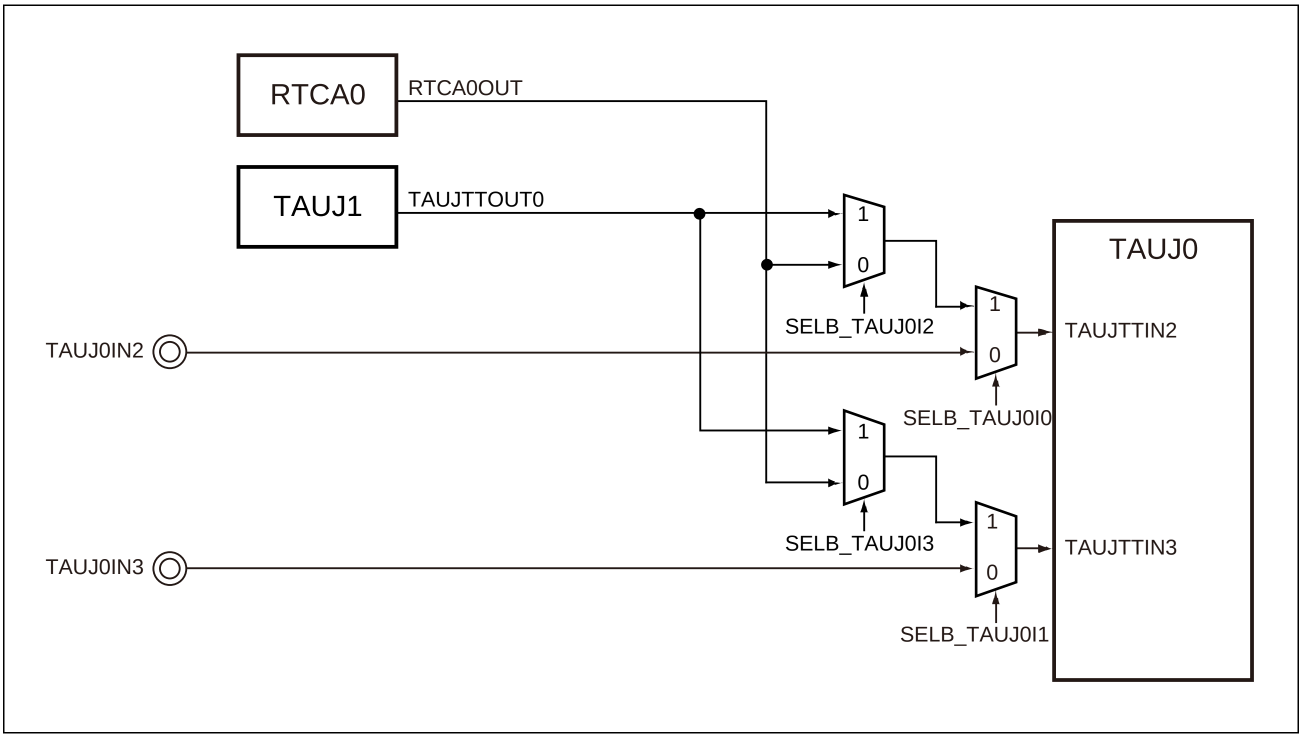


Figure 33.1 Selection of Signals Input to TAUJ0

TAUJ2 Input Selection (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

The output from TAUJ3 (TAUJTTOUT0) can be input to TAUJTTIN2 and TAUJTTIN3 as shown in the following figure.

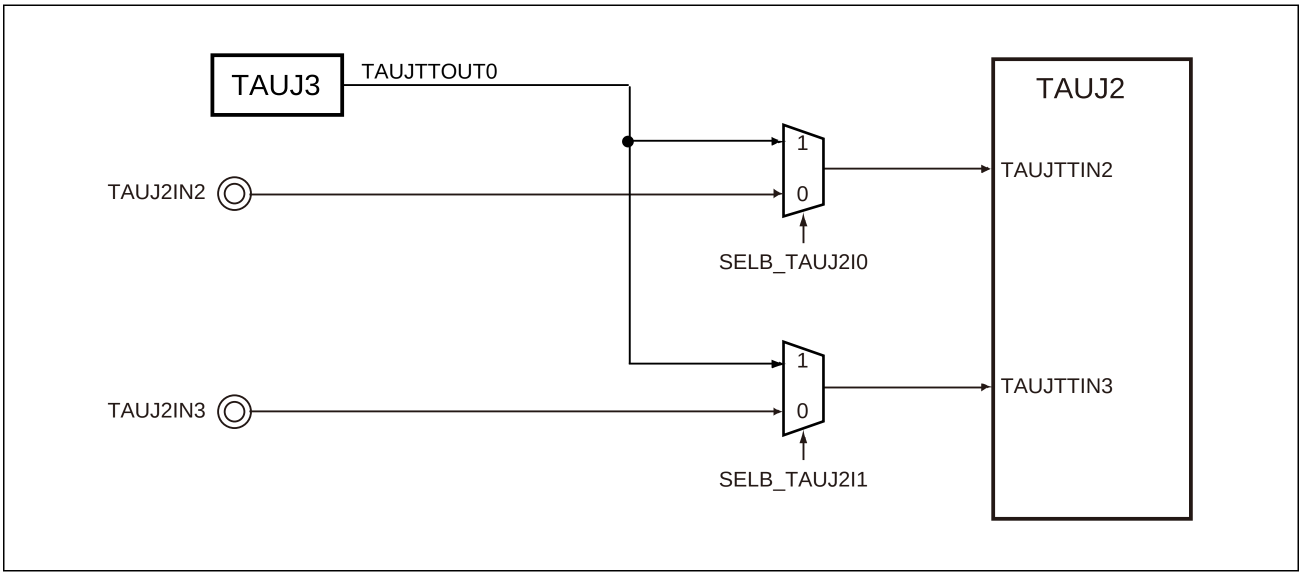


Figure 33.2 Selection of Signals Input to TAUJ2

The following table shows how to select signals input to the TAUJ.

Table 33.14 TAUJ0 Input Selections

Input Signal	Function	Settings			
		SELB_ TAUJ0I3	SELB_ TAUJ0I2	SELB_ TAUJ0I1	SELB_ TAUJ0I0
TAUJTTIN2	Port TAUJ0I2	—	—	—	0
	RTCA0OUT (Real-time clock 1-Hz output)	—	0	—	1
	TAUJTTOUT0 (TAUJ1)	—	1	—	1
TAUJTTIN3	Port TAUJ0I3	—	—	0	—
	RTCA0OUT (Real-time clock 1-Hz output)	0	—	1	—
	TAUJTTOUT0 (TAUJ1)	1	—	1	—

Table 33.15 TAUJ2 Input Selections

Input Signal	Function	Settings	
		SELB_ TAUJ2I1	SELB_ TAUJ2I0
TAUJTTIN2	Port TAUJ2I2	—	0
	TAUJTTOUT0 (TAUJ3)	—	1
TAUJTTIN3	Port TAUJ2I3	0	—
	TAUJTTOUT0 (TAUJ3)	1	—

33.1.8.1 List of Registers

Input signal selection register is listed in the following table.

Table 33.16 List of Registers

Module Name	Register Name	Symbol	Address
TAUJn input selection registers			
SL_TAUJn	TAUJTTINm input signal selection register	SELB_TAUJ0I	FFE5 4000$_H$
	TAUJTTINm input signal selection register	SELB_TAUJ2I	FFE5 4004$_H$

33.1.8.2 SELB_TAUJ0I — TAUJTTINm Input Signal Selection Register

This register selects the TAUJ0 input signals.

Access: This register can be read or written in 8-bit units.

Address: FFE5 4000$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SELB_TAUJ0I3	SELB_TAUJ0I2	SELB_TAUJ0I1	SELB_TAUJ0I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.17 SELB_TAUJ0I Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	SELB_TAUJ0I3	Selection of TAUJTTIN3 input signal: 0: RTCA0OUT 1: TAUJTTOUT0 (TAUJ1)
2	SELB_TAUJ0I2	Selection of TAUJTTIN2 input signal: 0: RTCA0OUT 1: TAUJTTOUT0 (TAUJ1)
1	SELB_TAUJ0I1	Selection of TAUJTTIN3 input signal: 0: Port TAUJ0I3 1: Timer Input
0	SELB_TAUJ0I0	Selection of TAUJTTIN2 input signal: 0: Port TAUJ0I2 1: Timer Input

33.1.8.3 SELB_TAUJ2I — TAUJTTINm Input Signal Selection Register

This register selects the input signals to several TAUJ2 inputs.

Access: This register can be read or written in 8-bit units.

Address: FFE5 4004_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SELB_TAUJ2I1	SELB_TAUJ2I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.18 SELB_TAUJ2I Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	SELB_TAUJ2I1	Selection of TAUJTTIN3 input signal: 0: Port TAUJ2I3 1: TAUJTTOUT0 (TAUJ3)
0	SELB_TAUJ2I0	Selection of TAUJTTIN2 input signal: 0: Port TAUJ2I2 1: TAUJTTOUT0 (TAUJ3)

33.2 Overview

33.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUJ is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

33.2.2 Terms

In this section, the following terms are used.

Independent channel operation function/synchronous operation channel operation function

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented by using a combination of channels.

- The independent channel operation function can be used any channel independently of all other channels.
- The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Upper/lower channel

Based on the channel number m, a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

33.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 33.19 Functional List of TAUJ Operations

Operation Function		Example
Independent Channel Operation Functions		**Section 33.12**
	Interval Timer Function	**Section 33.12.1**
	TAUJTTINm Input Interval Timer Function	**Section 33.12.2**
	TAUJTTINm Input Pulse Interval Measurement Function	**Section 33.12.3**
	TAUJTTINm Input Signal Width Measurement Function	**Section 33.12.4**
	TAUJTTINm Input Position Detection Function	**Section 33.12.5**
	TAUJTTINm Input Period Count Detection Function	**Section 33.12.6**
	Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)	**Section 33.12.7**
	Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)	**Section 33.12.8**
Synchronous Channel Operation Functions		**Section 33.13**
	PWM Output Function	**Section 33.13.1**

33.2.4 TAUJ I/O and Interrupt Request Signals

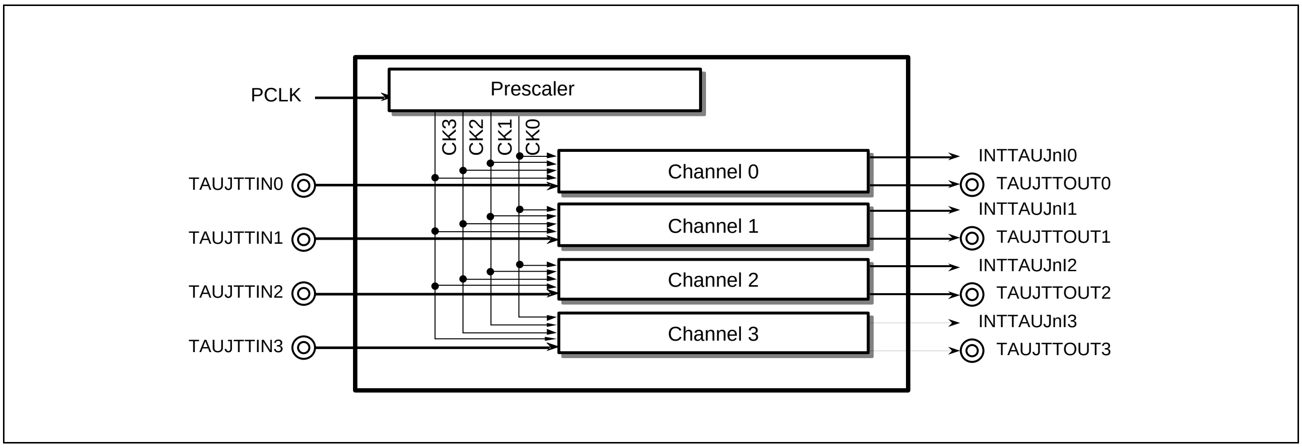


Figure 33.3 TAUJ I/O and Interrupt Request Signals

33.2.5 Block Diagram

The following figure shows the main components of the TAUJ.

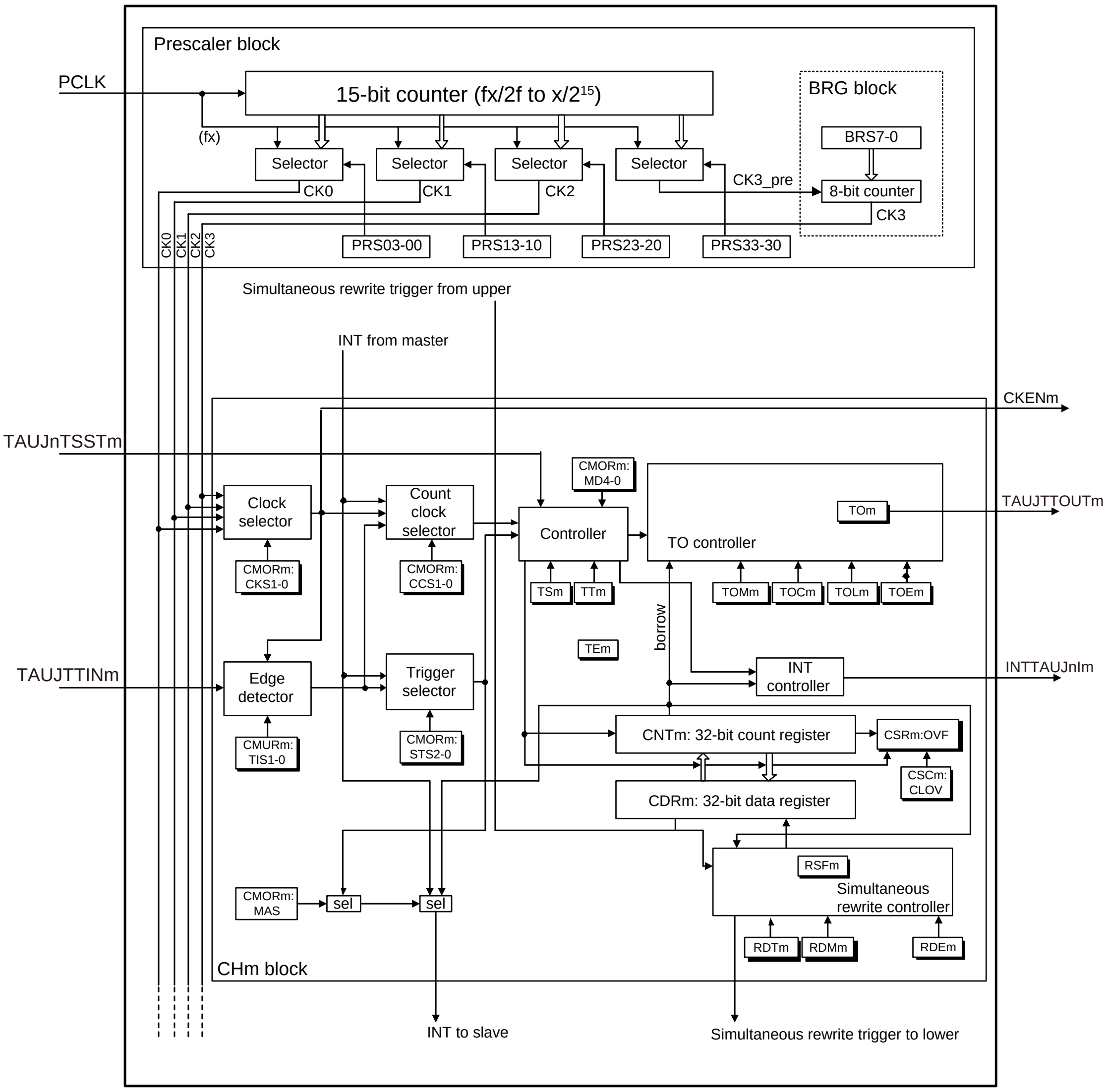


Figure 33.4 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

33.2.6 Description of Block Diagram

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived by dividing PCLK in the prescaler division factor of 2^0 to 2^{15}. The fourth count clock, CK3, is derived by dividing PCLK by a division factor that is not a power of 2 by using the baud rate generator.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of CK0 to CK3 clocks (selected by the clock selector)

Controller

The controller controls the main operations of the counter.

- Operating mode (selected with the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
- Valid edge of the TAUJTTINm input signal
- INTTAUJnIm from the master channel

Simultaneous rewrite controller

Simultaneous rewrite control is enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

33.3 Registers

33.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see **Section 33.1.2, Register Base Addresses**.

Table 33.20 List of Registers

Module Name	Register Name	Symbol	Address
TAUJn prescaler registers			
TAUJn	TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90_H
	TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94_H
TAUJn control registers			
TAUJn	TAUJn channel data register	TAUJnCDRm	<TAUJn_base> + m × 4_H
	TAUJn channel counter register	TAUJnCNTm	<TAUJn_base> + 10_H + m × 4_H
	TAUJn channel mode OS register	TAUJnCMORm	<TAUJn_base> + 80_H + m × 4_H
	TAUJn channel mode user register	TAUJnCMURm	<TAUJn_base> + 20_H + m × 4_H
	TAUJn channel status register	TAUJnCSRm	<TAUJn_base> + 30_H + m × 4_H
	TAUJn channel status clear trigger register	TAUJnCSCm	<TAUJn_base> + 40_H + m × 4_H
	TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54_H
	TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50_H
	TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58_H
TAUJn output registers			
TAUJn	TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60_H
	TAUJn channel output register	TAUJnTO	<TAUJn_base> + $5C_H$
	TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98_H
	TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + $9C_H$
	TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64_H
TAUJn reload data registers			
TAUJn	TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + $A0_H$
	TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + $A4_H$
	TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68_H
	TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + $6C_H$
TAUJn emulation register			
TAUJn	TAUJn emulation register	TAUJnEMU	<TAUJn_base> + $A8_H$

33.3.2 Details of TAUJn Prescaler Registers

33.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUJn_base> + 90_H

Value after reset: $FFFF_H$

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="4">TAUJnPRS3[3:0]</td><td colspan="4">TAUJnPRS2[3:0]</td><td colspan="4">TAUJnPRS1[3:0]</td><td colspan="4">TAUJnPRS0[3:0]</td></tr>
<tr><td>Value after reset</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>

Table 33.21 TAUJnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 12</td><td>TAUJnPRS3[3:0]</td><td>Specifies a CK3_PRE clock.
The CK3_PRE clock is an input clock of the BRG unit which supplies CK3 operation clocks to all channels.
<table>
<tr><th>TAUJnPRS3[3:0]</th><th>CK3_PRE Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

Table 33.21 TAUJnTPS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUJnPRS2[3:0]</td><td>Specifies a CK2 clock.
<table>
<tr><th>TAUJnPRS2[3:0]</th><th>CK2 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUJnPRS1[3:0]</td><td>Specifies a CK1 clock.
<table>
<tr><th>TAUJnPRS1[3:0]</th><th>CK1 Clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^{0}</td></tr>
<tr><td>0001_B</td><td>PCLK/2^{1}</td></tr>
<tr><td>0010_B</td><td>PCLK/2^{2}</td></tr>
<tr><td>0011_B</td><td>PCLK/2^{3}</td></tr>
<tr><td>0100_B</td><td>PCLK/2^{4}</td></tr>
<tr><td>0101_B</td><td>PCLK/2^{5}</td></tr>
<tr><td>0110_B</td><td>PCLK/2^{6}</td></tr>
<tr><td>0111_B</td><td>PCLK/2^{7}</td></tr>
<tr><td>1000_B</td><td>PCLK/2^{8}</td></tr>
<tr><td>1001_B</td><td>PCLK/2^{9}</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

Table 33.21 TAUJnTPS Register Contents

Bit Position	Bit Name	Function
3 to 0	TAUJnPRS0[3:0]	Specifies a CK0 clock.

TAUJnPRS0[3:0]	CK0 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 33.1.3, Clock Supply**.

33.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 94_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUJnBRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.22 TAUJnBRS Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 0</td><td>TAUJnBRS[7:0]</td><td>Specifies a CK3_PRE clock division factor for generating CK3.
<table>
<tr><th>TAUJnBRS[7:0]</th><th>CK3 Clock</th></tr>
<tr><td>$0000\ 0000_B$</td><td>CK3_PRE / 1</td></tr>
<tr><td>$0000\ 0001_B$</td><td>CK3_PRE / 2</td></tr>
<tr><td>$0000\ 0010_B$</td><td>CK3_PRE / 3</td></tr>
<tr><td>$0000\ 0011_B$</td><td>CK3_PRE / 4</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>$1111\ 1110_B$</td><td>CK3_PRE / 255</td></tr>
<tr><td>$1111\ 1111_B$</td><td>CK3_PRE / 256</td></tr>
</table>
</td></tr>
</table>

33.3.3 Details of TAUJn Control Registers

33.3.3.1 TAUJnCDRm — TAUJn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

Access: This register can be read or written in 32-bit units.

- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUJn_base> + 0_H + m × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.23 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR[31:0]	Data register for capture/compare values

33.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.24 TAUJnCNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT[31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 33.25 TAUJnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUJnCNTm		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF FFFF$_H$	Stop value	—
Capture mode	Count up	0000 0000$_H$	Stop value	—
One-count mode	Count down	FFFF FFFF$_H$	Stop value	FFFF FFFF$_H$
Capture and one-count mode	Count up	0000 0000$_H$	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count up	0000 0000$_H$	Stop value	—
Gate count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000 0000$_H$	Stop value	Stop value

Note 1. The value set for TAUJnCNTm when operating mode is changed after a reset is deasserted

33.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units.

Writable only when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 80_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.26 TAUJnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>TAUJnCKS[1:0]</td><td>Selects an operation clock, which is used with the TAUJTTINm input edge detection circuit.
Setting of TAUJnCMORm.TAUJnCCS[1:0] bits also allows the operation clock to serve as the TAUJnCNTm count clock.
<table>
<tr><th>TAUJnCKS1</th><th>TAUJnCKS0</th><th>Selection of Operation Clock</th></tr>
<tr><td>0</td><td>0</td><td>CK0</td></tr>
<tr><td>0</td><td>1</td><td>CK1</td></tr>
<tr><td>1</td><td>0</td><td>CK2</td></tr>
<tr><td>1</td><td>1</td><td>CK3</td></tr>
</table></td></tr>
<tr><td>13, 12</td><td>TAUJnCCS[1:0]</td><td>Selects a count clock for TAUJnCNTm counter.
<table>
<tr><th>TAUJnCCS1</th><th>TAUJnCCS0</th><th>Selection of Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>Operation clock specified by TAUJnCMORm.TAUJnCKS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>11</td><td>TAUJnMAS</td><td>Specifies whether the channel is a master or slave channel during synchronous channel operation.
0: Slave
1: Master
This bit setting is valid only for even channels (CHm_even). Odd channels (CHm-odd) are fixed to 0.</td></tr>
<tr><td>10 to 8</td><td>TAUJnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUJnSTS2</th><th>TAUJnSTS1</th><th>TAUJnSTS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUJnIm of master channel is used as a start trigger</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
</table>

Table 33.26 TAUJnCMORm Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TAUJnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m.
These bits are only valid if channel m is for capture function.
The bits must be fixed to 01_B in Capture mode and Capture and gate count mode.
<table>
<tr><th>TAUJn COS1</th><th>TAUJn COS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJTTINm input is detected.</td><td>Updated (cleared or set) when valid edge of TAUJTTINm input is detected.
● Set TAUJnCSRm. TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.
● Clear TAUJnCSRm. TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.
● Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.
● Occurrence of overflow: FFFF FFFF_H is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</td><td>No setting</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 33.26 TAUJnCMORm Register Contents

Bit Position	Bit Name	Function
4 to 0	TAUJnMD[4:0]	Specifies an operating mode.

TAUJn MD4	TAUJn MD3	TAUJn MD2	TAUJn MD1	TAUJn MD0	Functional Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Setting prohibited
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Setting prohibited
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Setting prohibited
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Setting prohibited
1	0	1	0	1/0	Setting prohibited
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUJnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUJnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not generated. 1: INTTAUJnIm is generated.
One-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION:** In one-count mode, INTTAUJnIm signal is not output at the beginning of count operation.
Capture and one-count mode Gate count mode Capture and gate count mode	This bit should be set to 0. **CAUTION:** INTTAUJnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.

33.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 20_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.27 TAUJnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	Specifies a valid edge of TAUJTTINm input signal.

TAUJnTIS1	TAUJnTIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

Edge detection of TAUJTTINm input signal is based on the operation clock selected by TAUJnCMORm.TAUJnCKS[1:0].

33.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnOVF
Value after reset	0	0	0	0	0	0	—	0
R/W	R	R	R	R	R	R	R	R

Table 33.28 TAUJnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	Reserved	When read, an undefined value is returned.
0	TAUJnOVF	Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: ● Capture mode ● Capture and one-count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

33.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 40_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 33.29 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

33.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 54_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 33.30 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: No function 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1. Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1. Whether the counter is started or not depends on the selected operating mode.

33.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 33.31 TAUJnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnTEm	Indicates whether channel m's counter operation is enabled. 0: Counter operation is disabled 1: Counter operation is enabled This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is reset to 0 when TAUJnTT.TAUJnTTm is set to 1.

33.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 33.32 TAUJnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTTm	Stops channel m’s counter operation. 0: No function 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm. TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJTTOUTm retain the values provided before the counter is stopped.

33.3.4 Details of TAUJn Simultaneous Rewrite Register

33.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOLm for the PWM output function.

Access: This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.33 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

33.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.34 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDMm	Specifies when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: No function These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

33.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 68_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 33.35 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnRDTm	Triggers the simultaneous rewrite enabling state. 0: No function 1: Simultaneous rewrite enabling state is triggered. The simultaneous rewrite enabling flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: • TAUJnRDE.TAUJnRDEm = 1

33.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + $6C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 33.36 TAUJnRSF Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUJnRDTm = 1).

33.3.5 Details of TAUJn Output Registers

33.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 60_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.37 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOEm	Enables/disables independent channel output function: 0: Disables independent timer output function (controlled by software) 1: Enables independent timer output function

33.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJTTOUTm.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + $5C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.38 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOm	Specifies and reads the level of TAUJTTOUTm: 0: Low 1: High Only TAUJnTOm bits for which Independent Channel Output function is disabled (TAUJnTOEm = 0) can be written.

33.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 98_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.39 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on the settings of channel output control (TAUJnTOE.TAUJnTOEm) bits.

33.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

Access: This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + $9C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.40 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOCm	Specifies the output mode: 0: Operation mode 1 (= Toggle mode) 1: No function This bit must be set to 0 for all output modes except independent channel output mode controlled by software. The output mode also depends on TAUJnTOM.TAUJnTOMm, as shown in the following table.

TAUJnTOMm	TAUJnTOCm	Functional Description
0	0	Toggle mode: Toggling proceeds when INTTAUJnIm occurs.
0	1	No function
1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.
1	1	No function

33.3.5.5 TAUJnTOL — TAUJn Channel Output Active Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.41 TAUJnTOL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOm): 0: Positive logic (active high) 1: Negative logic (active low) These bits apply in all channel output modes except independent channel output mode controlled by software and independent channel output mode 1.

33.3.6 TAUJn Emulation Register

33.3.6.1 TAUJnEMU — TAUJn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 8-bit units.

A write should be performed when counters are stopped (TAUJnTE.TAUJnTEm = 0) and (EPC.SVSTOP = 0).

Address: <TAUJn_base> + $A8_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUJn SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 33.42 TAUJnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUJnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

33.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After a reset is deasserted, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   - Set the operation mode
   - Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSm bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 33.12, Independent Channel Operation Functions** and **Section 33.13, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

33.5 Concepts of Synchronous Channel Operation Function

The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 33.5.1, Rules of Synchronous Channel Operation Function**.

The synchronous channel operation function are detailed in the following section.

- **Section 33.13, Synchronous Channel Operation Functions**

33.5.1 Rules of Synchronous Channel Operation Function

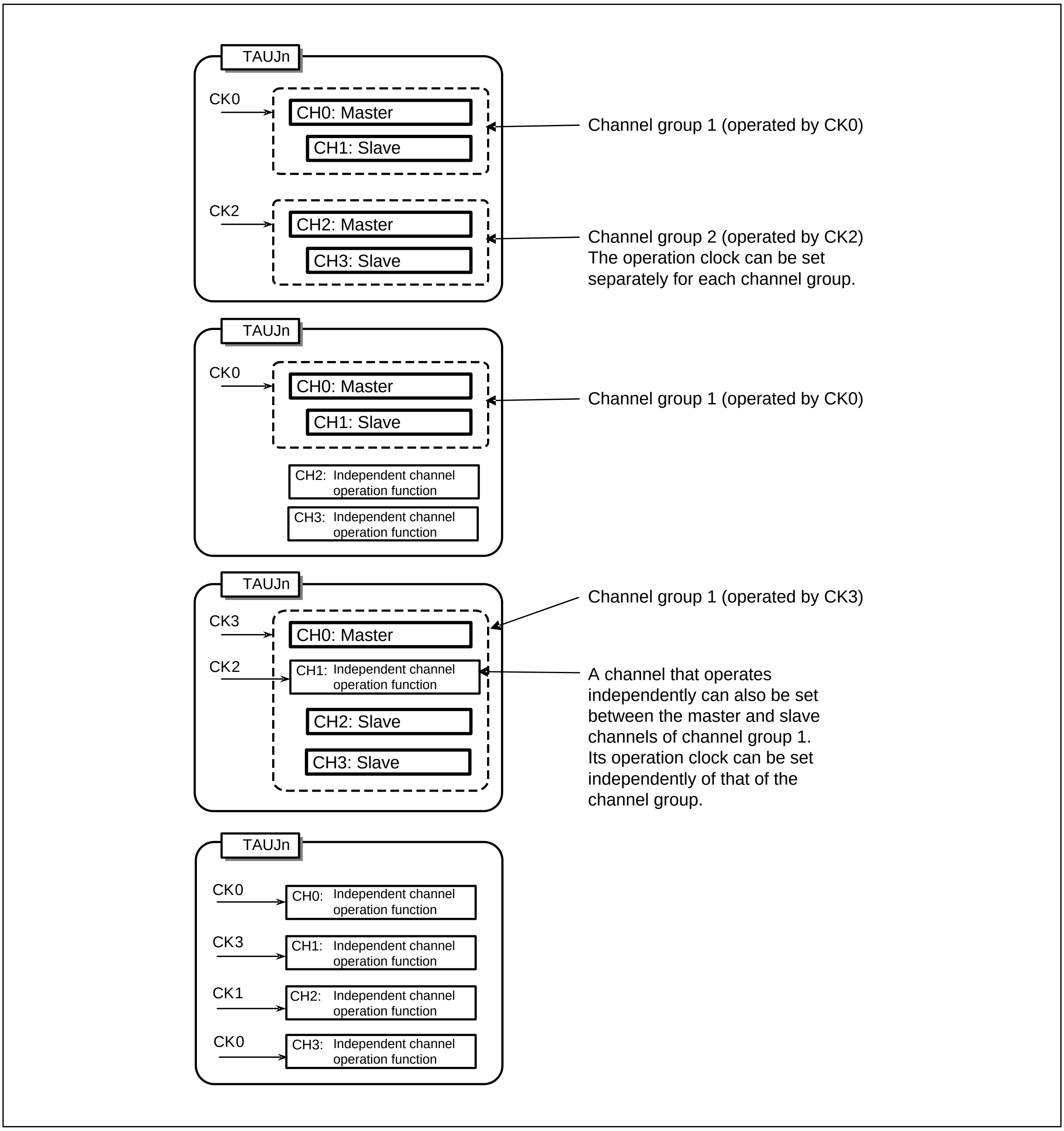
Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel other than CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
  Example: If CH0 and CH2 are master channels, CH1 can be set as slave channel for CH0, but CH3 cannot.

Operation clock

- The same operation clock should be set for the master channel and the synchronized slave channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 33.5, Grouping of Channels and Assignment of Operation Clocks**.



Channel group 1 (operated by CK0)

Channel group 2 (operated by CK2)
The operation clock can be set
separately for each channel group.

Channel group 1 (operated by CK0)

Channel group 1 (operated by CK3)

A channel that operates independently can also be set between the master and slave channels of channel group 1. Its operation clock can be set independently of that of the channel group.

Figure 33.5 Grouping of Channels and Assignment of Operation Clocks

33.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

33.5.2.1 Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEm bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

33.5.2.2 Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to perform simultaneous start between the units, see **Section 36.8, Simultaneous Start Trigger Function**.

33.6 Simultaneous Rewrite

33.6.1 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

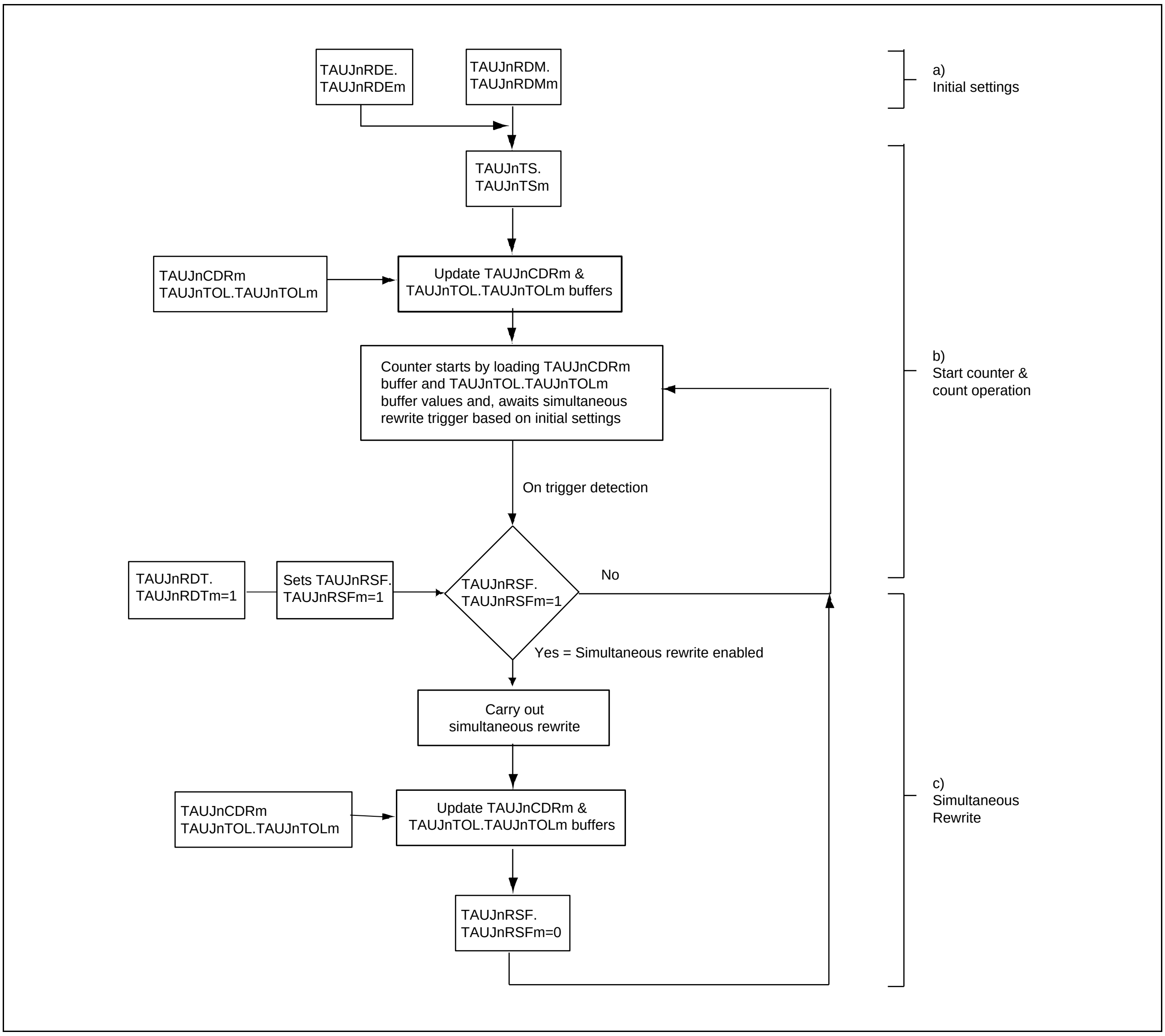


Figure 33.6 General Procedure for Simultaneous Rewrite

33.6.1.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1.
- To select simultaneous rewrite when the master channel starts counting, set TAUJnRDM.TAUJnRDMm.

33.6.1.2 Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. The values of TAUJnTOL.TAUJnTOLm and the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

33.6.1.3 Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is complete, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

33.6.2 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJTTOUTm outputs an invalid waveform.

33.6.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

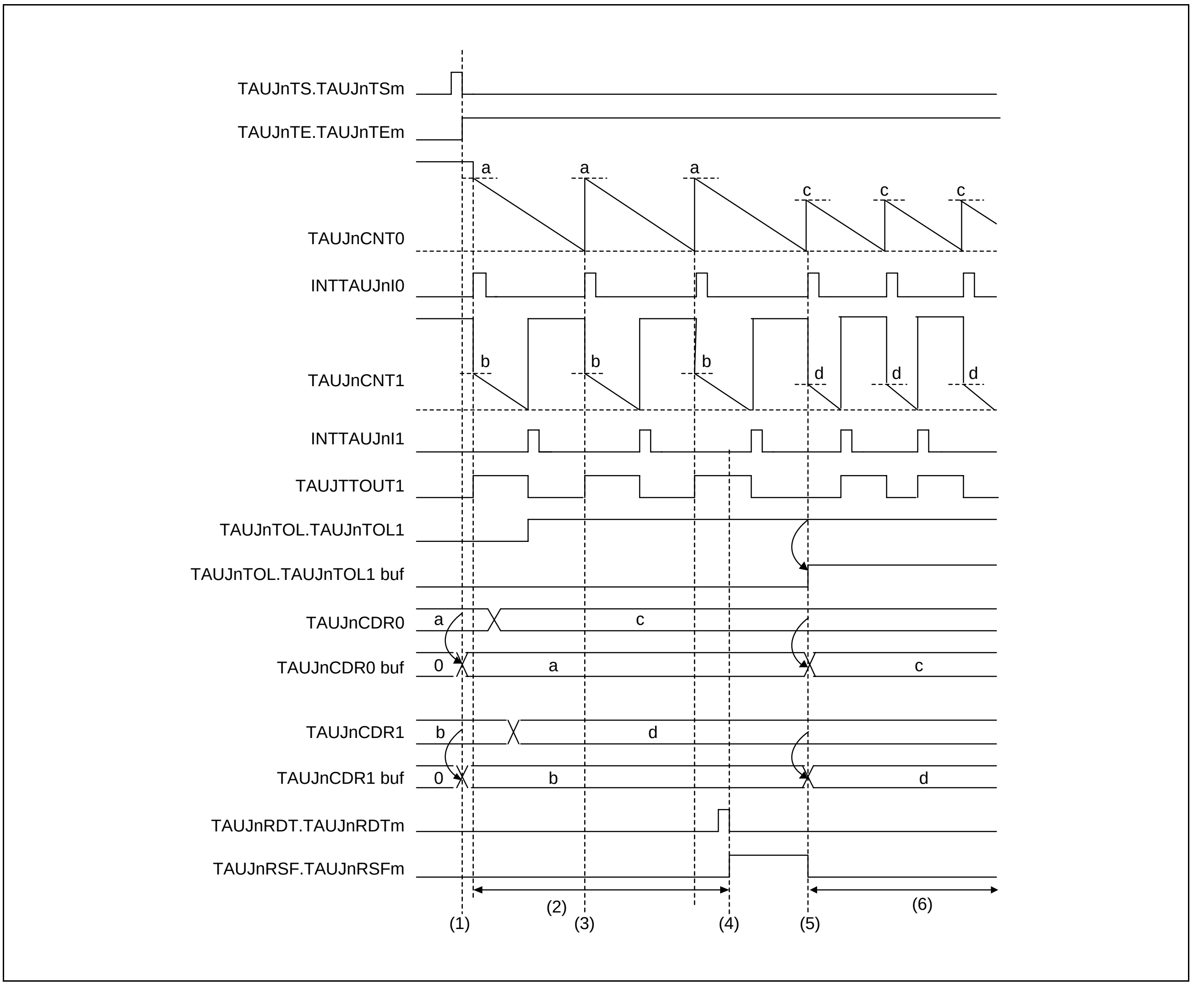


Figure 33.7 Simultaneous Rewrite with PWM Output Function

Setting:

CH0 is a master channel of PWM output function, and CH1 is a slave channel of PWM output function. Simultaneous rewrite is applied when the master channel starts counting.

Description:

(1) When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.

(2) The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).

(4) The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

33.7 Channel Output Modes

The output of the TAUJTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is sent to the output pin (TAUJTTOUTm).
- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)
  When controlled by TAUJ signals, the output level of TAUJTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJTTOUTm.
  - Independently (TAUJnTOM.TAUJnTOMm = 0)
    In case of independent operation, the output of the TAUJTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).
  - Synchronously (TAUJnTOM.TAUJnTOMm = 1)
    In case of synchronous operation, the output of the TAUJTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 33.43, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 33.7.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 33.7.3, Channel Output Modes Controlled Synchronously by TAUJn Signals**

Batch operation of TAUJnTOm bit

Whether a set value is reflected to the TAUJnTOm bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTOm setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted. No TAUJnTOm setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

NOTE

TAUJnTO.TAUJnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an undefined TAUJTTOUTm signal output.

See **Section 33.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 33.43, Channel Output Modes**.

Table 33.43 Channel Output Modes

Channel Output Mode	TAUJnTOE.TAUJnTOEm	TAUJnTOM.TAUJnTOMm
By software		
Independent channel output mode controlled by software	0	x
By TAUJ signals, independently		
Independent channel output mode 1	1	0
By TAUJ signals, synchronously		
Synchronous channel output mode 1	1	1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUJnTE.TAUJnTEm = 1):

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

33.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

(1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJTTOUTm output.

(2) Set channel output mode according to **Table 33.43, Channel Output Modes**, and the output logic using the TAUJnTOL.TAUJnTOLm bit.

(3) Start the counter (TAUJnTS.TAUJnTSm = 1).

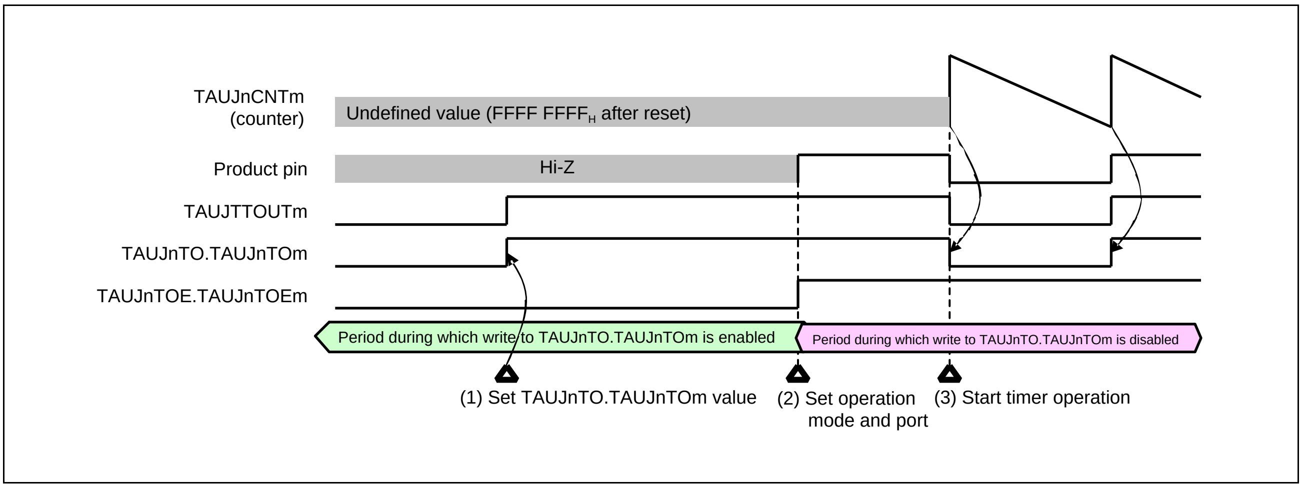


Figure 33.8 General Procedure for Specifying a TAUJTTOUTm Channel Output Mode

33.7.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in **Table 33.43, Channel Output Modes**.

33.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUJTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 33.43, Channel Output Modes**.

33.7.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in **Table 33.43, Channel Output Modes**.

33.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 33.43, Channel Output Modes**.

33.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

33.8.1 Interval Timer Mode, Capture Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.

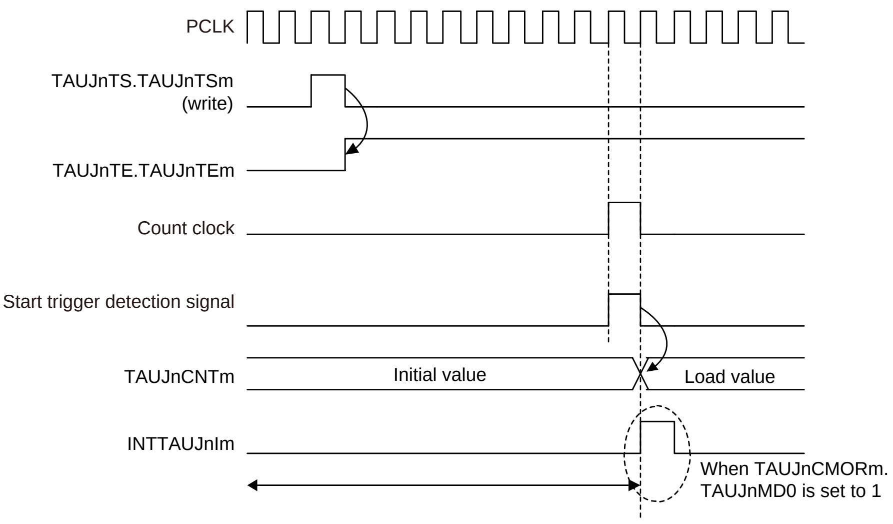


Figure 33.9 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

33.8.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter operation start timing is triggered only upon detection of a valid edge of TAUJTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

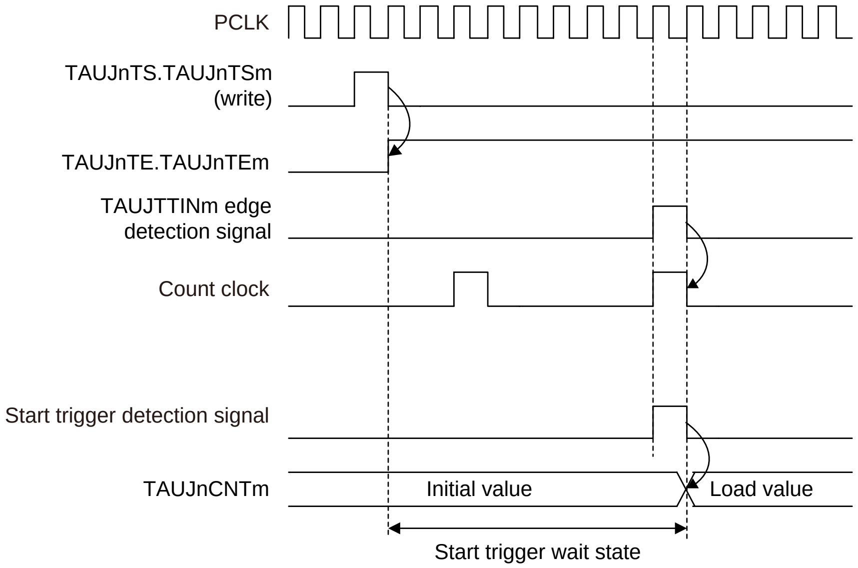


Figure 33.10 Count Start Timing in Other Operating Modes

33.9 TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The generation of INTTAUJnIm when the TAUJnCMORm.TAUJnMD0 bit starts counting and the effect to TAUJTTOUTm depend on the selected function. For details, refer to the description of TAUJnCMORm.TAUJnMD0 of each function.

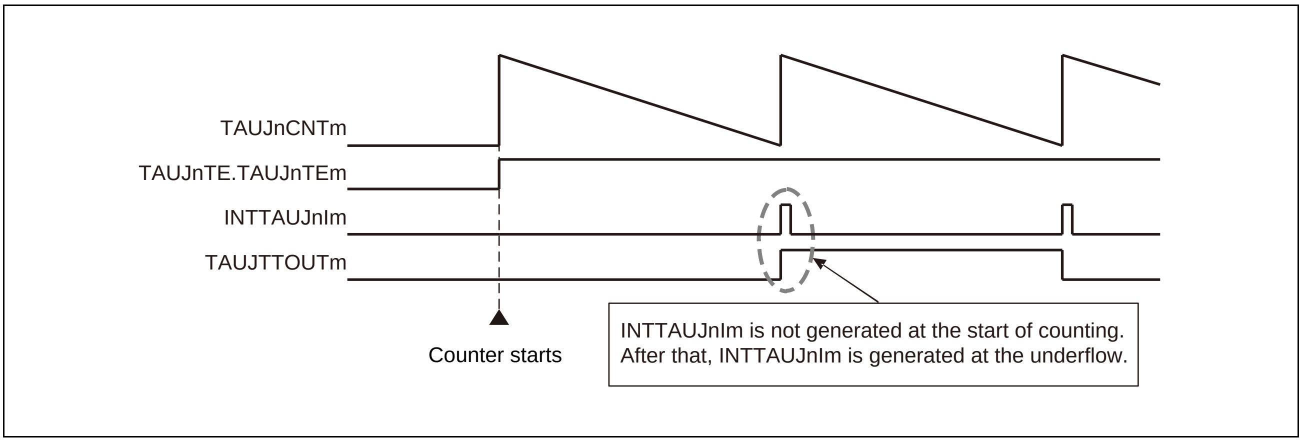


Figure 33.11 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 0)

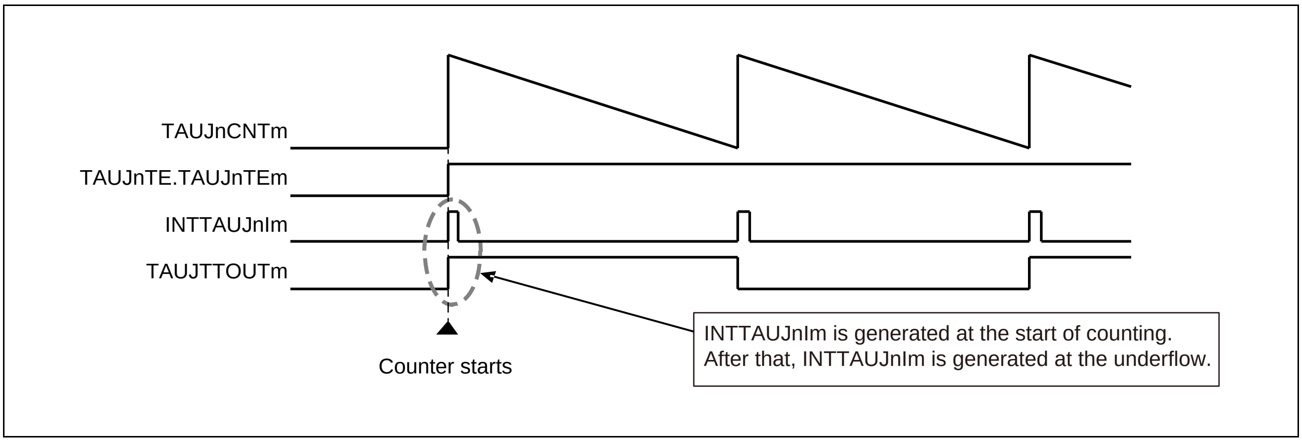


Figure 33.12 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 1)

33.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches FFFF FFFF_H and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operations in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000 0000_H at the same time as the first channel overflows (TAUJnCNTm = FFFF FFFF_H).
- Set TAUJnCDRm of the second channel to FFFF FFFF_H.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUJTTINm input.
- The trigger detection settings (TAUJnCMORm.TAUJnSTS[2:0] and TAUJnCMURm.TAUJnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF FFFF_H). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

33.10.1 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

When the capture trigger is input simultaneously to TAUJTTINm of both channels, INTTAUJnIm of the interval timer function can detect the overflow when TAUJnCNTm of the TAUJTTINm input position detection function exceeds FFFF FFFF$_H$.

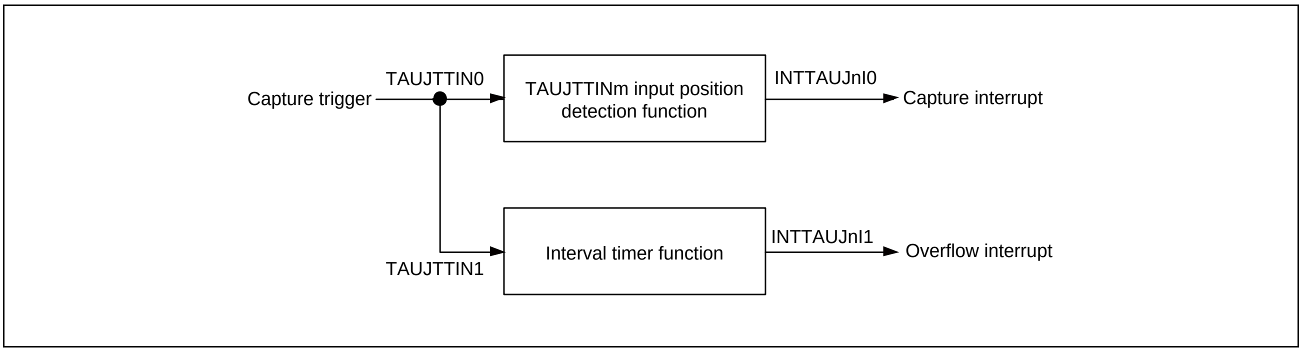


Figure 33.13 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

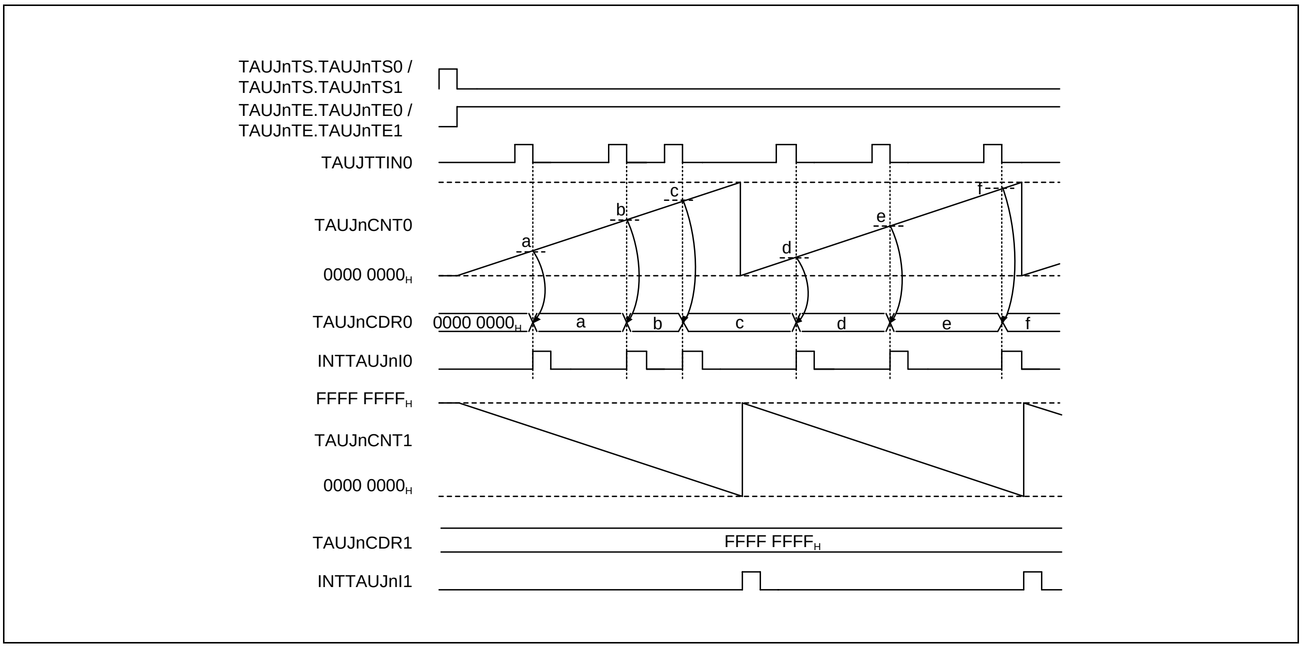


Figure 33.14 Interrupt Generation via Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

33.11 TAUJTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

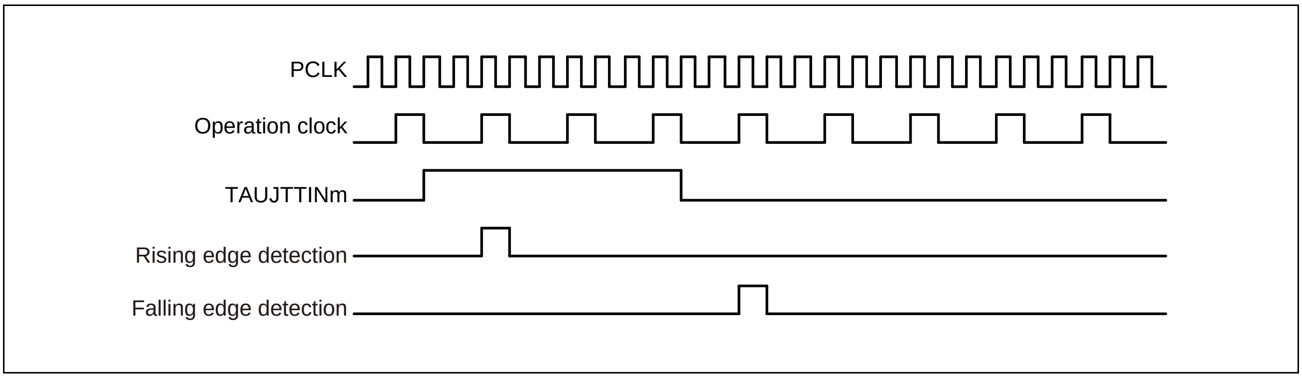


Figure 33.15 Basic Edge Detection Timing

Figure 33.15, Basic Edge Detection Timing shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

33.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see **Section 33.2, Overview**.

33.12.1 Interval Timer Function

33.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave.

Functional description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJTTOUTm does not toggle. This results in a reverted TAUJTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

33.12.1.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm+1) × 2

33.12.1.3 Block Diagram and General Timing Diagram

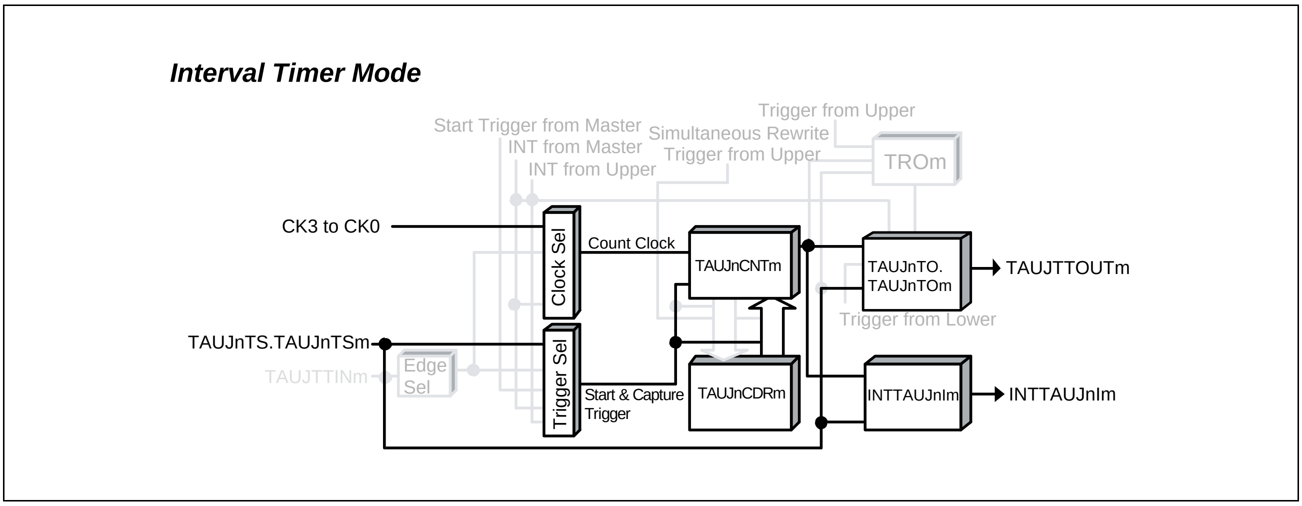


Figure 33.16 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).

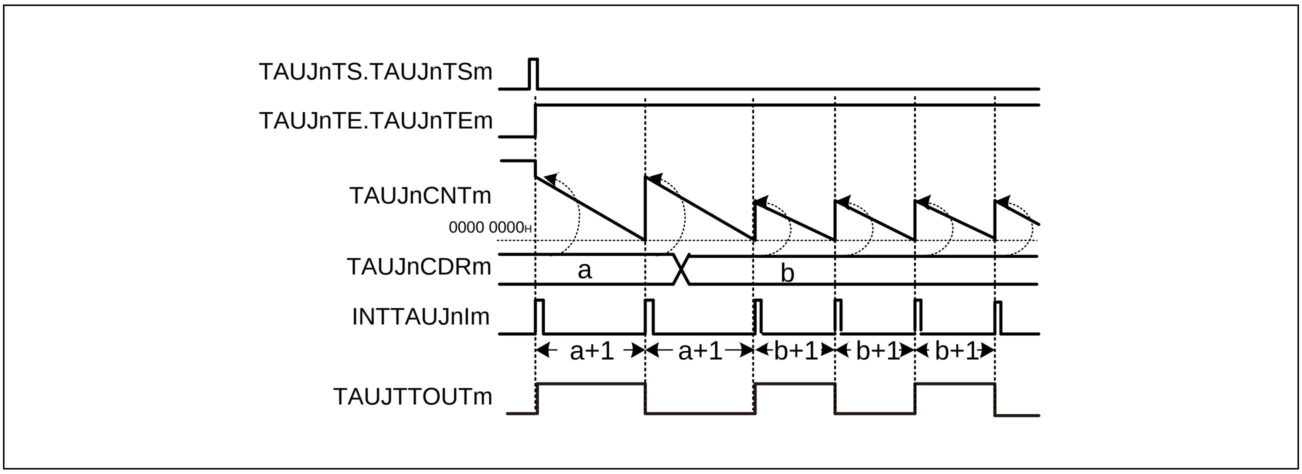


Figure 33.17 General Timing Diagram for Interval Timer Function

33.12.1.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.44 Contents of the TAUJnCMORm register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts or restarts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts or restarts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.45 Contents of the TAUJnCMURm register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel Output Mode

Table 33.46 Control Bit Settings in Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details see **Section 33.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the interval timer function. Therefore, these registers must be set to 0.

Table 33.47 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

33.12.1.5 Operating Procedure for Interval Timer Function

Table 33.48 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.44, Contents of the TAUJnCMORm register for Interval Timer Function** and **Table 33.45, Contents of the TAUJnCMURm register for Interval Timer Function**. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in **Table 33.46, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: ● TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. ● INTTAUJnIm is generated and TAUJTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

33.12.1.6 Specific Timing Diagrams

(1) TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK/2

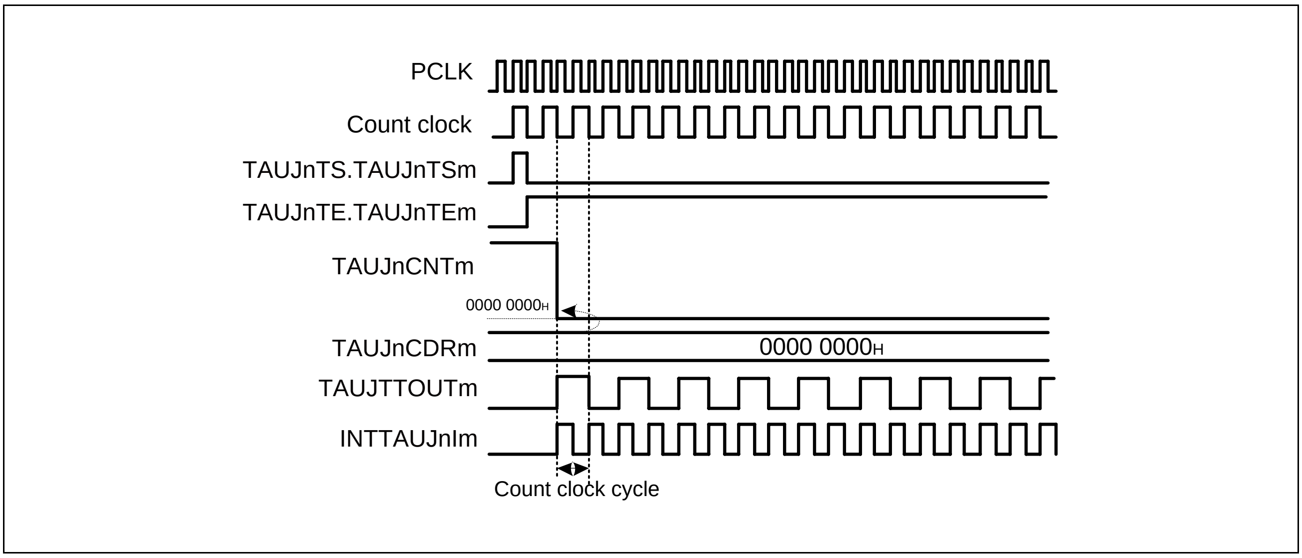


Figure 33.18 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK/2, the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is generated every count clock, resulting in TAUJTTOUTm toggling every count clock.

(2) TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

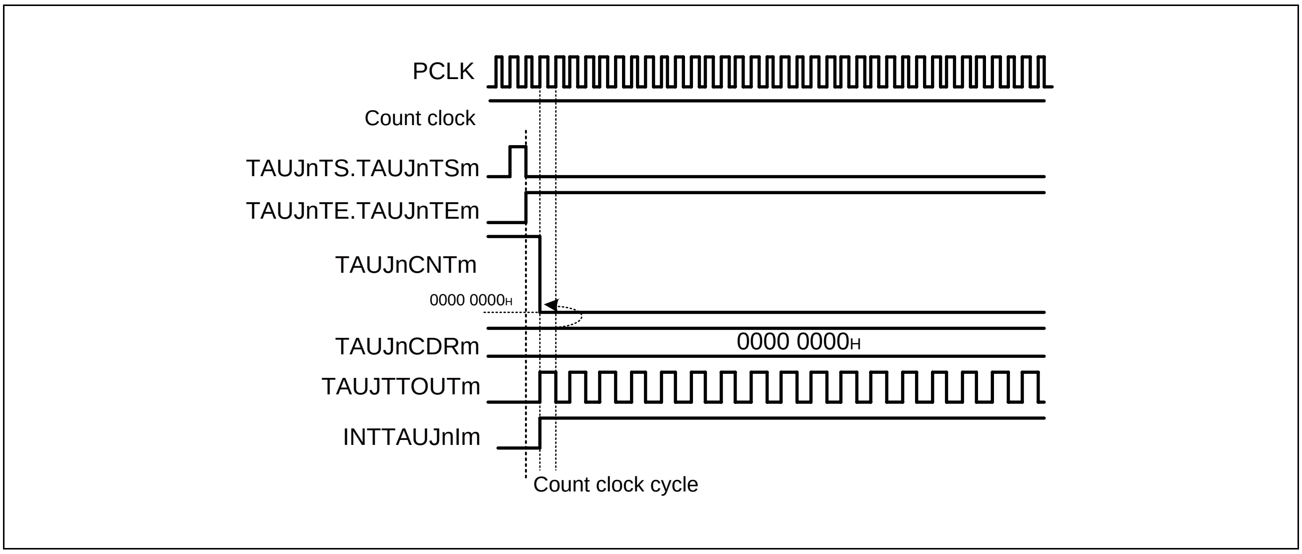


Figure 33.19 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUJTTOUTm is toggled every PCLK clock.

(3) Operation Stop and Restart

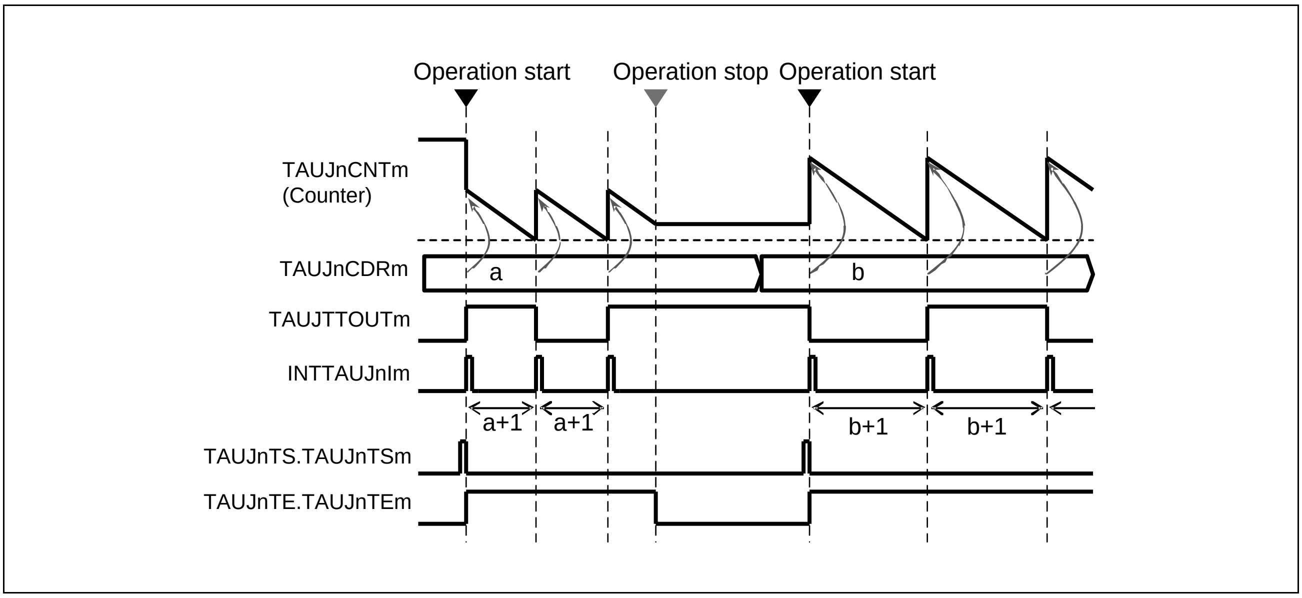


Figure 33.20 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(4) Forced Restart

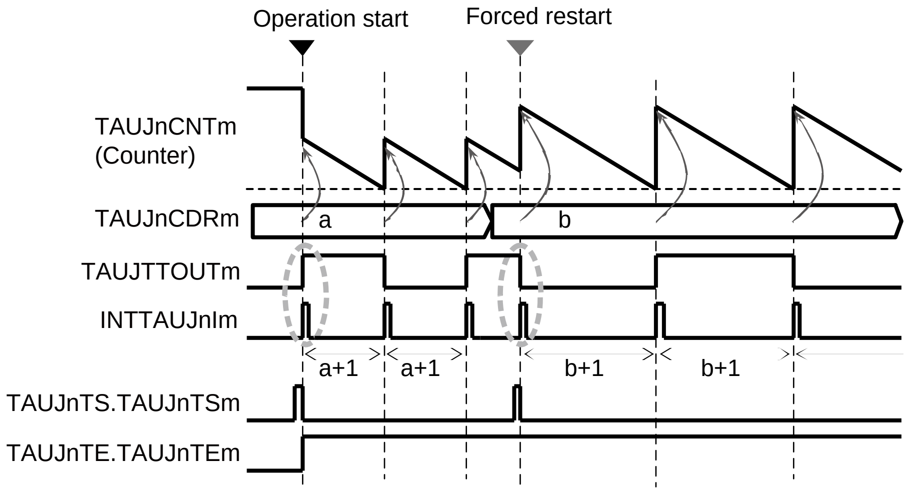


Figure 33.21 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUJnCDRm value is reflected to TAUJnCNTm and counting starts. Execute a forced restart to reflect the changed TAUJnCDRm value immediately.

33.12.2 TAUJTTINm Input Interval Timer Function

33.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJTTINm input edge is detected. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave. Output of square waves is only supported for TAUJ0.

Description

This function operates in an identical manner to the interval timer function (see **Section 33.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUJTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

33.12.2.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

33.12.2.3 Block Diagram and General Timing Diagram

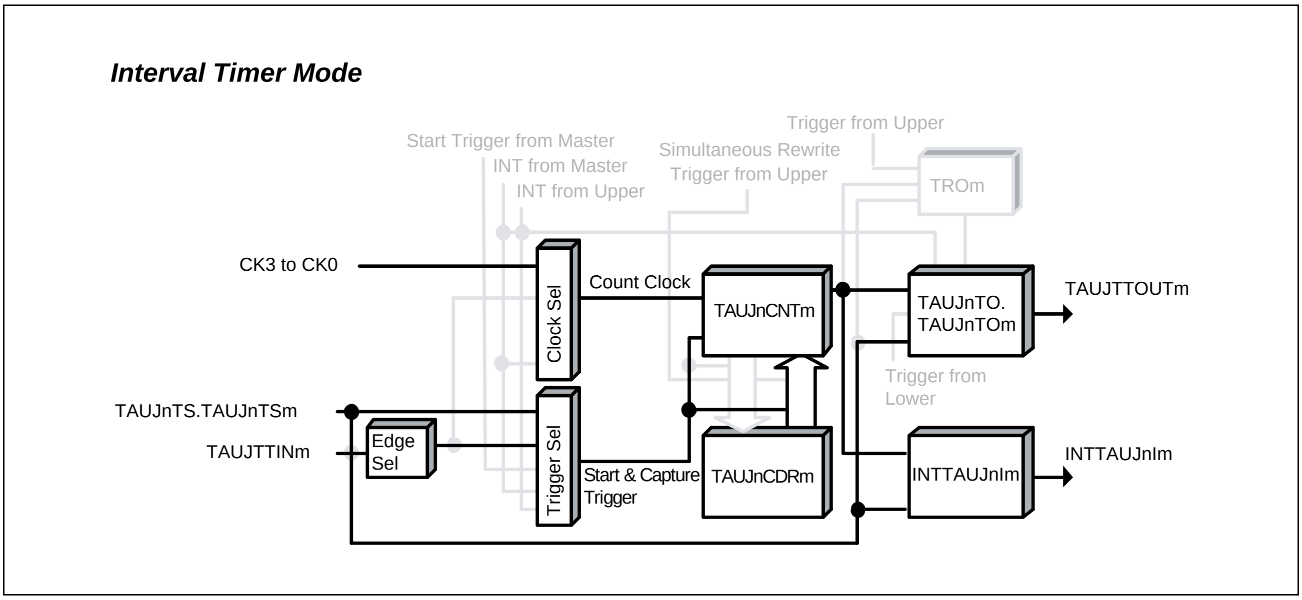


Figure 33.22 Block Diagram for TAUJTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

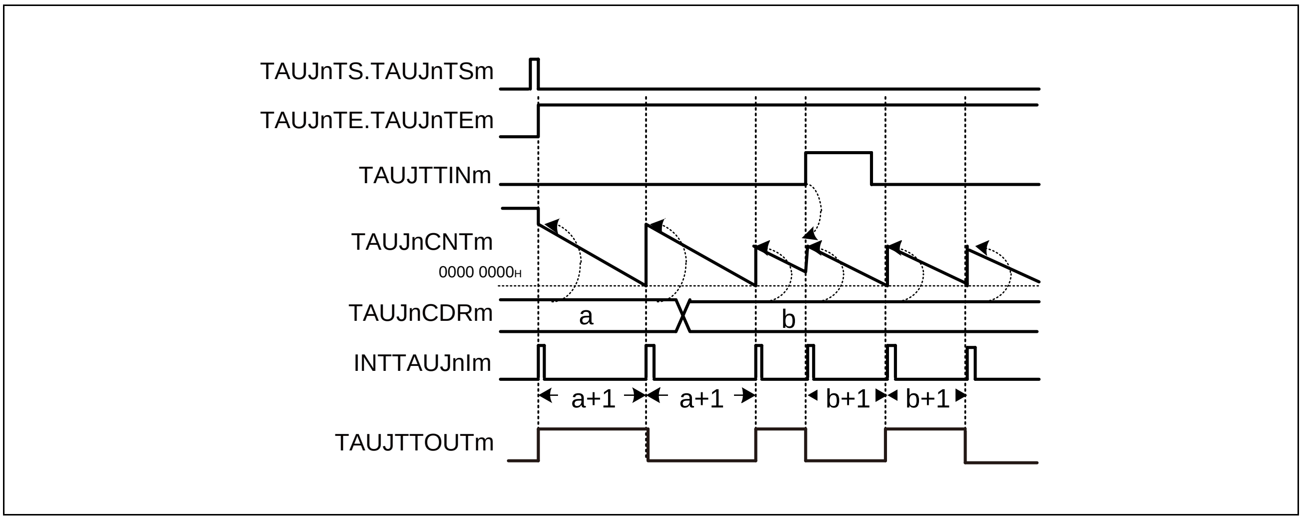


Figure 33.23 General Timing Diagram for TAUJTTINm Input Interval Timer Function

33.12.2.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.49 Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.50 Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel Output Mode

Table 33.51 Control Bit Settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details see **Section 33.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input interval timer function. Therefore, these registers must be set to 0.

Table 33.52 Simultaneous Rewrite Settings for TAUJTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

33.12.2.5 Operating Procedure for TAUJTTINm Input Interval Timer Function

Table 33.53 Operating Procedure for TAUJTTINm Input Interval Timer Function

		Operation	Status of TAUJn
	Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.49, Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function** and **Table 33.50, Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function**. Set the value of the TAUJnCDRm register Set the channel output mode by setting the control bits as described in **Table 33.51, Control Bit Settings for Independent Channel Output Mode 1**.	Channel operation is stopped.
Restart operation	Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
	During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJTTINm edge	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: ● TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. ● INTTAUJnIm is generated and TAUJTTOUTm toggles. When a TAUJTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
	Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

33.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 33.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUJTTINm input edge.

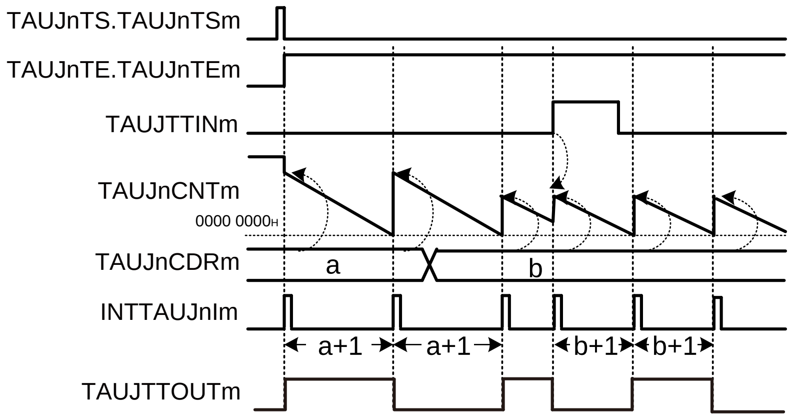


Figure 33.24 Counter Triggered by Rising TAUJTTINm Input Edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

If a valid TAUJTTINm input edge is detected, an interrupt is generated which causes TAUJTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

33.12.3 TAUJTTINm Input Pulse Interval Measurement Function

33.12.3.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJTTINm input signals.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm edge is detected, it overflows to 0000 0000_H. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 33.54 Effects of an Overflow

TAUJnCMORm. TAUJnCOS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input is then Detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJTTINm input valid edge detection and TAUJnCNTm capture are not performed.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 33.9, TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

33.12.3.2 Equations

TAUJTTINm input pulse interval = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF FFFF_H + 1)) + TAUJnCDRm capture value + 1]

33.12.3.3 Block Diagram and General Timing Diagram

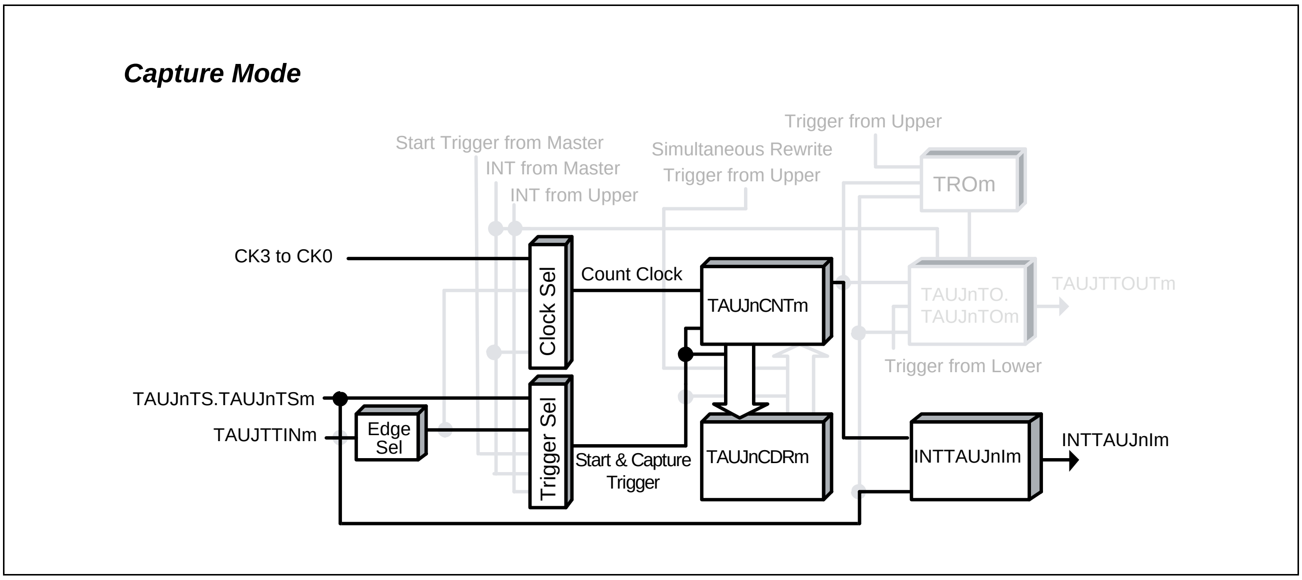


Figure 33.25 Block Diagram for TAUJTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

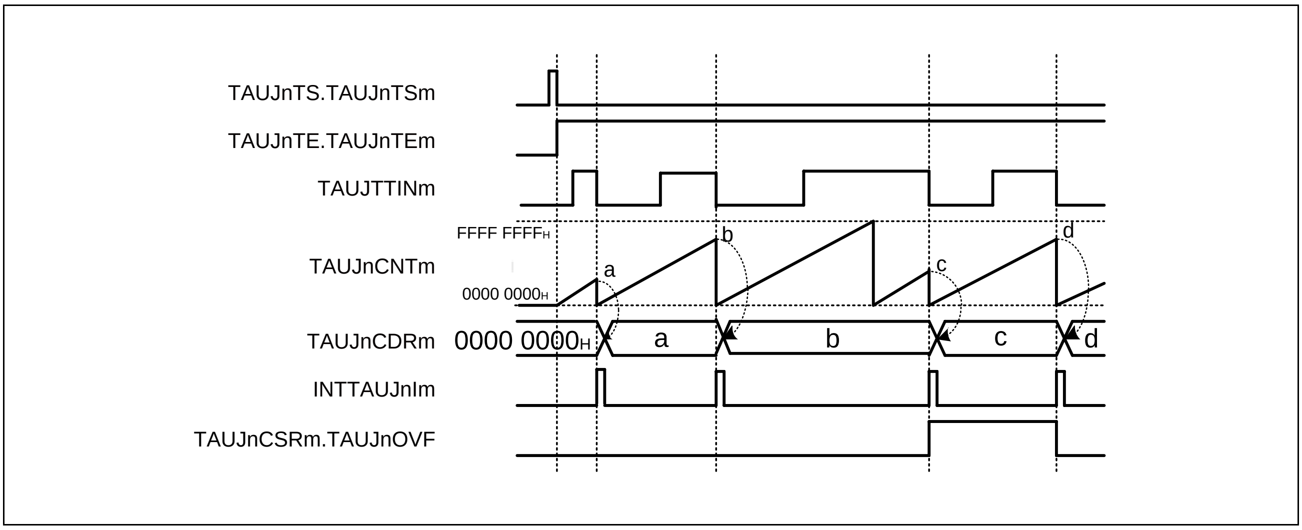


Figure 33.26 General Timing Diagram For TAUJTTINm Input Pulse Interval Measurement Function

33.12.3.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.55 Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_{B}.
11	TAUJnMAS	Write 0_{B}.
10 to 8	TAUJnSTS[2:0]	Write 001_{B}.
7, 6	TAUJnCOS[1:0]	See **Table 33.54, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0010_{B}.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.56 Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input pulse interval measurement function. Therefore, these registers must be set to 0.

Table 33.57 Simultaneous Rewrite Settings for TAUJTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

33.12.3.5 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

Table 33.58 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.55, Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function** and **Table 33.56, Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJTTINm edges. The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: ● TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H. ● INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

33.12.3.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

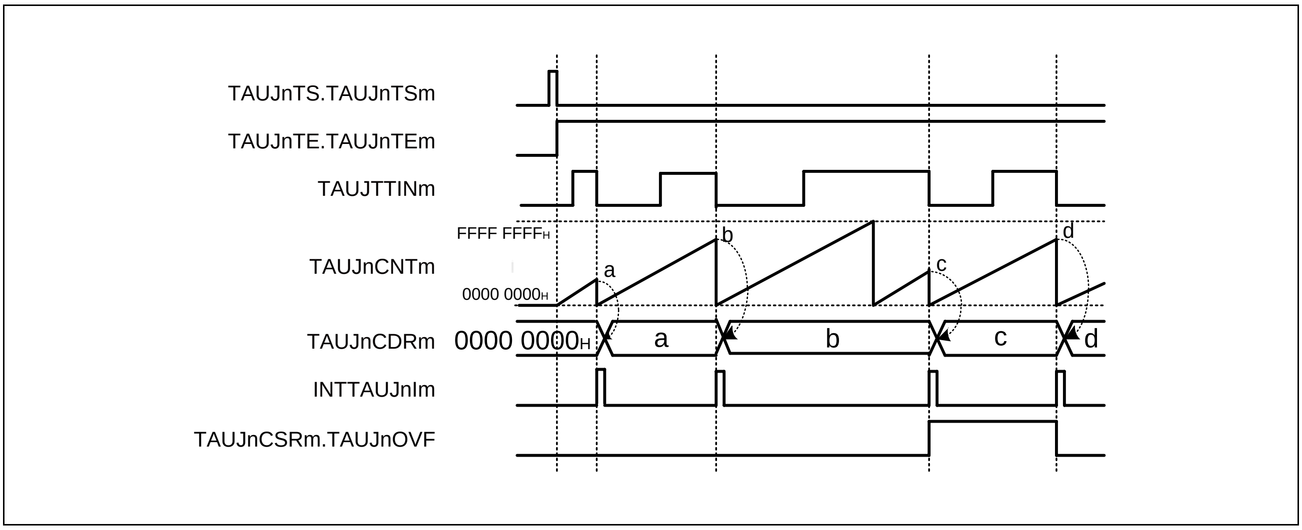


Figure 33.27 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

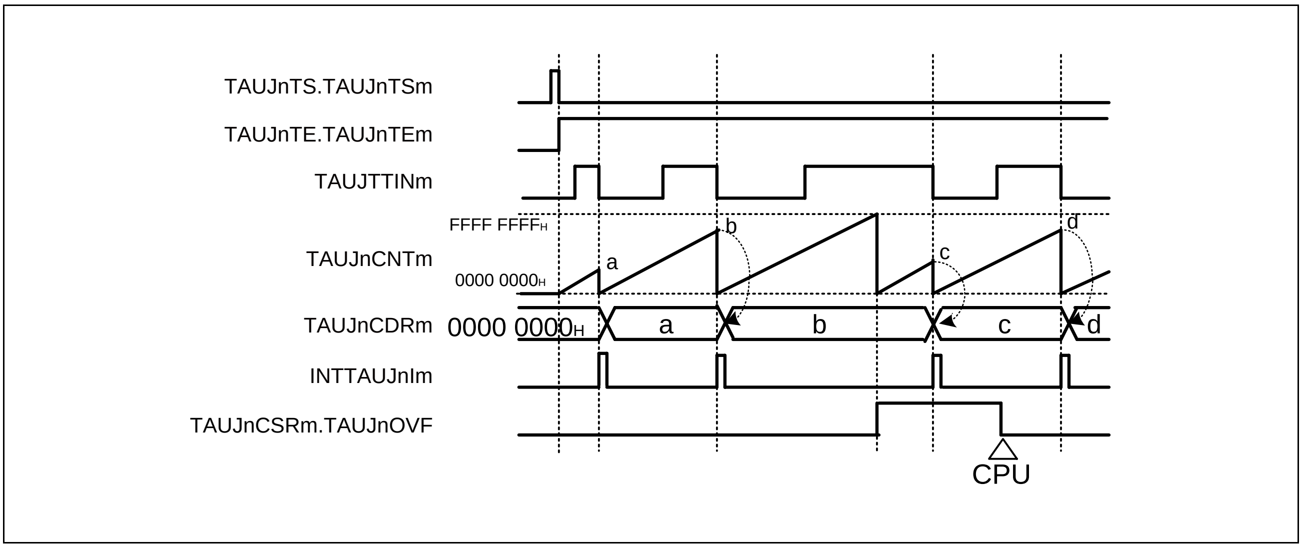


Figure 33.28 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

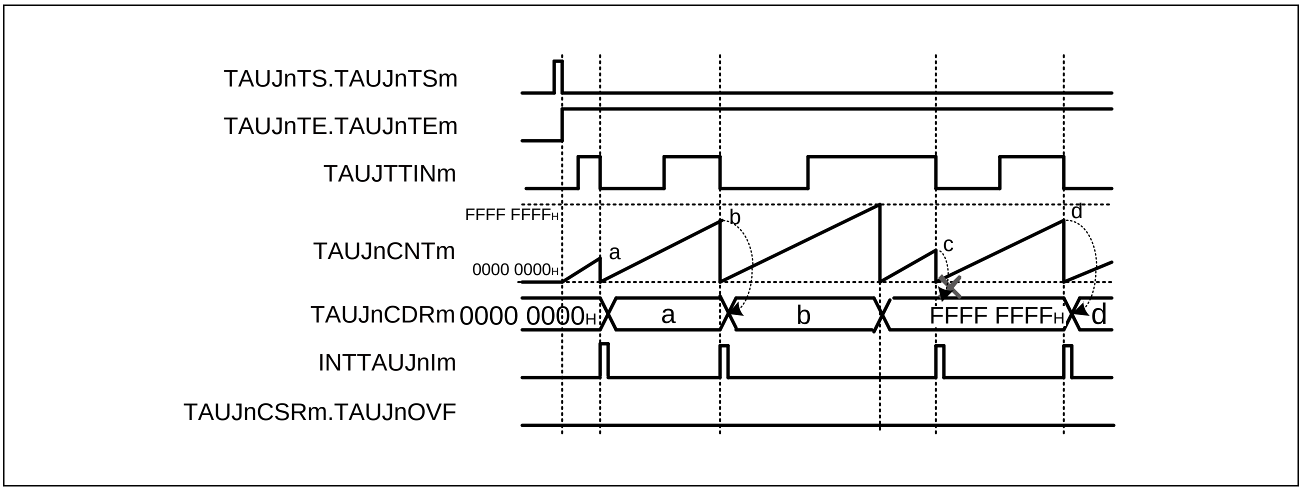


Figure 33.29 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF_H and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

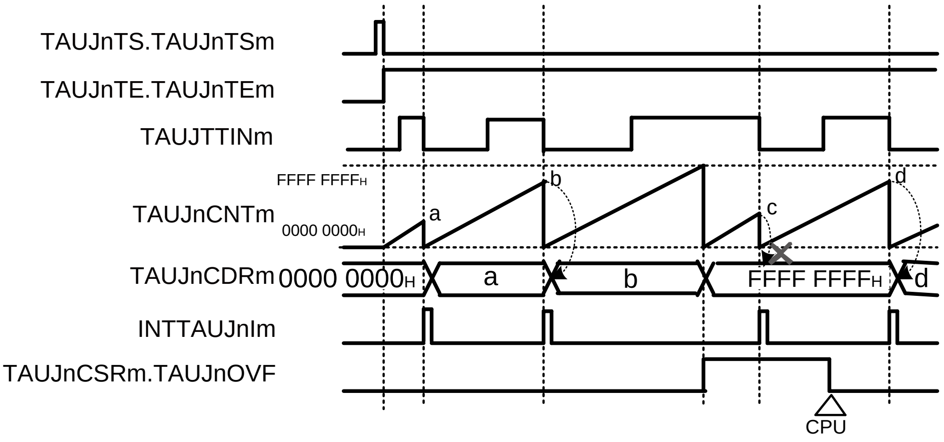


Figure 33.30 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

33.12.4 TAUJTTINm Input Signal Width Measurement Function

33.12.4.1 Overview

Summary

This function measures the width of a TAUJTTINm signal by starting counting on one edge of the TAUJTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 33.59 Effects of an Overflow

TAUJnCMORm.TAUJnCOS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input Stop Edge is Detected	
	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded to TAUJnCDRm.	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUJnCMORm.COS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

33.12.4.2 Equations

TAUJTTINm input signal width = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

33.12.4.3 Block Diagram and General Timing Diagram

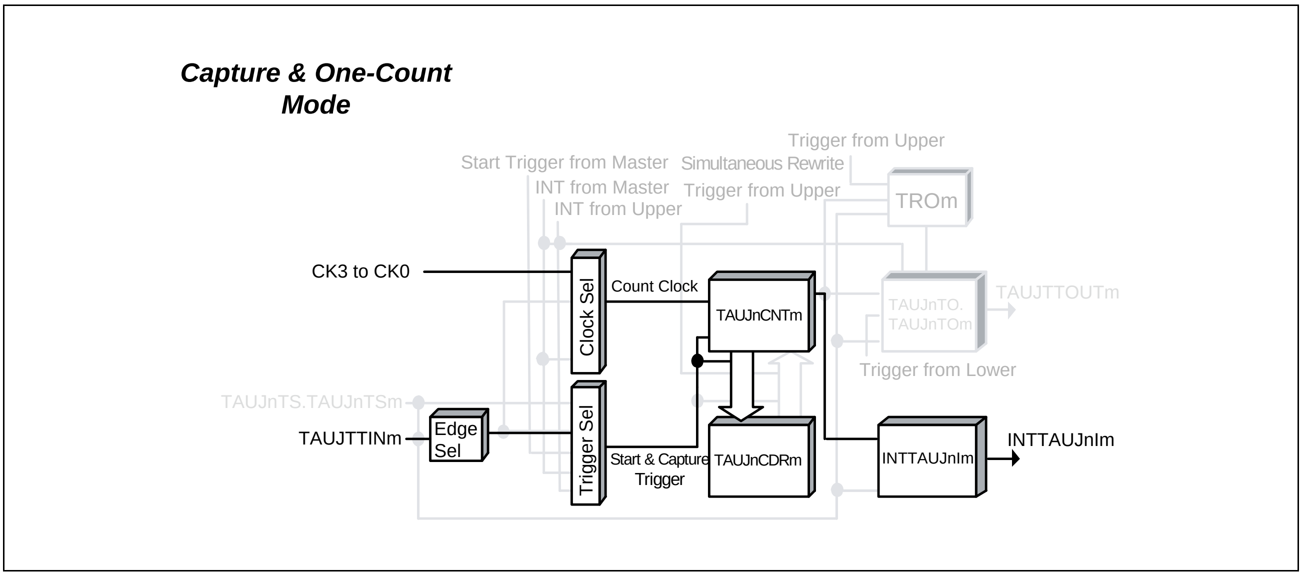


Figure 33.31 Block Diagram for TAUJTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

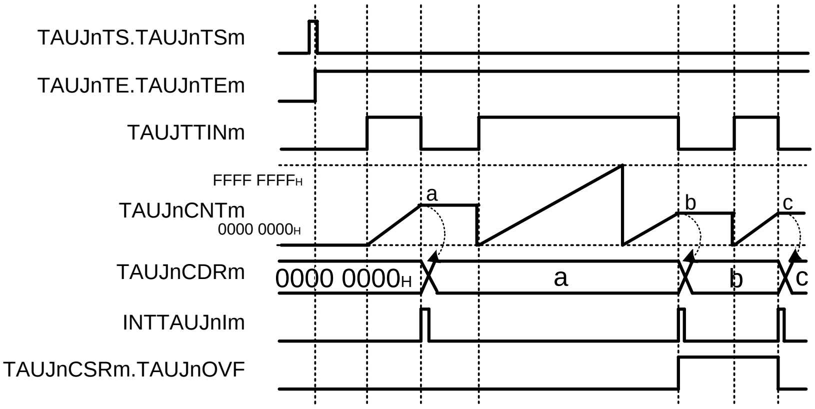


Figure 33.32 General Timing Diagram for TAUJTTINm Input Signal Width Measurement Function

33.12.4.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.60 Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	See **Table 33.59, Effects of an Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0110_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.61 Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input signal width measurement function. Therefore, these registers must be set to 0.

Table 33.62 Simultaneous Rewrite Settings for TAUJTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

33.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

Table 33.63 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

		Operation	Status of TAUJn
	Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.60, Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function** and **Table 33.61, Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Restart operation	Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When a TAUJTTINm start edge is detected, TAUJnCNTm starts to count up.
	During operation	The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: ● TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value. ● INTTAUJnIm is then generated. ● Counting stops at the "value that transferred to TAUJnCDRm + 1" and TAUJnCNTm waits for detection of the TAUJTTINm start edge. Afterwards, this procedure is repeated.
	Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

33.12.4.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

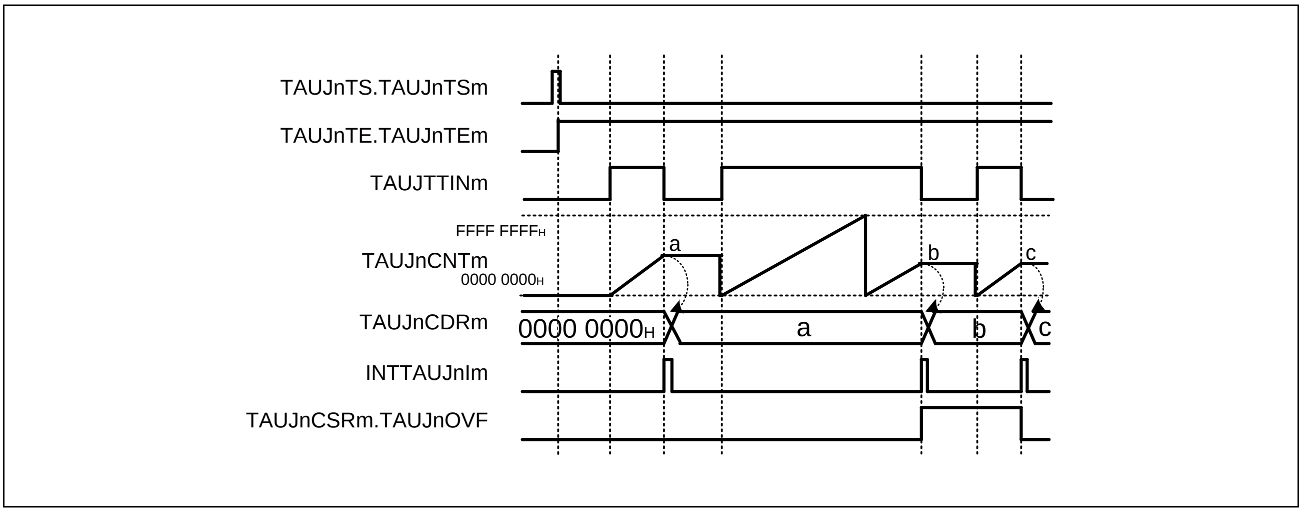


Figure 33.33 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

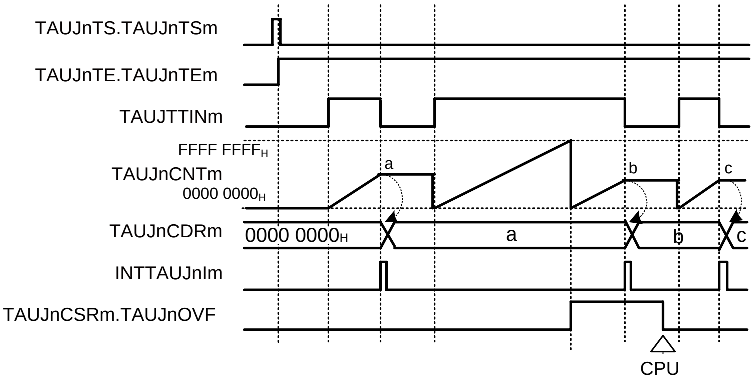


Figure 33.34 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

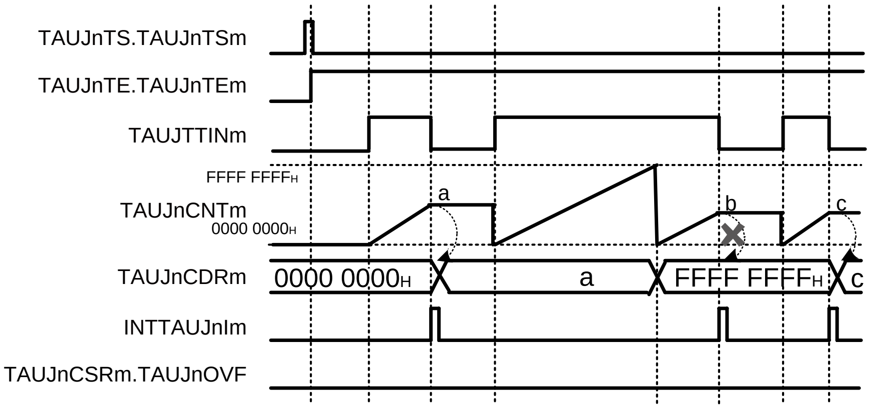


Figure 33.35 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

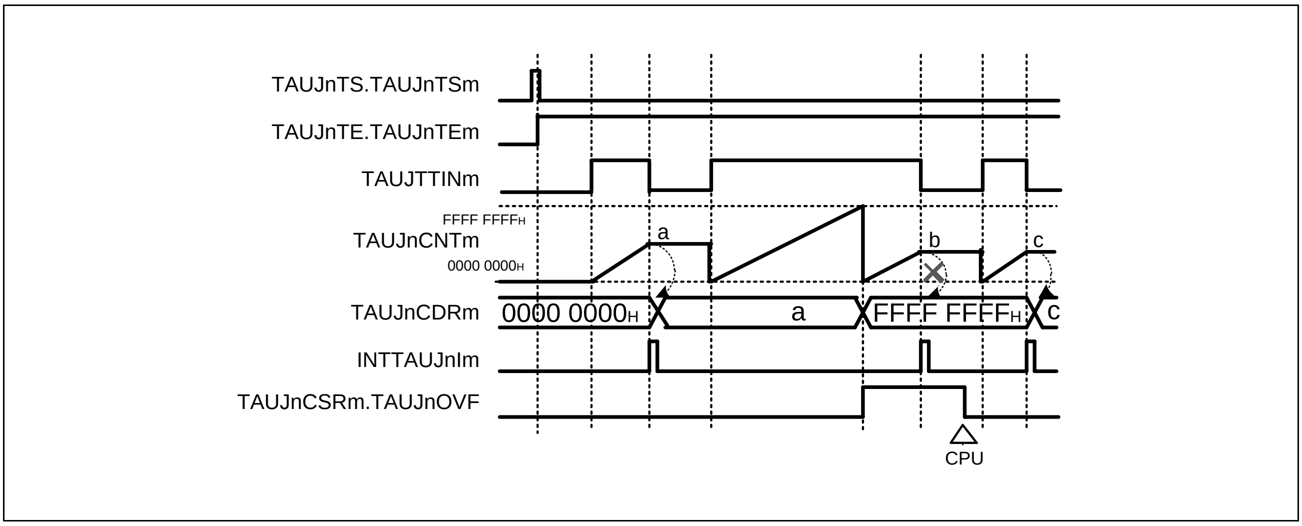


Figure 33.36 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

33.12.5 TAUJTTINm Input Position Detection Function

33.12.5.1 Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUJTTINm signal.

Prerequisites

TAUJTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 0000_H. When a valid TAUJTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

NOTE

The input TAUJTTINm is sampled at the frequency of the operation clock, specified by TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

If the TAUJnCMORm. TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

33.12.5.2 Equations

Function duration at a TAUJTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

33.12.5.3 Block Diagram and General Timing Diagram

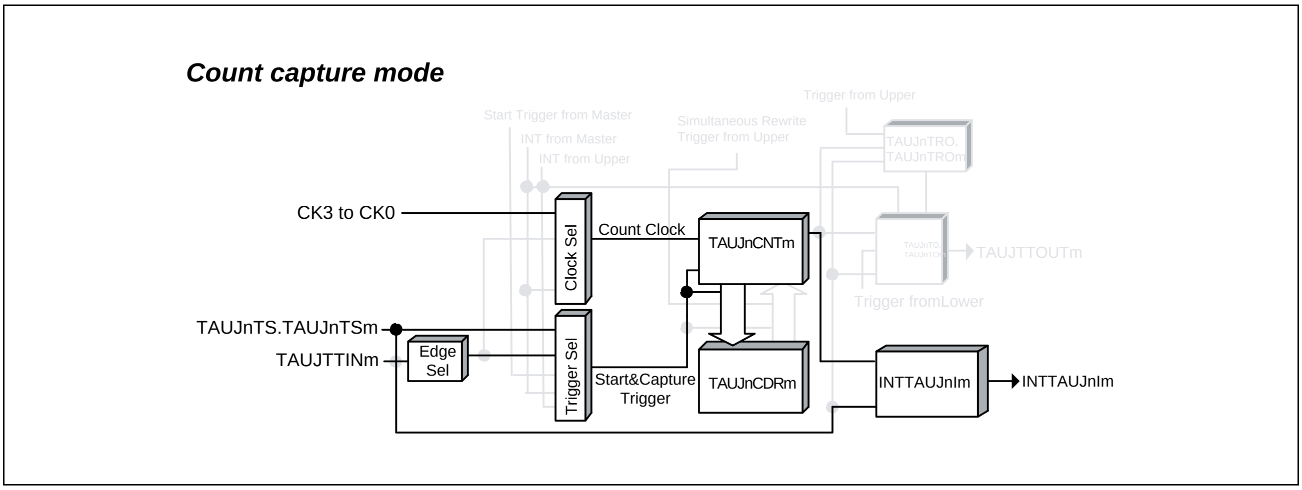


Figure 33.37 Block Diagram of TAUJTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

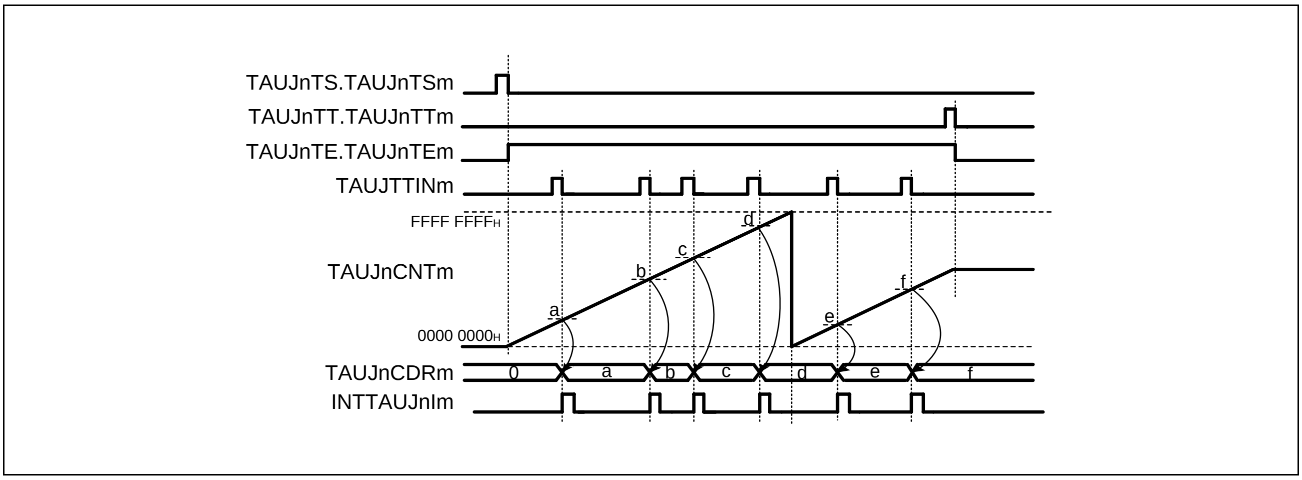


Figure 33.38 General Timing Diagram for TAUJTTINm Input Position Detection Function

33.12.5.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.64 Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_{B}.
11	TAUJnMAS	Write 0_{B}.
10 to 8	TAUJnSTS[2:0]	Write 001_{B}.
7, 6	TAUJnCOS[1:0]	Write 01_{B}.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1011_{B}.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.65 Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel Output Mode

The channel output mode is not used by this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input position detection function. Therefore, these registers must be set to 0.

Table 33.66 Simultaneous Rewrite Settings for TAUJTTINm Input Position Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

33.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function

Table 33.67 Operating Procedure for TAUJTTINm Input Position Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 33.64, Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function** and **Table 33.65, Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time.	TAUJnCNTm starts to count up from $0000\ 0000_H$. When a TAUJTTINm valid edge is detected: ● TAUJnCNTm transfers (captures) its value to TAUJnCDRm. ● INTTAUJnIm is output. ● The counter value is not cleared to $0000\ 0000_H$ and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated. When TAUJnCNTm reaches $FFFF\ FFFF_H$, the counter restarts from $0000\ 0000_H$.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

33.12.5.6 Specific Timing Diagrams

(1) Operation Stop and Restart

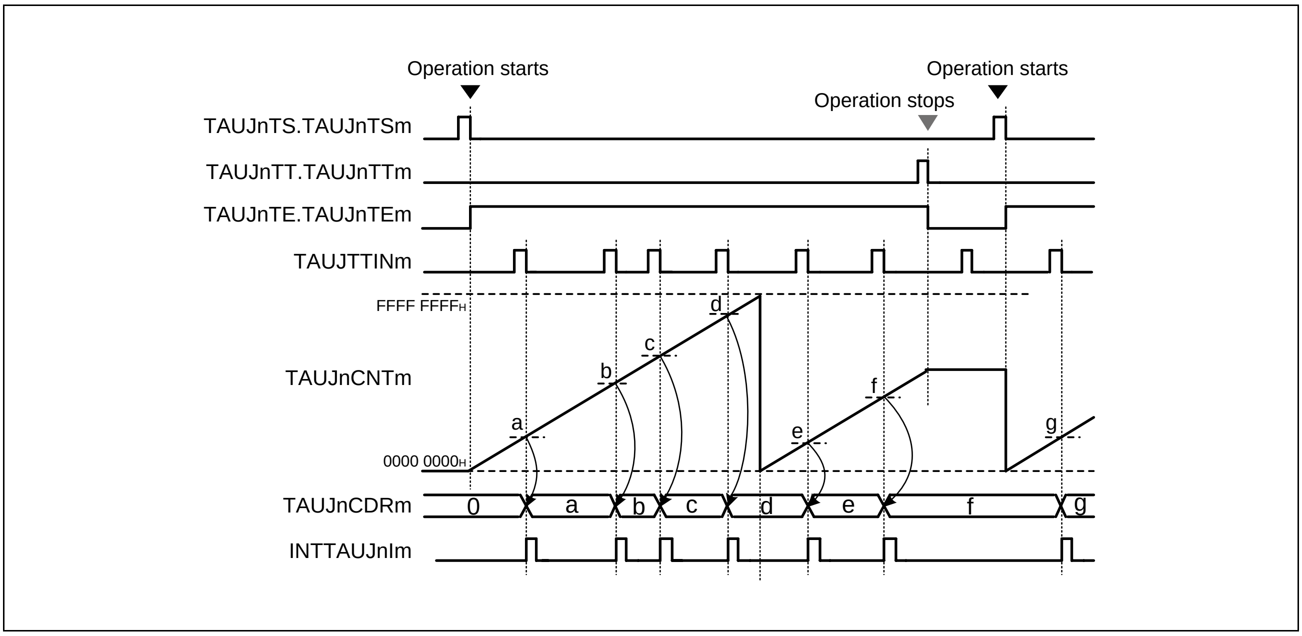


Figure 33.39 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

33.12.6 TAUJTTINm Input Period Count Detection Function

33.12.6.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJTTINm input edge.

When a valid TAUJTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJTTINm input start edge is detected.

When the next valid TAUJTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

This function cannot be forcibly restarted.

NOTE

The input TAUJTTINm signal is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

33.12.6.2 Equations

Cumulative TAUJTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

33.12.6.3 Block Diagram and General Timing Diagram

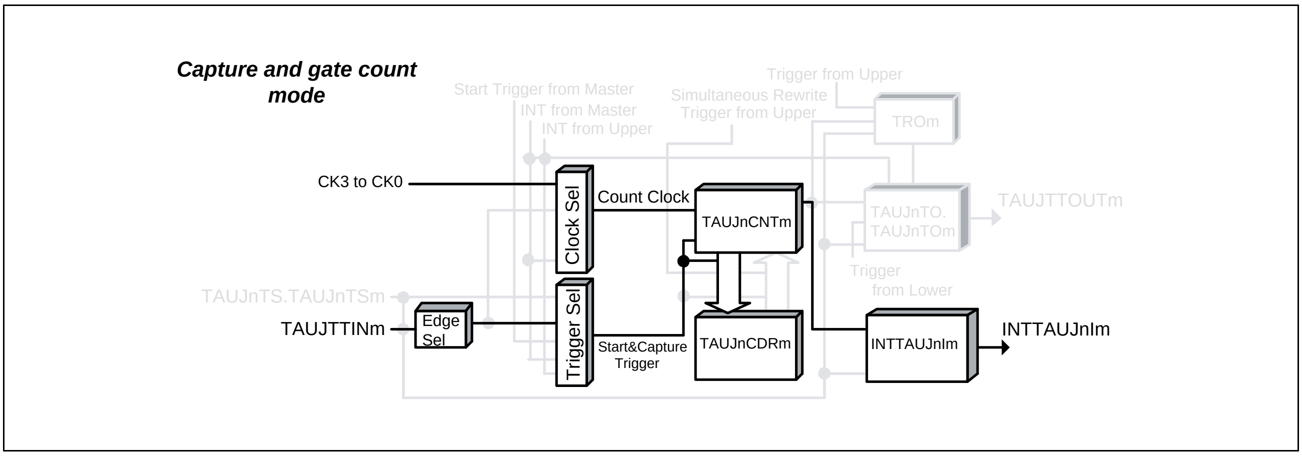


Figure 33.40 Block Diagram for TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

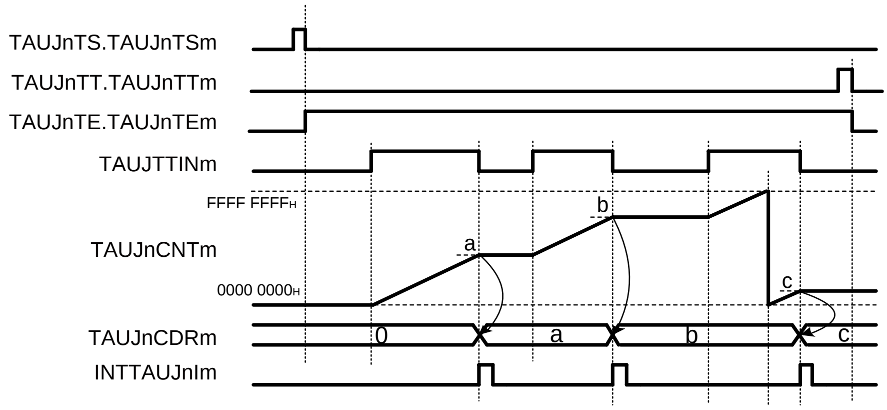


Figure 33.41 General Timing Diagram for TAUJTTINm Input Period Count Detection Function

33.12.6.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.68 Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1101_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.69 Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input period count detection function. Therefore, these registers must be set to 0.

Table 33.70 Simultaneous Rewrite Settings for TAUJTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

33.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function

Table 33.71 Operating Procedure for TAUJTTINm Input Period Count Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.68, Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function** and **Table 33.69, Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge.
During operation	TAUJTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time.	When a TAUJTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When TAUJnCNTm reaches FFFF FFFF$_H$, the counter restarts from 0000 0000$_H$. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

33.12.6.6 Specific Timing Diagrams

(1) Operation Stop and Restart

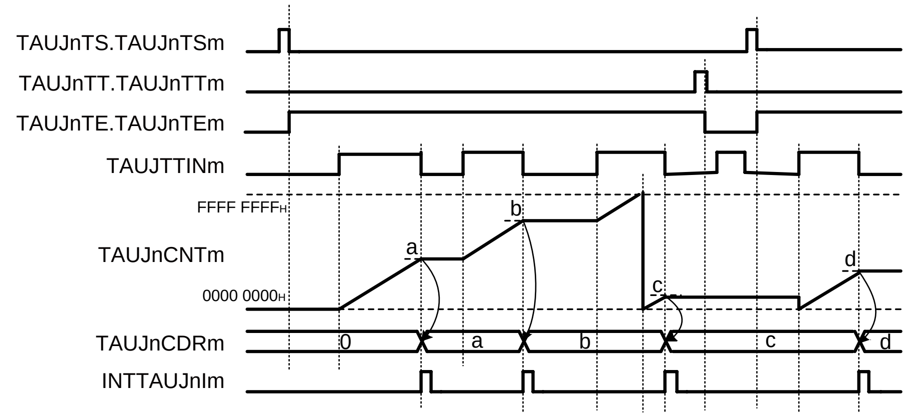


Figure 33.42 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

33.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

33.12.7.1 Overview

Summary

This function measures the width of an individual TAUJTTINm input signal. An interrupt is generated if the TAUJTTINm input width is longer than FFFF FFFF_H + 1.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF_H.

Description

he counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF FFFF_H is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJTTINm input start edge is detected, TAUJnCNTm loads FFFF FFFF_H and starts to count down.

If the counter reaches 0000 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

33.12.7.2 Block Diagram and General Timing Diagram

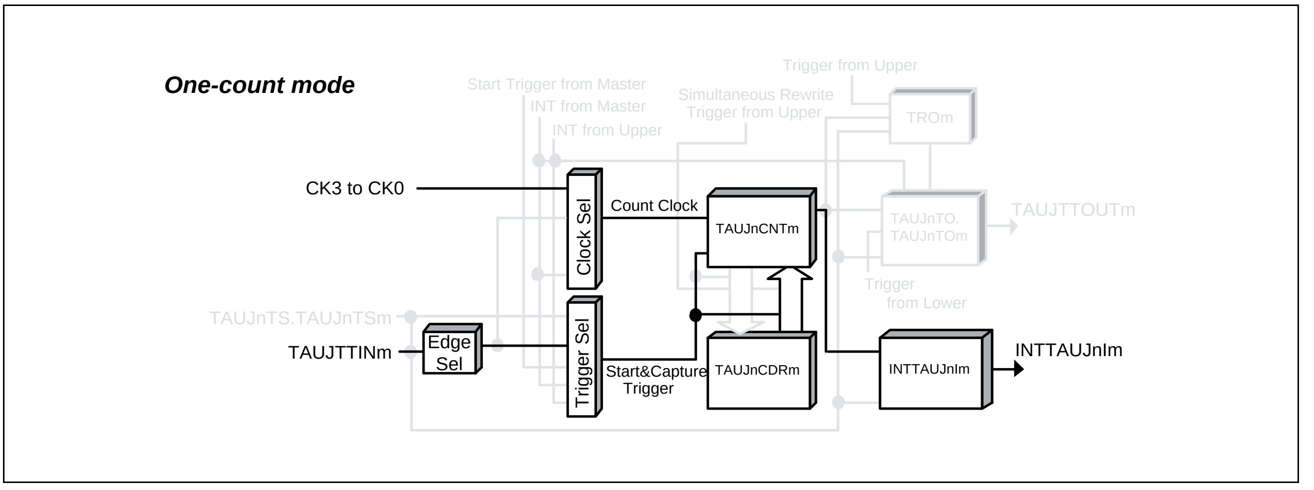


Figure 33.43 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

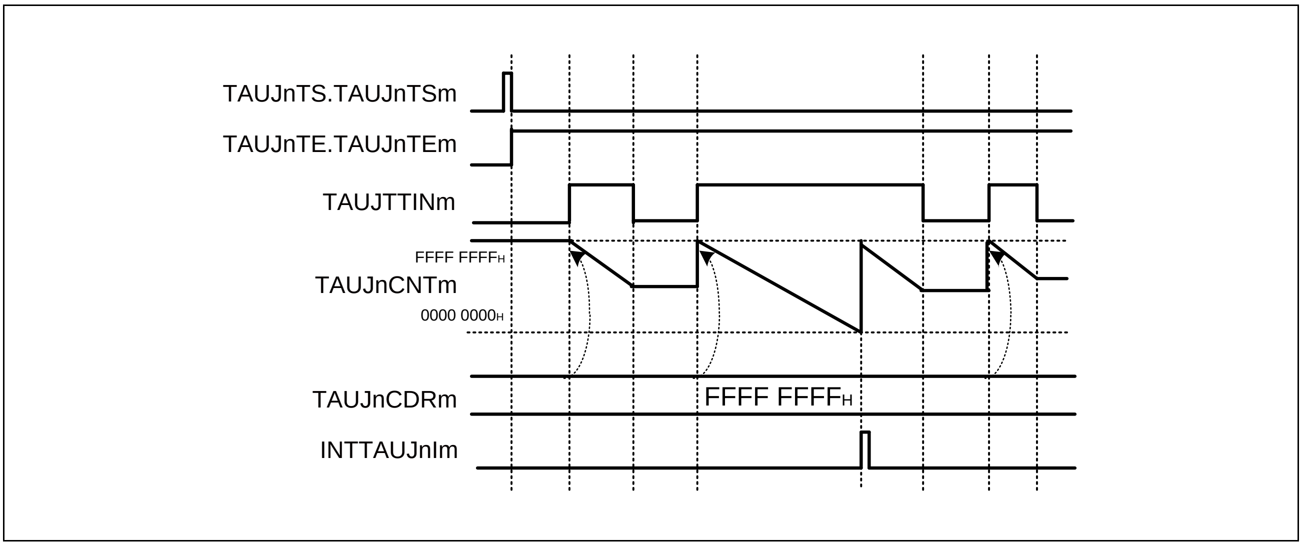


Figure 33.44 General Timing Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

33.12.7.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.72 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.73 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the overflow interrupt output function (during TAUJTTINm width measurement). Therefore, these registers must be set to 0.

Table 33.74 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

33.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Table 33.75 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 33.72, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)** and **Table 33.73, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at any time.	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: ● INTTAUJnIm is generated. When a reverse edge of TAUJTTINm is detected during count operation: ● TAUJnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

33.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

33.12.8.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal. An interrupt is generated if the cumulative TAUJTTINm input width is longer than FFFF FFFF_H, and an overflow interrupt can be output.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF_H.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF FFFF_H is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000 0000_H an interrupt is generated. FFFF FFFF_H is loaded to TAUJnCNTm and the counter continues to count down until a TAUJTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

33.12.8.2 Block Diagram and General Timing Diagram

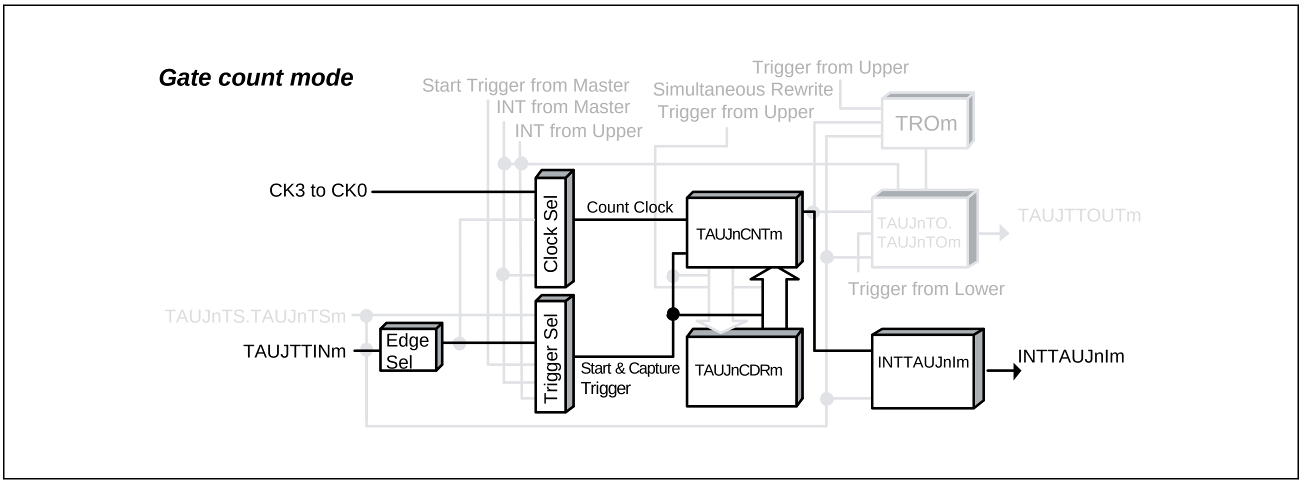


Figure 33.45 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

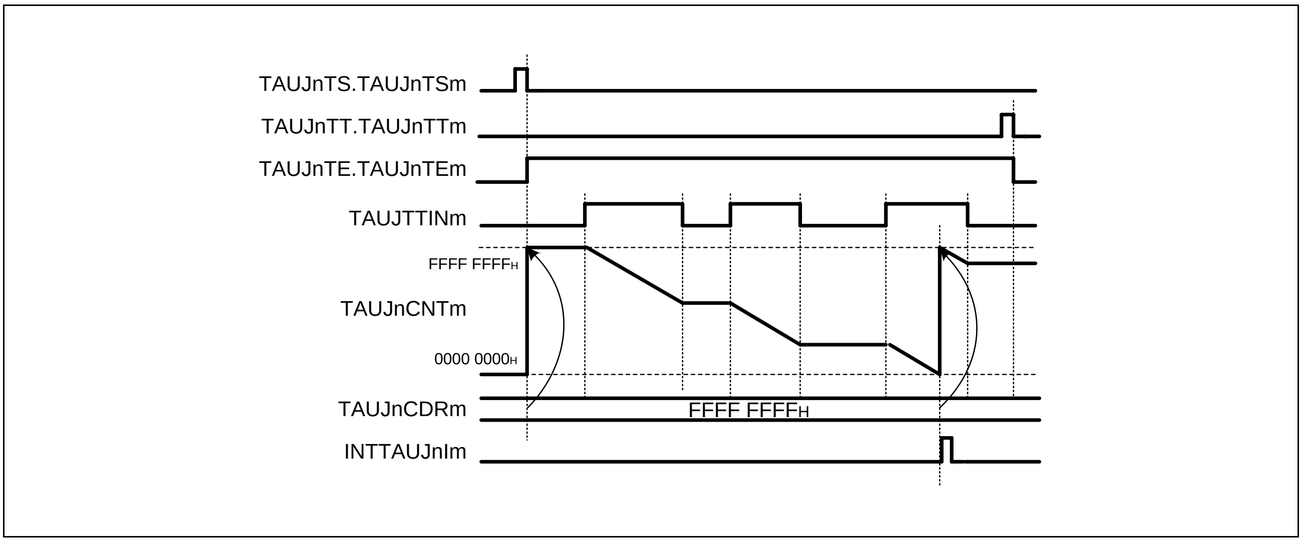


Figure 33.46 General Timing Diagram For Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

33.12.8.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.76 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_{B}.
11	TAUJnMAS	Write 0_{B}.
10 to 8	TAUJnSTS[2:0]	Write 010_{B}.
7, 6	TAUJnCOS[1:0]	Write 00_{B}.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1100_{B}.
0	TAUJnMD0	Write 0_{B}.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.77 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel Output Mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 33.78 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

33.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Table 33.79 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 33.76, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)** and **Table 33.77, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_{H}$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, the value of TAUJnCDRm (FFFF FFFF$_{H}$) is loaded to TAUJnCNTm.
During operation	The TAUJnCNTm register can be read at all times	TAUJnCNTm counts down. When the counter reaches 0000 0000$_{H}$: ● INTTAUJnIm is generated. ● TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_{H}$) and continues to count down. When a reverse edge of TAUJTTINm is detected during count operation: ● TAUJnCNTm stops and retains the stop value. When a TAUJTTINm valid edge is detected while the counter is stopped: ● TAUJnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

33.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see **Section 33.2, Overview**.

33.13.1 PWM Output Function

33.13.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operation mode for the master channel should be set to the interval timer mode. (See **Table 33.80, Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operation mode for the slave channel should be set to the one-count mode. (See **Table 33.83, Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUJTTOUTm is not used for the master channel of this function.
- The channel output mode for the slave channels should be set to synchronous channel output mode 1. (See **Section 33.7, Channel Output Modes**.)

Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

  When the counter of the master channel reaches 0000 0000_H and pulse cycle time has elapsed, INTTAUJnIm is generated. The TAUJnCDRm value is loaded to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

  INTTAUJnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJTTOUTm signal is set to the active level.
  When the counter reaches 0000 0000_H, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJTTOUTm signal is set to the inactive level. The counter returns to FFFF $FFFF_H$ and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions

Set Simultaneous rewrite with this function. Please see **Section 33.6, Simultaneous Rewrite**.

33.13.1.2 Equations

Pulse cycle = (TAUJnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave)/(TAUJnCDRm (master) +1)) × 100

- Duty cycle = 0%
  TAUJnCDRm (slave) = $0000\ 0000_{H}$
- Duty cycle = 100%
  TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

33.13.1.3 Block Diagram and General Timing Diagram

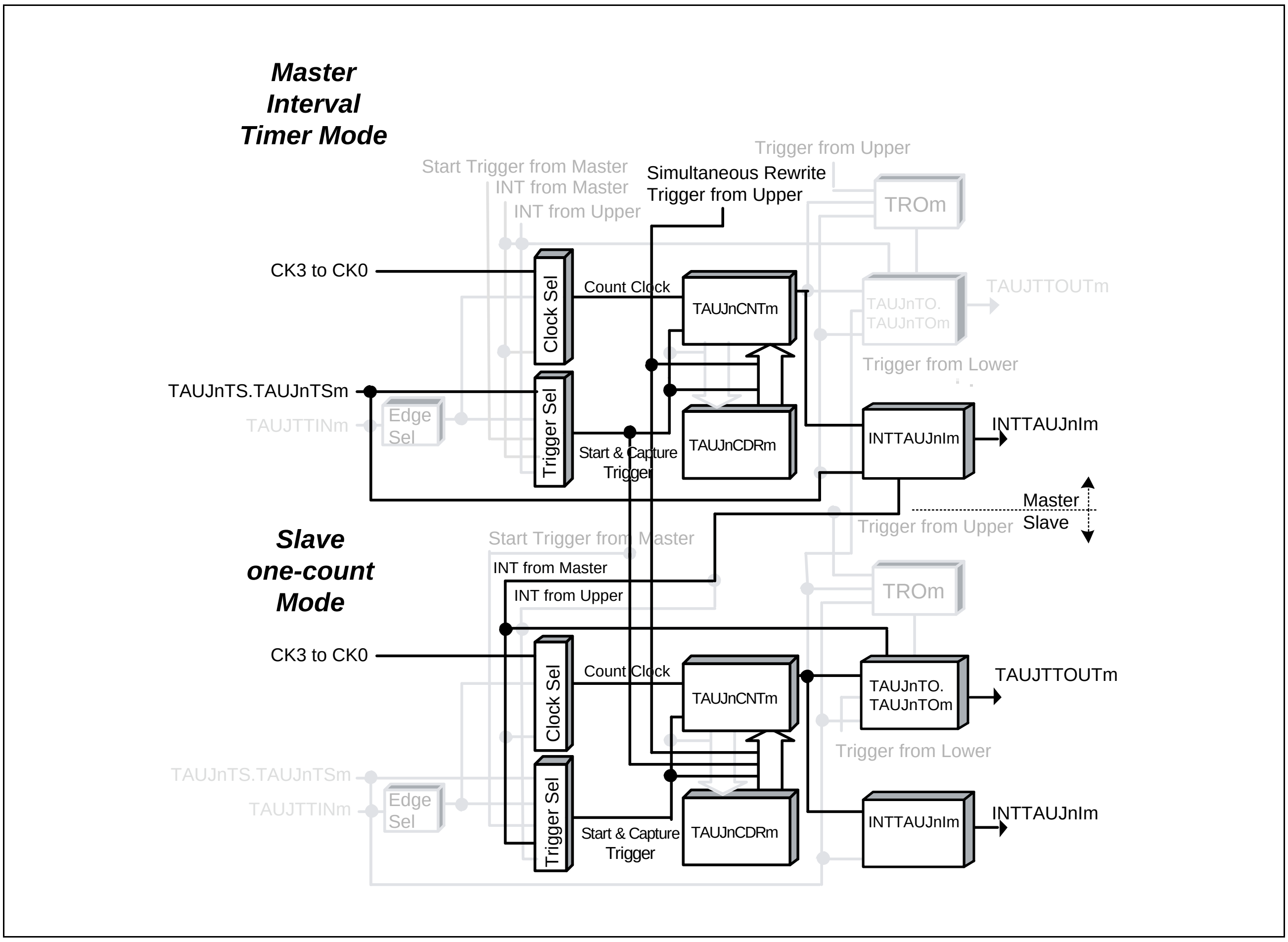


Figure 33.47 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

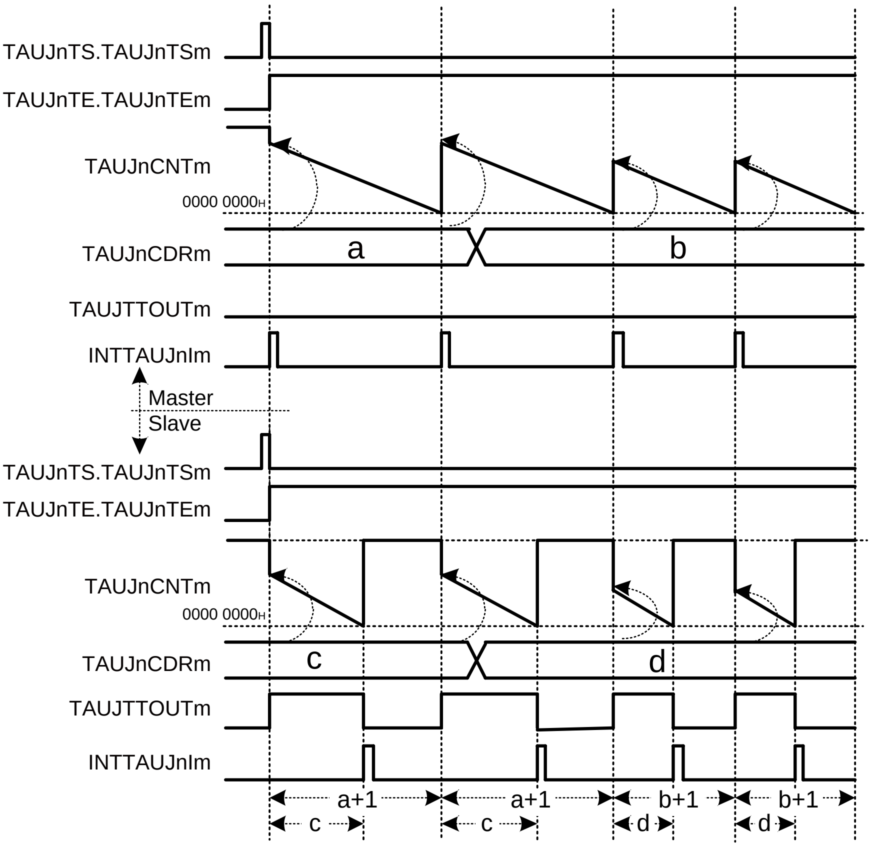


Figure 33.48 General Timing Diagram for PWM Output Function

NOTE

- The interval between the start of the count and an interrupt being generated is the value of corresponding TAUJnCDRm + 1.
- TAUJTTOUTm of the slave channel will rise with a delay of one count clock after the rising of INTTAUJnIm of the master channel.

33.13.1.4 Register Settings for the Master Channel

(1) TAUJnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.80 Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 1_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.81 Contents of the TAUJnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel Output Mode for the Master Channel

The channel output mode is not used by this function.

(4) Simultaneous Rewrite for the Master Channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 33.82 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

33.13.1.5 Register Settings for the Slave Channel(s)

(1) TAUJnCMORm for the Slave Channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.83 Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 100_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the Slave Channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.84 Contents of the TAUJnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel Output Mode for the Slave Channel(s)

Table 33.85 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 1_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Negative logic

(4) Simultaneous Rewrite for the Slave Channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 33.86 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

33.13.1.6 Operating Procedure for PWM Output Function

Table 33.87 Operating Procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	Master channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 33.13.1.4, Register Settings for the Master Channel**. Slave channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 33.13.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUJnCDRm registers of all channels.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set.
During operation	TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation.	TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches $0000\ 0000_H$: ● INTTAUJnIm (master) is generated. ● TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation. ● TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down. ● TAUJTTOUTm (slave) is set to the active level. When TAUJnCNTm (slave) reaches $0000\ 0000_H$: ● INTTAUJnIm (slave) is generated. ● TAUJTTOUTm (slave) is set to the inactive level.
Stop operation	Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

33.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

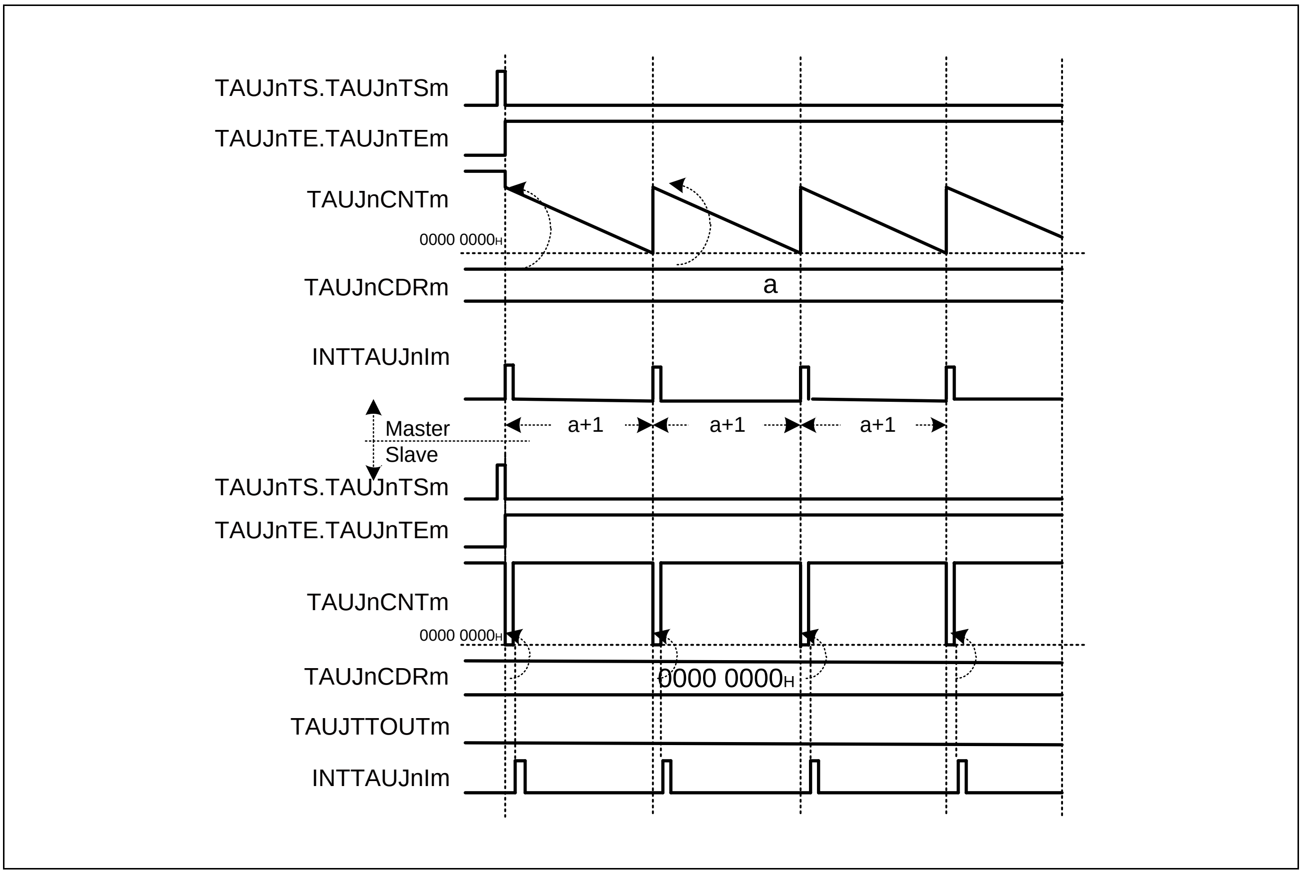


Figure 33.49 TAUJnCDRm (slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded to TAUJnCNTm (slave). As a result, a slave channel interrupt (INTTAUJnIm) is generated at the same time and TAUJTTOUTm remains inactive.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

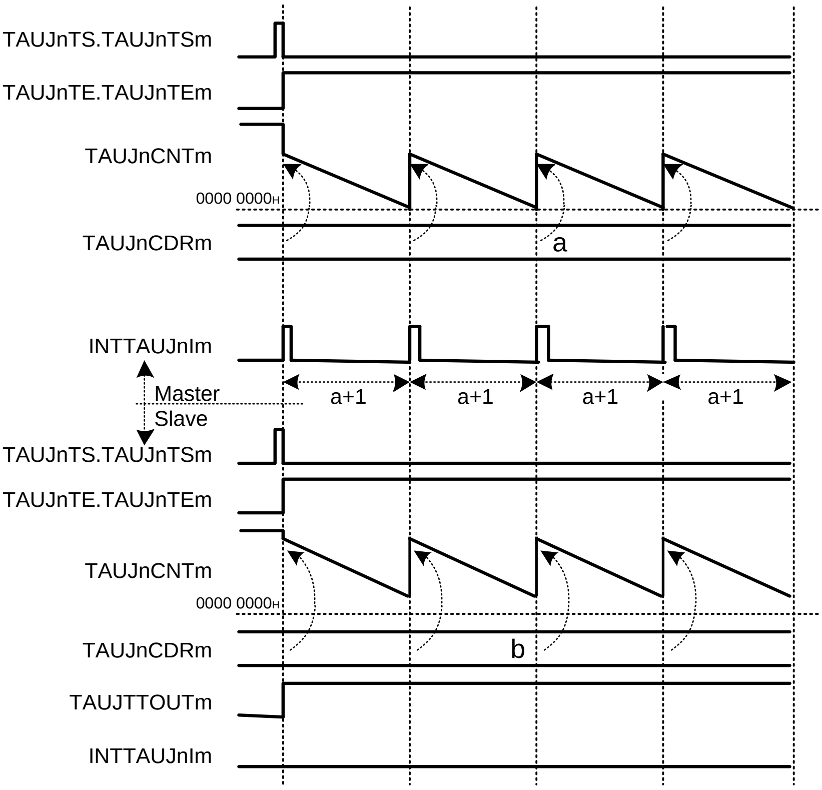


Figure 33.50 TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000_H. TAUJTTOUTm remains active.

(3) Operation Stop and Restart

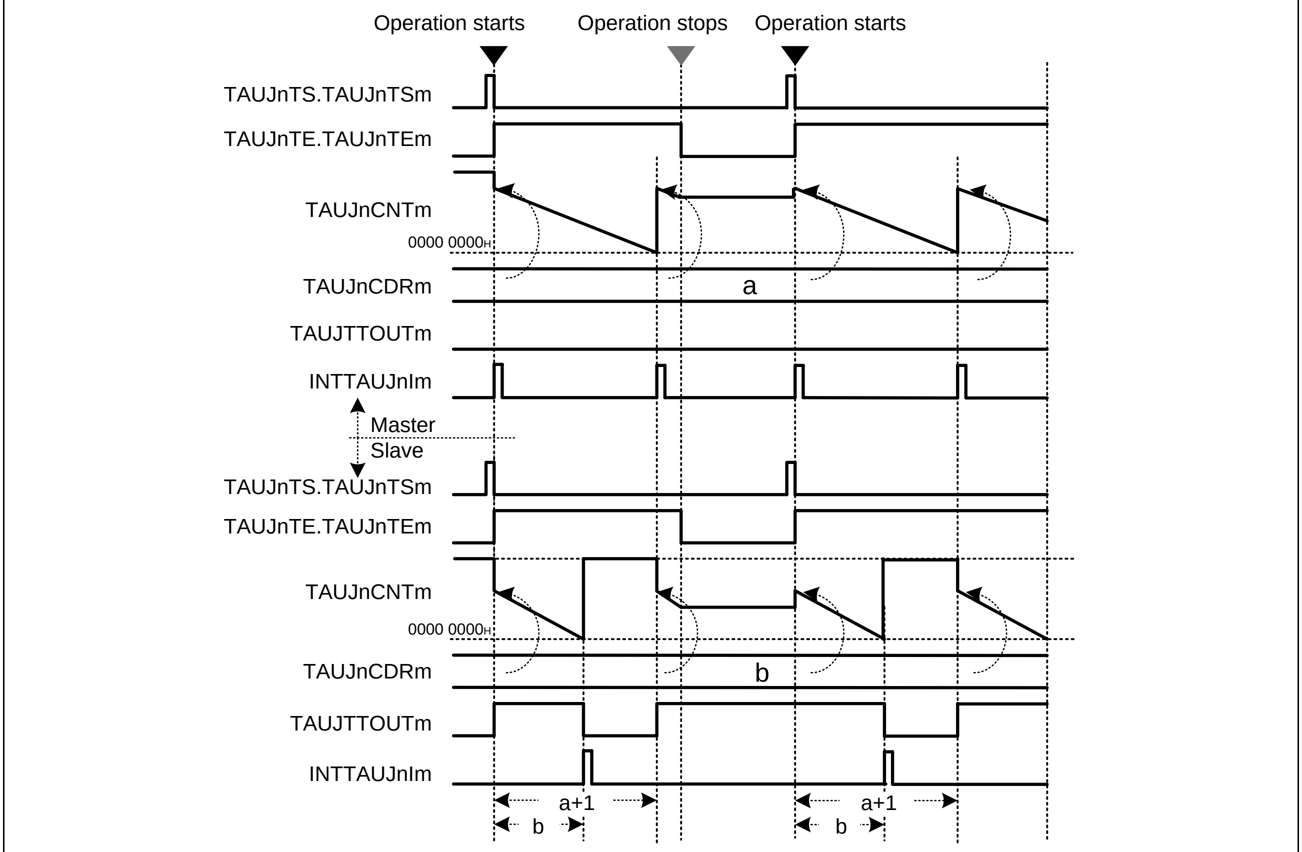


Figure 33.51 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. The TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

Section 34 Real-Time Clock (RTCA)

This section contains a generic description of the Real-Time Clock (RTCA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of RTCA.

34.1 Features of RH850/F1KH, RH850/F1KM RTCA

34.1.1 Number of Units

This microcontroller has the following number of RTCA units.

Table 34.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)

Table 34.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)

Table 34.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)	RTCAn (n = 0)

Table 34.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual RTCA units are identified by the index “n”; for example, RTCAnCTL0 (n = 0) is the RTCAn control register 0.

34.1.2 Register Base Address

RTCAn base address is listed in the following table.

RTCAn register addresses are given as an offset from the base address.

Table 34.5 Register Base Address (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<RTCA0_base>	FFE7 8000$_H$

34.1.3 Clock Supply

The RTCAn clock supply is shown in the following table.

Table 34.6 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
RTCAn	RTCATCKI	CKSCLK_ARTCA	Macro clock
	PCLK	CPUCLK_L	Module clock
	Register access clock	CPUCLK_L	Bus clock

34.1.4 Interrupt Requests

RTCAn interrupt requests are listed in the following table.

Table 34.7 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RTCA0			
INTRTCA01S	1-second interval interrupt	209	—
INTRTCA0AL	Alarm interrupt	210	—
INTRTCA0R	Fixed interval interrupt	211	—

34.1.5 Reset Sources

RTCAn reset sources are listed in the following table. RTCAn is initialized by these reset sources.

Table 34.8 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
RTCAn	Power-up reset (PURES)

34.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

Table 34.9 External Input/Output Signal (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
RTCA0		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT^{*1}

Note 1. RTCA0OUT is connected to TAUJ0. For details, see **Section 33, Timer Array Unit J (TAUJ)**.

CAUTION

When port P8_6 is used as RTCA0OUT, port P8_6 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$, Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

34.2 Overview

34.2.1 Functional Overview

The Real-Time Clock (RTCA) has the following features:

- Count clock selection from 32 kHz to 4 MHz
- Counters for years, months, day of the month, day of the week, hours, minutes, seconds, and a sub-counter. The calendar covers 99 years. Leap years are handled by hardware automatically.
- One Hz pulse output function
- Fixed interval interrupt function
- Alarm interrupt function
- Clock error correction function if a 32.768-kHz count clock is used

34.2.2 Block Diagram

The block diagram shows the main components of the RTCA.

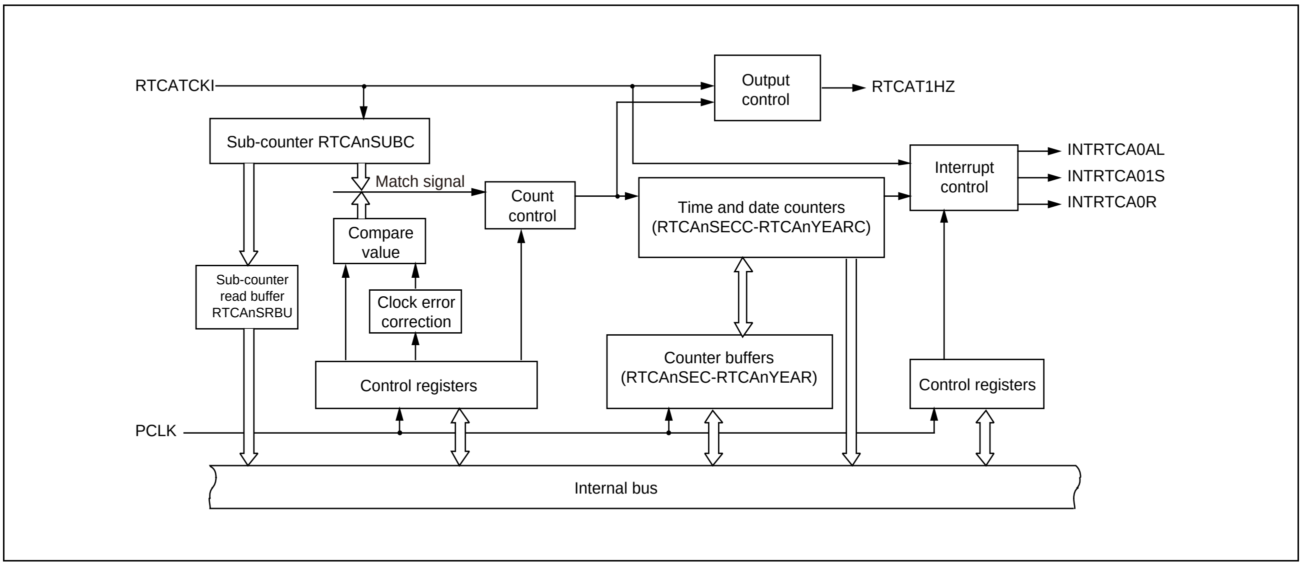


Figure 34.1 Block Diagram of the RTCA

34.2.3 Description of Blocks

The Real-Time Clock RTCA provides information about the present time and date and can generate wake-up signals (interrupts, alarms). This information is derived from the count clock RTCATCKI.

Sub-counter

RTCATCKI is the input to the sub-counter RTCAnSUBC. The sub-counter counts up from 0 until it reaches the compare value. The compare value is always defined as the frequency of RTCATCKI – 1 (in Hz). Thus, the sub-counter overflows after one second. It is then reset to 0 and triggers the seconds counter RTCAnSECC (and, if specified, the interrupt INTRTCA01S).

The sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5 seconds, or 1 second, and a 1-Hz output pulse.

Time and date counters

The counters for minutes, hours, day of the week, day of the month, months, and years also count up. They have their own overflow limits. If all the lower counters overflow, the upper counter counts up.

The overflow limit of the counter for the day of the month (RTCAnDAYC) depends on the present month (28, 30, or 31 days) and (in February) on the year counter RTCAnYEARC (years 0, 4, 8, 12, etc. are considered leap years).

The hours counter RTCAnHOURC can be switched between 12- and 24-hour formats.

The counters for seconds, minutes, hours, day of the month, and months can generate a fixed interval interrupt upon overflow (INTRTCA0R).

The counters for minutes, hours, and day of the week can also generate an alarm interrupt (INTRTCA0AL), e.g. every Tuesday and Thursday at 10:32.

Counter buffers

All counters can be read directly at any time. The clock signal used to access the read/write registers and the count clock are usually asynchronous. An overflow of the sub-counter during the read operation can make all read values obsolete. Therefore, reading the counters must be performed using a special procedure. For details, see **Section 34.5.3, Reading Clock Counters**.

For reasons of synchronization, the counters cannot be written directly.

For reading and writing, all counters are accompanied by buffer registers. The buffer registers provide a synchronized way for reading the counters and for setting time and date. When they are used, the operation of the sub-counter must first be suspended and then re-activated (see also **Section 34.5.3, Reading Clock Counters** and **Section 34.5.2, Updating Clock Counters**).

The RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to check and set the time (hours, minutes and seconds) or the date (day of the week, day of the month, month, and year) with one read/write operation.

34.3 Registers

34.3.1 List of Registers

RTCA registers are listed in the following table.

<RTCAn_base> is defined in **Section 34.1.2, Register Base Address**.

Table 34.10 List of Registers

Module Name	Register Name	Symbol	Address
RTCA control registers			
RTCAn	RTCA control register 0	RTCAnCTL0	<RTCAn_base> + 00_H
	RTCA control register 1	RTCAnCTL1	<RTCAn_base> + 04_H
	RTCA control register 2	RTCAnCTL2	<RTCAn_base> + 08_H
RTCA sub-counter registers			
RTCAn	RTCA sub-count register	RTCAnSUBC	<RTCAn_base> + $0C_H$
	RTCA sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10_H
	RTCA clock error correction register	RTCAnSUBU	<RTCAn_base> + 38_H
	RTCA sub-counter compare register	RTCAnSCMP	<RTCAn_base> + $3C_H$
RTCA clock counter and buffer registers			
RTCAn	RTCA seconds count register	RTCAnSECC	<RTCAn_base> + $4C_H$
	RTCA seconds count buffer register	RTCAnSEC	<RTCAn_base> + 14_H
	RTCA minutes count register	RTCAnMINC	<RTCAn_base> + 50_H
	RTCA minutes count buffer register	RTCAnMIN	<RTCAn_base> + 18_H
	RTCA hours count register	RTCAnHOURC	<RTCAn_base> + 54_H
	RTCA hours count buffer register	RTCAnHOUR	<RTCAn_base> + $1C_H$
	RTCA day of the week count register	RTCAnWEEKC	<RTCAn_base> + 58_H
	RTCA day of the week count buffer register	RTCAnWEEK	<RTCAn_base> + 20_H
	RTCA day of the month count register	RTCAnDAYC	<RTCAn_base> + $5C_H$
	RTCA day of the month count buffer register	RTCAnDAY	<RTCAn_base> + 24_H
	RTCA month count register	RTCAnMONC	<RTCAn_base> + 60_H
	RTCA month count buffer register	RTCAnMONTH	<RTCAn_base> + 28_H
	RTCA year count register	RTCAnYEARC	<RTCAn_base> + 64_H
	RTCA year count buffer register	RTCAnYEAR	<RTCAn_base> + $2C_H$
RTCA special counter and buffer registers			
RTCAn	RTCA time count register	RTCAnTIMEC	<RTCAn_base> + 68_H
	RTCA time count buffer register	RTCAnTIME	<RTCAn_base> + 30_H
	RTCA calendar count register	RTCAnCALC	<RTCAn_base> + $6C_H$
	RTCA calendar count buffer register	RTCAnCAL	<RTCAn_base> + 34_H
RTCA alarm time setting registers			
RTCAn	RTCA alarm minute setting register	RTCAnALM	<RTCAn_base> + 40_H
	RTCA alarm hour setting register	RTCAnALH	<RTCAn_base> + 44_H
	RTCA alarm day of the week setting register	RTCAnALW	<RTCAn_base> + 48_H
RTCA emulation register			
RTCAn	RTCA emulation register	RTCAnEMU	<RTCAn_base> + 74_H

34.3.2 Details of RTCA Control Registers

34.3.2.1 RTCAnCTL0 — RTCA Control Register 0

This register controls the count operation of the sub-counter RTCAnSUBC, the format (12-hour/24-hour) of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH, and the operation mode.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnCE	RTCAnCEST	RTCAnAMPM	RTCAnSLSB	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R	R

Table 34.11 RTCAnCTL0 Register Contents

Bit Position	Bit Name	Function
7	RTCAnCE	Starts/stops the sub-counter RTCAnSUBC operation. 0: Stops the sub-counter operation. All output pins and all status flags in control register RTCAnCTL2 are cleared. 1: Starts the sub-counter operation. The sub-counter counts up.
6	RTCAnCEST	Indicates the operation enabled/stopped status of the sub-counter: 0: Operation stopped status 1: Operation enabled status For details on how to use this status flag, see **Section 34.5.1, Initial Setting of the RTCA**.
5	RTCAnAMPM	Selects the format of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH: 0: 12-hour format (1 to 12, am/pm) 1: 24-hour format (0 to 23, military time) For details on the format, see **Table 34.23, 12- and 24-Hour Format**.
4	RTCAnSLSB	Selects the operation mode: 0: 32.768 kHz mode*[1] 1: Frequency selection mode For details on the operation modes, see **Section 34.4, Operation**. The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1). For details on the initialization of RTCAn, see **Section 34.5.1, Initial Setting of the RTCA**.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. This setting not supported in the RH850/F1KM-S4 100-pin product and RH850/F1KM-S1 all products.

34.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls the interrupt request generation and the 1-Hz pulse output.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnEN1HZ	RTCAnENALM	RTCAnEN1S	RTCAnCT[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.12 RTCAnCTL1 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>5</td><td>RTCAnEN1HZ</td><td>Enables/stops 1-Hz pulse output (RTCAT1HZ):
0: RTCAT1HZ disabled (RTCAT1HZ is fixed to 0)
1: RTCAT1HZ enabled</td></tr>
<tr><td>4</td><td>RTCAnENALM</td><td>Enables/disables alarm interrupt request generation (INTRTCA0AL):
0: INTRTCA0AL disabled
1: INTRTCA0AL enabled</td></tr>
<tr><td>3</td><td>RTCAnEN1S</td><td>Enables/disables 1-second interrupt request generation (INTRTCA01S):
0: INTRTCA01S disabled
1: INTRTCA01S enabled</td></tr>
<tr><td>2 to 0</td><td>RTCAnCT[2:0]</td><td>Specifies the fixed interval interrupt request (INTRTCA0R) setting:
<table>
<tr><th rowspan="2">RTCAnCT[2:0]</th><th colspan="2">INTRTCA0R Interrupt Request Generation</th></tr>
<tr><th>Interval</th><th>Timing</th></tr>
<tr><td>000</td><td colspan="2">No interrupt request generation</td></tr>
<tr><td>001</td><td>Every 0.25 seconds</td><td>Every 0.25, 0.5, 0.75 and 1 second</td></tr>
<tr><td>010</td><td>Every 0.5 seconds</td><td>Every 0.5 and 1 second</td></tr>
<tr><td>011</td><td>Every second</td><td>Every 1 second</td></tr>
<tr><td>100</td><td>Every minute</td><td>Every 1 minute 00 seconds</td></tr>
<tr><td>101</td><td>Every hour</td><td>Every 1 hour 00 minutes 00 seconds</td></tr>
<tr><td>110</td><td>Every day</td><td>Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)</td></tr>
<tr><td>111</td><td>Every month</td><td>Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)</td></tr>
</table>
If the settings of RTCAnCT[2:0] are changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to INTRTCA0R. Implement appropriate interrupt mask processing procedures.</td></tr>
</table>

34.3.2.3 RTCAnCTL2 — RTCA Control Register 2

This register contains status information and controls the data transfer from the sub-counter RTCAnSUBC to the dedicated sub-counter read buffer RTCAnSRBU and the operation setting of the clock counters (RTCAnSECC to RTCAnYEARC).

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnWUST	RTCAnWSST	RTCAnRSST	RTCAnRSUB	RTCAnWST	RTCAnWAIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W

Table 34.13 RTCAnCTL2 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	RTCAnWUST	Indicates whether RTCAnSUBU write operation has been completed: 0: RTCAnSUBU write completed 1: RTCAnSUBU write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSUBU is completed, this bit is set to 1. See **Section 34.5.5, Writing to RTCAnSUBU**, for details.
4	RTCAnWSST	Indicates whether RTCAnSCMP write operation has been completed: 0: RTCAnSCMP write completed 1: RTCAnSCMP write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSCMP is completed, this bit is set to 1. See **Section 34.5.6, Writing to RTCAnSCMP**, for details.
3	RTCAnRSST	Indicates whether the value of the sub-counter (RTCAnSUBC) has been transferred to the sub-count register read buffer (RTCAnSRBU): 0: Transfer in progress, or waiting for a transfer trigger 1: Transfer completed This bit is cleared (transfer is triggered) by RTCAnRSUB=1.This bit is automatically set when the transfer is completed. See **Section 34.5.4, Reading RTCAnSRBU**, for details.
2	RTCAnRSUB	Triggers transfer of the value of the sub-counter (RTCAnSUBC) to the dedicated read buffer (RTCAnSRBU) or clears the transfer state of the sub-counter: 0: Transfer status (RTCAnRSST) is cleared. 1: Transfer is triggered. This bit is used to read the value of RTCAnSRBU when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). The value of RTCAnSUBC is synchronized with RTCATCKI and loaded to RTCAnSRBU. For details, see **Section 34.5.4, Reading RTCAnSRBU**.

Table 34.13 RTCAnCTL2 Register Contents

Bit Position	Bit Name	Function
1	RTCAnWST	Indicates the status of all clock counters (RTCAnSECC to RTCAnYEARC): 0: All clock counters are running. 1: All clock counters are stopped The sub-counter is still running. The clock counters must be stopped before reading or writing clock counter values during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). To stop the clock counters, set RTCAnWAIT = 1.
0	RTCAnWAIT	Restarts/stops all clock counters (RTCAnSECC to RTCAnYEARC): 0: Restarts all clock counters either immediately or immediately after the clock counter write operation finishes. 1: Stops all clock counters temporarily. The sub-counter is still running. The clock counters must be stopped before reading or writing counter buffers during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). **CAUTION:** Only one overflow can be held internally. When two overflows occur, the seconds counter is incremented only by one when it is restarted. Thus, the procedure must be completed within one second.

34.3.3 Details of RTCA Sub-Counter Registers

34.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts the 1-second reference time. It operates using the count clock RTCATCKI.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<RTCAn_base> + 0C$_H$
Value after reset:	0000 0000$_H$

This register is initialized:

- When write operation is performed to the seconds count buffer register (RTCAnSEC) or to the time count buffer register (RTCAnTIME) and the value is reflected to the seconds count register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSUBC[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSUBC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.14 RTCAnSUBC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSUBC[21:0]	Sub-counter value The sub-counter only operates while RTCAnCTL0.RTCAnCEST = 1.

NOTES

1. This sub-counter operates with RTCATCKI while the read operation is clocked by PCLK. Reading this sub-counter during operation (RTCAnCTL0.RTCAnCEST = 1) is asynchronous to RTCATCKI and can lead to wrong results. Use the sub-count register read buffer (RTCAnSRBU) to read the sub-counter value during operation. For details, see **Section 34.5.4, Reading RTCAnSRBU**.
2. The count-operation of this sub-counter depends on the selected operation mode. See **Section 34.4, Operation**, for details.

34.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is the read buffer for the sub-counter RTCAnSUBC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSRBU[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSRBU[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.15 RTCAnSRBU Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSRBU[21:0]	Sub-counter value at the time of the last RTCAnSUBC read. When RTCAnCTL2.RTCAnRSUB is set to 1, the value of the RTCAnSUBC is loaded to the read buffer in synchronization with RTCATCKI.

NOTE

Perform RTCAnSRBU read according to the flow described in **Section 34.5.4, Reading RTCAnSRBU**.

34.3.3.3 RTCAnSUBU — RTCA Clock Error Correction Register

This register enables and specifies clock error correction. This register only applies in 32.768-kHz mode (RTCAnCTL0.RTCAnSLSB = 0).

For details on clock error correction, see **Section 34.4.4, Clock Error Correction**.

Access: This register can be read or written in 8-bit units.

Note the following when writing this register during sub-counter operation:

- Previous RTCAnSUBU write must be completed (RTCAnCTL2.RTCAnWUST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnDEV	RTCAnF6	RTCAnF[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.16 RTCAnSUBU Register Contents

Bit Position	Bit Name	Function
7	RTCAnDEV	Specifies how often clock error correction is performed per minute: 0: Three times every minute (when RTCAnSECC equals 00, 20, and 40) 1: Once every minute (when RTCAnSECC equals 00)
6	RTCAnF6	Specifies whether the sub-counter value is incremented or decremented: 0: Incremented (+ correction) Incrementation value = (RTCAnF[5:0] value – 1) × 2 1: Decremented (– correction) Decrementation value = (inverted data of RTCAnF[5:0] value + 1) × 2
5 to 0	RTCAnF[5:0]	Error correction value

NOTES

1. When RTCAnF[5:1] = 00000_B, clock error correction is not performed.
2. Perform RTCAnSUBU write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.5, Writing to RTCAnSUBU**.

34.3.3.4 RTCAnSCMP — RTCA Sub-Counter Compare Register

This register sets the compare value of the sub-counter RTCAnSUBC in frequency selection mode (RTCAnCTL0.RTCAnSLSB = 1).

When the sub-counter values matches the value of this register, an overflow signal is output to the seconds counter RTCAnSECC and the sub-counter is cleared.

Set the value for this register according to the frequency of the input clock RTCATCKI.

Access: This register can be read or written in 32-bit units.

Note the following when writing this register during sub-counter operation:

- Previous RTCAnSCMP write must be completed (RTCAnCTL2.RTCAnWSST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + $3C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSCMP[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.17 RTCAnSCMP Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 0	RTCAnSCMP[21:0]	Sub-counter compare value in frequency selection mode.

Example

The following example illustrates the setting of RTCAnSCMP:

- RTCATCKI = 4 MHz = 4,000,000 Hz
- RTCAnSCMP = 4,000,000 – 1 = 3,999,999 (decimal code) = $3D08FF_H$
- The seconds counter RTCAnSECC is triggered when the sub-counter value changes from $3D08FF_H$ to 0_H.

NOTES

1. The operation of the RTCA cannot be guaranteed if a value of 3198 (decimal code) or lower is set in this register.
2. Perform RTCAnSCMP write as described in **Section 34.5.1, Initial Setting of the RTCA** and **Section 34.5.6, Writing to RTCAnSCMP**.

34.3.4 Details of RTCA Clock Counter and Buffer Registers

34.3.4.1 RTCAnSECC — RTCA Seconds Count Register

This register is the seconds counter. It counts seconds from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the sub-counter RTCAnSUBC.

  If the sub-counter overflows while the seconds counter is stopped (RTCAnCTL2.RTCAnWST = 1), the seconds counter behaves as follows:

  - If one sub-counter overflow occurs while the seconds counter is stopped, the overflow is held internally.
    The seconds counter is incremented by one when it is restarted.
  - If two or more overflows occur while the seconds counter is stopped, the overflow count cannot be held internally.
    The seconds counter is incremented by one when it is restarted.
  - If the seconds counter was updated while the seconds counter is stopped, the sub-counter overflow(s) are ignored.
- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the minutes counter (RTCAnMINC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $4C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.18 RTCAnSECC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC[6:0]	Seconds in BCD

NOTES

1. Perform RTCAnSECC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the seconds count buffer register RTCAnSEC or to the clock time setting register RTCAnTIME. See

- **Section 34.5.1, Initial Setting of the RTCA**, and
- **Section 34.5.2, Updating Clock Counters**

34.3.4.2 RTCAnSEC — RTCA Seconds Count Buffer Register

This register is a buffer register to read/write the seconds counter RTCAnSECC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.19 RTCAnSEC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC[6:0]	Seconds in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnSEC read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.4.3 RTCAnMINC — RTCA Minutes Count Register

This register is the minutes counter. It counts minutes from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the seconds counter RTCAnSECC.
- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the hours counter (RTCAnHOURC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.20 RTCAnMINC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnMINC[6:0]	Minutes in BCD

NOTES

1. Perform RTCAnMINC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the minutes count buffer register RTCAnMIN or to the time count buffer register RTCAnTIME. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.4.4 RTCAnMIN — RTCA Minutes Count Buffer Register

This register is a buffer register to read/write the minutes counter RTCAnMINC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.21 RTCAnMIN Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnMIN[6:0]	Minutes in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnMIN read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.4.5 RTCAnHOURC — RTCA Hours Count Register

This register is the hours counter. It counts the hours in BCD. The count range depends on the selected hour format. See **Table 34.23, 12- and 24-Hour Format**.

This register counts as follows.

- It is triggered by every overflow of the minutes counter RTCAnMINC.
- It outputs an overflow signal when the value changes from 23 to 00 (in 24-hour format) or from 31 to 12 (in 12-hour format). The overflow signal triggers two counters:
  - Day of the week counter (RTCAnWEEKC)
  - Day of the month counter (RTCAnDAYC)

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 54_H

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R

Table 34.22 RTCAnHOURC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnHOURC[5:0]	Hours in BCD. See **Table 34.23, 12- and 24-Hour Format**, for details.

NOTES

1. Perform RTCAnHOURC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the hours count buffer register RTCAnHOUR or to the time count buffer register RTCAnTIME. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

12- or 24-hour format

The count values of RTCAnHOURC depend on the selected hour format.

If 12-hour format is selected (RTCAnCTL0.RTCAnAMPM = 0), bit 5 in the RTCAnHOURC register is the am/pm indicator:

- RTCAnHOURC[5] = 0: am
- RTCAnHOURC[5] = 1: pm

The following table shows the count range of RTCAnHOURC in both 12- and 24-hour format.

Table 34.23 12- and 24-Hour Format

12-Hour Format (RTCAnAMPM = 0)			24-Hour Format (RTCAnAMPM = 1)	
Time	RTCAnHOURC		Time	RTCAnHOURC
0 am	12_H		0	00_H
1 am	01_H		1	01_H
2 am	02_H		2	02_H
3 am	03_H		3	03_H
4 am	04_H		4	04_H
5 am	05_H		5	05_H
6 am	06_H		6	06_H
7 am	07_H		7	07_H
8 am	08_H		8	08_H
9 am	09_H		9	09_H
10 am	10_H		10	10_H
11 am	11_H		11	11_H
0 pm	32_H	↓	12	12_H
1 pm	21_H	pm indicator in 12-hour format: RTCAnHOUR.RTCAnHOUR[5] = 1	13	13_H
2 pm	22_H		14	14_H
3 pm	23_H		15	15_H
4 pm	24_H		16	16_H
5 pm	25_H		17	17_H
6 pm	26_H		18	18_H
7 pm	27_H		19	19_H
8 pm	28_H		20	20_H
9 pm	29_H		21	21_H
10 pm	30_H		22	22_H
11 pm	31_H		23	23_H

34.3.4.6 RTCAnHOUR — RTCA Hours Count Buffer Register

This register is a buffer register to read/write the hours counter RTCAnHOURC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $1C_H$

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.24 RTCAnHOUR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnHOUR[5:0]	Hours in BCD See **Table 34.23, 12- and 24-Hour Format**, for details.

NOTES

1. When writing this register, only the following decimal values in BCD are allowed:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0):
     01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1):
     00 to 23
2. Perform RTCAnHOUR read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.4.7 RTCAnWEEKC — RTCA Day of the Week Count Register

This register is the day of the week counter. It counts from 0 to 6.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.25 RTCAnWEEKC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC[2:0]	Day of the week

NOTES

1. Perform RTCAnWEEKC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the week count buffer register RTCAnWEEK or to the calendar count buffer register RTCAnCAL. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.4.8 RTCAnWEEK — RTCA Day of the Week Count Buffer Register

This register is a buffer register to read/write the day of the week counter RTCAnWEEKC.

There is no particular correspondence between the value of RTCAnWEEK and the day of the week. Set the correspondence according to the application to be used.

Example: 0 = Sunday, 1 = Monday, …, 6 = Saturday

Access:	This register can be read or written in 8-bit units.
Address:	<RTCAn_base> + 20_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 34.26 RTCAnWEEK Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAnWEEK[2:0]	Day of the week

NOTES

1. When writing this register, only decimal values between 0 and 6 in BCD are allowed.
2. Perform RTCAnWEEK read/write as described in
    - **Section 34.5.1, Initial Setting of the RTCA**,
    - **Section 34.5.2, Updating Clock Counters**, and
    - **Section 34.5.3, Reading Clock Counters**.

34.3.4.9 RTCAnDAYC — RTCA Day of the Month Count Register

This register is the day of the month counter. It counts from 01 to a maximum of 31 in BCD, depending on the value of the month counter (RTCAnMONC) and the year counter (RTCAnYEARC):

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, non-leap year)

Years 0, 4, 8, 12, etc., are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.
- It outputs an overflow signal when the value changes from 28, 29, 30, or 31 to 01, depending on the current month and year. The overflow signal triggers the month counter (RTCAnMONC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $5C_H$

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 34.27 RTCAnDAYC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnDAYC[5:0]	Day of the month in BCD

NOTES

1. Perform RTCAnDAYC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the month count buffer register RTCAnDAY or to the calendar count buffer register RTCAnCAL. See
    - **Section 34.5.1, Initial Setting of the RTCA**, and
    - **Section 34.5.2, Updating Clock Counters**.

34.3.4.10 RTCAnDAY — RTCA Day of the Month Count Buffer Register

This register is a buffer register to read/write the day of the month counter RTCAnDAYC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 24_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.28 RTCAnDAY Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnDAY[5:0]	Day of the month in BCD

NOTES

1. When writing this register, only decimal values between 01 and 31 in BCD are allowed:
   - 01 to 31 (January, March, May, July, August, October, December)
   - 01 to 30 (April, June, September, November)
   - 01 to 29 (February, leap year)
   - 01 to 28 (February, non-leap year)
2. Perform RTCAnDAY read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.4.11 RTCAnMONC — RTCA Month Count Register

This register is the month counter. It counts the month of the year, starting from 01 to 12 in BCD.

This register counts as follows.

- It is triggered by every overflow of the counter for the day of the month RTCAnDAYC.
- It outputs an overflow signal when the value changes from 12 to 01. The overflow signal triggers the year counter (RTCAnYEARC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 60_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 34.29 RTCAnMONC Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	RTCAnMONC[4:0]	Month of the year in BCD

NOTES

1. Perform RTCAnMONC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the month count buffer register RTCAnMONTH or to the calendar count buffer register RTCAnCAL. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.4.12 RTCAnMONTH — RTCA Month Count Buffer Register

This register is a buffer register to read/write the month counter RTCAnMONC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 28_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 34.30 RTCAnMONTH Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	RTCAnMONTH[4:0]	Month of the year in BCD

NOTES

1. When writing this register, only decimal values between 01 and 12 in BCD are allowed.
2. Perform RTCAnMONTH read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.4.13 RTCAnYEARC — RTCA Year Count Register

This register is the year counter. It counts years from 00 to a maximum of 99 in BCD.

Years 00, 04, 08, …, 92, and 96 (every four years) are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the month counter RTCAnMONC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEARC[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.31 RTCAnYEARC Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEARC[7:0]	Year in BCD

NOTES

1. Perform RTCAnYEARC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the year count buffer register RTCAnYEAR or to the calendar count buffer register RTCAnCAL. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.4.14 RTCAnYEAR — RTCA Year Count Buffer Register

This register is a buffer register to read/write the year counter RTCAnYEARC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEAR[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.32 RTCAnYEAR Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEAR[7:0]	Year in BCD

NOTES

1. When writing this register, only decimal values between 00 and 99 in BCD are allowed.
2. Perform RTCAnYEAR read/write as described in
   - **Section 34.5.1, Initial Setting of the RTCA**,
   - **Section 34.5.2, Updating Clock Counters**, and
   - **Section 34.5.3, Reading Clock Counters**.

34.3.5 Details of RTCA Special Counter and Buffer Registers

34.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables the RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 68_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]							—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.33 RTCAnTIMEC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	RTCAnHOURC[5:0]	Hours in BCD. See **Table 34.23, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned.
14 to 8	RTCAnMINC[6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC[6:0]	Seconds in BCD

NOTES

1. Perform RTCAnTIMEC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the time count buffer register RTCAnTIME. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.5.2 RTCAnTIME — RTCA Time Count Buffer Register

This register enables the RTCAnHOUR, RTCAnMIN, and RTCAnSEC buffer registers to be read/written simultaneously.

Access:	This register can be read or written in 32-bit units.
Address:	<RTCAn_base> + 30_H
Value after reset:	$0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]							—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.34 RTCAnTIME Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	RTCAnHOUR[5:0]	Hours in BCD See **Table 34.23, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	RTCAnMIN[6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC[6:0]	Seconds in BCD

NOTE

Perform RTCAnTIME read/write as described in

- **Section 34.5.1, Initial Setting of the RTCA**,
- **Section 34.5.2, Updating Clock Counters**, and
- **Section 34.5.3, Reading Clock Counters**.

34.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables the RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 6C$_H$

Value after reset: 0001 0100$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEARC[7:0]								—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]						—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.35 RTCAnCALC Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEARC[7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned.
20 to 16	RTCAnMONC[4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RTCAnDAYC[5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC[2:0]	Day of the week in BCD

NOTES

1. Perform RTCAnCALC read according to the flow described in **Section 34.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the clock time setting register RTCAnCAL. See
   - **Section 34.5.1, Initial Setting of the RTCA**, and
   - **Section 34.5.2, Updating Clock Counters**.

34.3.5.4 RTCAnCAL — RTCA Calendar Count Buffer Register

This register enables the RTCAnYEAR, RTCAnMONTH, RTCAnDAY, and RTCAnWEEK buffer registers to be read/written simultaneously.

Access:	This register can be read or written in 32-bit units.
Address:	<RTCAn_base> + 34_H
Value after reset:	$0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEAR[7:0]								—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]						—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 34.36 RTCAnCAL Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEAR[7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	RTCAnMONTH[4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RTCAnDAY[5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAnWEEK[2:0]	Day of the week in BCD

NOTE

Perform RTCAnCAL read/write as described in

- **Section 34.5.1, Initial Setting of the RTCA**,
- **Section 34.5.2, Updating Clock Counters**, and
- **Section 34.5.3, Reading Clock Counters**.

34.3.6 Details of RTCA Alarm Time Setting Registers

34.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies the minute of the alarm interrupt.

For details and example settings, see **Section 34.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 40_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALM[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.37 RTCAnALM Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALM[6:0]	Minute of the alarm interrupt in BCD

NOTES

1. If decimal values outside the range of 00 to 59 in BCD are set, no alarm interrupt request will be generated.
2. When the setting of RTCAnALM is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

34.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies the hour of the alarm interrupt.

For details and example settings, see **Section 34.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.
Address: <RTCAn_base> + 44_H
Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnALH[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.38 RTCAnALH Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnALH[5:0]	Hour of the alarm interrupt in BCD

NOTES

1. If decimal values outside the following range are set, no alarm interrupt request will be generated:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0): 01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1): 00 to 23
2. When the setting of RTCAnALH is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

34.3.6.3 RTCAnALW — RTCA Alarm Day of the Week Setting Register

This register specifies the day(s) of the week of the alarm interrupt.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 48_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALW[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.39 RTCAnALW Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALW[6:0]	Specifies day of the week m (m = 0 to 6) as a day, when an alarm interrupt request is generated: 0: No alarm interrupt request is generated on day m. 1: Alarm interrupt request is generated on day m at the time set using RTCAnALM and RTCAnALH. The bits of this register correspond to the count value of the day of the week counter (RTCAnWEEKC).

NOTE

When the setting of RTCAnALW is changed during sub-counter operation (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

Example

If Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

- To set the alarm for Sunday, set RTCAnALW = $0000\ 0001_B$.
- To set the alarm for Monday and Wednesday, set RTCAnALW = $0000\ 1010_B$.
- To set the alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = $0101\ 0100_B$.

For more examples, see **Section 34.4.3, Alarm Interrupt Function**.

34.3.7 RTCA Emulation Register

34.3.7.1 RTCAnEMU — RTCA Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 8-bit or 1-bit units.

A write should be performed when EPC.SVSTOP = 0.

Address: <RTCAn_base> + 74_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 34.40 RTCAnEMU Register Contents

Bit Position	Bit Name	Function
7	RTCAnSVSDIS	When the EPC.SVSTOP bit is set to 0: Count clock is supplied when the debugger gains microcontroller control (at a breakpoint, etc.) regardless of the value of this bit. When the EPC.SVSTOP bit is set to 1: 0: Count clock is stopped when the debugger gains microcontroller control (as at a breakpoint). 1: Count clock continues to be supplied when the debugger gains microcontroller control (at a breakpoint, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

34.4 Operation

The RTCA provides two operation modes:

- Frequency selection mode
- 32.768-kHz mode

The operation mode that can be used depends on the available input clock RTCATCKI. The operation mode specifies the sub-counter compare value that is used to trigger the seconds counter and thus all subsequent counters. Clock error correction is only possible in 32.768-kHz mode.

The following table provides an overview of the properties of the two operation modes.

Table 34.41 RTCA Operation Mode Overview

		32.768-kHz Mode	
	Frequency Selection Mode	Clock Correction Disabled	Clock Correction Enabled
Allowed input clock RTCATCKI	Any frequency from 32 kHz to 4 MHz	32.768 kHz	Any frequency from 32.76180000 kHz to 32.77420000 kHz
Sub-counter RTCAnSUBC operation	● Counter overflow at value of RTCAnSCMP ● RTCAnSCMP must be set to RTCATCKI-1 (in Hz)	Counter overflow at $7FFF_H$	Counter overflow at $7FFF_H$ or Every 20 or 60 seconds: $7FFF_H$ ±RTCAnSUBU.RTCAnF[5:0]

The operation mode is selected by control bit RTCAnCTL0.RTCAnSLSB. For details on how to set the operation mode during RTCA initialization, see **Section 34.5.1, Initial Setting of the RTCA**.

CAUTIONS

1. The input clock RTCATCKI must not be outside the allowed frequency range.
2. The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1).

34.4.1 Clock Counter Format

The clock counters (RTCAnSECC to RTCAnYEARC) operate on binary coded decimals (BCD): Each digit is represented by its own binary sequence.

Depending on the valid data range, the number of bits for a digit differs. For example, the tens digit of the month of the year counter has only one bit (for 0 and 1) whereas the tens digit of the minutes counter has 3 bits (for 0 to 5).

The following table lists the decimals 0 to 59 in binary and BCD.

Table 34.42 Example of BCD Code – Seconds or Minutes Counter (0 to 59)

Decimal	Binary	BCD
0	000000	000 0000
1	000001	000 0001
2	000010	000 0010
3	000011	000 0011
4	000100	000 0100
5	000101	000 0101
6	000110	000 0110
7	000111	000 0111
8	001000	000 1000
9	001001	000 1001
10	001010	001 0000
11	001011	001 0001
12	001100	001 0010
:	:	:
58	111010	101 1000
59	111011	101 1001

34.4.2 Fixed Interval Interrupt Function

Interrupt INTRTCA0R can be specified to occur after every 0.25 seconds, 0.5 seconds, 1 (full) second, 1 (full) minute, 1 (full) hour, 1 (full) day, or 1 (full) month.

The fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

34.4.3 Alarm Interrupt Function

Interrupt INTRTCA0AL can be specified to occur at a certain time on one or several days of the week. This interrupt can be used as a wake-up signal.

The alarm interrupt function is enabled and disabled by bit RTCAnCTL1.RTCAnENALM.

The alarm setting is specified by the following control registers:

- RTCAnALW selects the weekday(s).

The allocation of bits to weekdays is defined by the day of the week count buffer register RTCAnWEEK.

- RTCAnALH and RTCAnALM specify the hour and minute in BCD.

Examples

The following tables show some exemplary settings of the alarm control registers for both 12-hour and 24-hour format.

In this example, Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

Table 34.43 Alarm Setting in 12-Hour Format (RTCAnCTL0.RTCAnAMPM = 0)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00 am	01_H	07_H	00_H
Sunday, Monday 12:15 pm	03_H	32_H	15_H
Monday, Wednesday, Friday 5:30 pm	$2A_H$	25_H	30_H
Daily, 10:45 pm	$7F_H$	30_H	45_H

Table 34.44 Alarm Setting in 24-Hour Format (RTCAnCTL0.RTCAnAMPM = 1)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00	01_H	07_H	00_H
Sunday, Monday 12:15	03_H	12_H	15_H
Monday, Wednesday, Friday 17:30	$2A_H$	17_H	30_H
Daily, 22:45	$7F_H$	22_H	45_H

34.4.4 Clock Error Correction

CAUTION

This is not supported in the RH850/F1KM-S4 100-pin product and RH850/F1KM-S1 all products.

Clock error correction compensates for deviations of the oscillator from the nominal clock rate. With clock error correction input clock rates from 32.76180 kHz to 32.77420 kHz are possible.

The clock error correction function is only available in 32.768-kHz operation mode. In this operation mode, a nominal clock rate of 32.768 kHz is expected and the sub-counters overflow value is fixed to $7FFF_H$.

The following figures illustrate the clock error when the input clock rate deviates from the nominal clock.

RTCATCKI = 32.768 kHz

Figure 34.2, RTCATCKI = 32.768 kHz, No Clock Error Correction Required shows the timing diagram if RTCATCKI matches the nominal clock rate of 32.768 kHz. No clock error correction is required.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.768-kHz clock is exactly equal to one second.

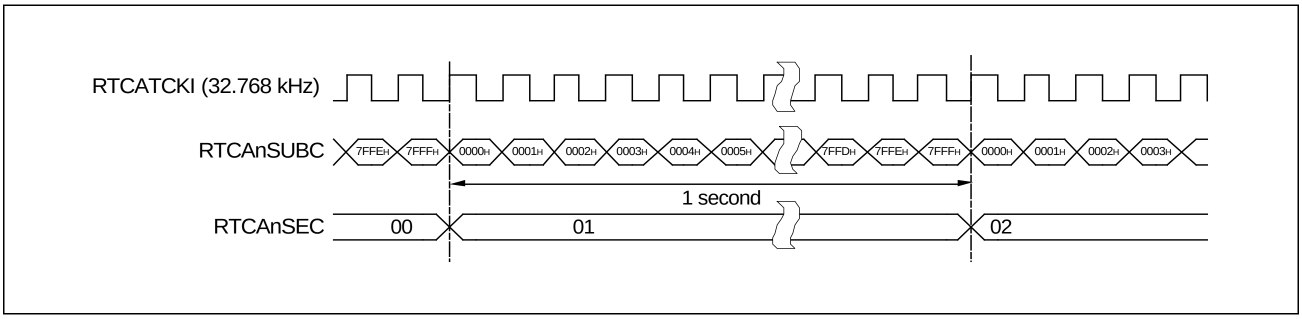


Figure 34.2 RTCATCKI = 32.768 kHz, No Clock Error Correction Required

RTCATCKI = 32.769 kHz

Figure 34.3, RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled shows the timing diagram if RTCATCKI deviates from the nominal clock rate of 32.768 kHz. In this example, RTCATCKI is connected to a 32.769-kHz oscillator. Clock error correction is not enabled.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.769-kHz clock is equal to approximately 0.99997 seconds (32768/32769). A "+ error" (faster than 32.768-kHz) occurs. In one month, RTCA deviates approximately –79 seconds from the real time.

Error = (32768/32769 - 1) × 60 (s) × 60 (min) × 24 (h) × 30 (d)

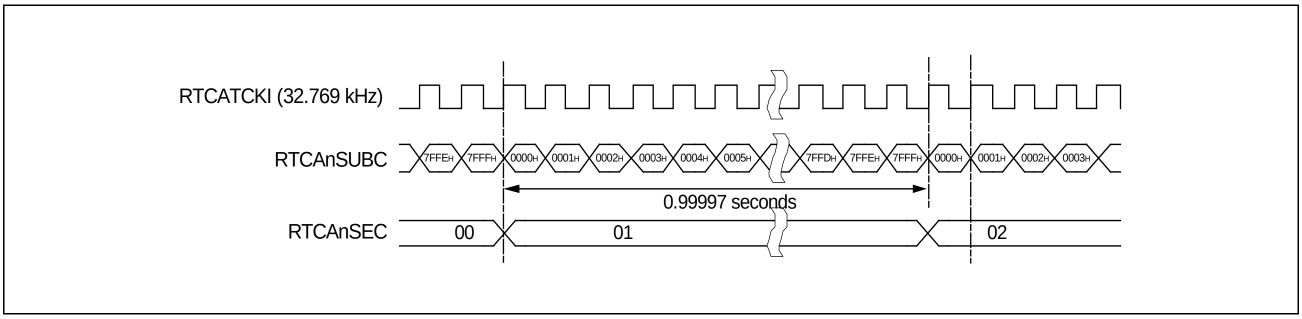


Figure 34.3 RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled

Clock error correction is performed by stretching/reducing the 1-second period of the sub-counter at regular intervals. The sub-counter's upper limit of $7FFF_H$ is increased or decreased by setting the following parameters in register RTCAnSUBU:

- A correction value greater than one
- An operator (add/subtract)
- An interval (20 or 60 seconds)

The corrected overflow value becomes effective every 20 or 60 seconds, so that on the average RTCAnSECC is triggered exactly every second.

34.4.4.1 Setting the Correction Value and the Operator

The correction value and operator are specified by the RTCAnF6, RTCAnF[5:0] bits of the RTCAnSUBU register:

- RTCAnF6 specifies whether the overflow value is incremented or decremented.
- RTCAnF[5:0] specifies the correction value.

The correction values are calculated as follows:

Table 34.45 Correction Value Settings

RTCAnF6	Increment/Decrement	Correction Value
0	Increment	(Value of RTCAnF[5:0] – 1) × 2
1	Decrement	(Inverted value of RTCAnF[5:0] + 1) × 2

Some examples are given in the following table:

Table 34.46 Correction Value Examples

RTCAnF6	RTCAnF[5:0]	Correction Value	Count Limit of RTCAnSUBC
0	15_H	$(15_H – 1) × 2 = 40$	32768 + 40 = 32808
1	15_H	$(\overline{15_H} + 1) × 2$ $= (2A_H + 1) × 2$ $= 86$	32768 – 86 = 32682

34.4.4.2 Impact of the Repetition Interval

The correction value set by RTCAnF6, RTCAnF[5:0] does not change the count limit of RTCAnSUBC every second. The repetition interval at which the correction value becomes effective is specified by bit RTCAnDEV.

This bit also influences the size of the correctable frequency range and the correction accuracy.

The following table summarizes the RTCAnDEV settings.

Table 34.47 Setting of Bit RTCAnSUBU.RTCAnDEV

RTCAnDEV	Count Limit of RTCAnSUBC is Changed	Frequency Range that can be Corrected	Correction Accuracy
0	Every 20 seconds when RTCAnSECC = 00, 20, or 40	32.76180000 to 32.77420000 kHz	
1	Every 60 seconds when RTCAnSECC = 00	32.76593333 to 32.77006667 kHz	Three times higher than for RTCAnDEV = 0

34.4.4.3 Sample Settings

The frequencies that can be corrected, as well as the setting values of bits RTCAnDEV, RTCAnF6, and RTCAnF[5:0], are listed in the following table.

Table 34.48 Correctable Frequency Range when RTCAnDEV = 0

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76810000 kHz	0	000010	Once every 20 s, RTCAnSUBC count value + 2
32.76820000 kHz	0	000011	Once every 20 s, RTCAnSUBC count value + 4
32.76830000 kHz	0	000100	Once every 20 s, RTCAnSUBC count value + 6
:	:	:	:
32.77400000 kHz	0	111101	Once every 20 s, RTCAnSUBC count value + 120
32.77410000 kHz	0	111110	Once every 20 s, RTCAnSUBC count value + 122
32.77420000 kHz (upper limit)	0	111111	Once every 20 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76180000 kHz (lower limit)	1	000010	Once every 20 s, RTCAnSUBC count value – 124
32.76190000 kHz	1	000011	Once every 20 s, RTCAnSUBC count value – 122
32.76200000 kHz	1	000100	Once every 20 s, RTCAnSUBC count value – 120
:	:	:	:
32.76770000 kHz	1	111101	Once every 20 s, RTCAnSUBC count value – 6
32.76780000 kHz	1	111110	Once every 20 s, RTCAnSUBC count value – 4
32.76790000 kHz	1	111111	Once every 20 s, RTCAnSUBC count value – 2

Table 34.49 Correctable Frequency Range when RTCAnDEV = 1

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76803333 kHz	0	000010	Once every 60 s, RTCAnSUBC count value + 2
32.76806667 kHz	0	000011	Once every 60 s, RTCAnSUBC count value + 4
32.76810000 kHz	0	000100	Once every 60 s, RTCAnSUBC count value + 6
:	:	:	:
32.77000000 kHz	0	111101	Once every 60 s, RTCAnSUBC count value + 120
32.77003333 kHz	0	111110	Once every 60 s, RTCAnSUBC count value + 122
32.77006667 kHz (upper limit)	0	111111	Once every 60 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76593333 kHz (lower limit)	1	000010	Once every 60 s, RTCAnSUBC count value – 124
32.76596667 kHz	1	000011	Once every 60 s, RTCAnSUBC count value – 122
32.76600000 kHz	1	000100	Once every 60 s, RTCAnSUBC count value – 120
:	:	:	:
32.76790000 kHz	1	111101	Once every 60 s, RTCAnSUBC count value – 6
32.76793333 kHz	1	111110	Once every 60 s, RTCAnSUBC count value – 4
32.76796667 kHz	1	111111	Once every 60 s, RTCAnSUBC count value – 2

34.5 Procedures for Setup, Writing and Reading

The following subsections provide flow charts that illustrate the procedures for RTCA setup and for reading and writing the RTCA clock counters.

34.5.1 Initial Setting of the RTCA

The RTCA must be stopped before setting the initial setting value of each counter.

34.5.1.1 RTCA Stop Procedure

Stop the RTCA according to the following flow.

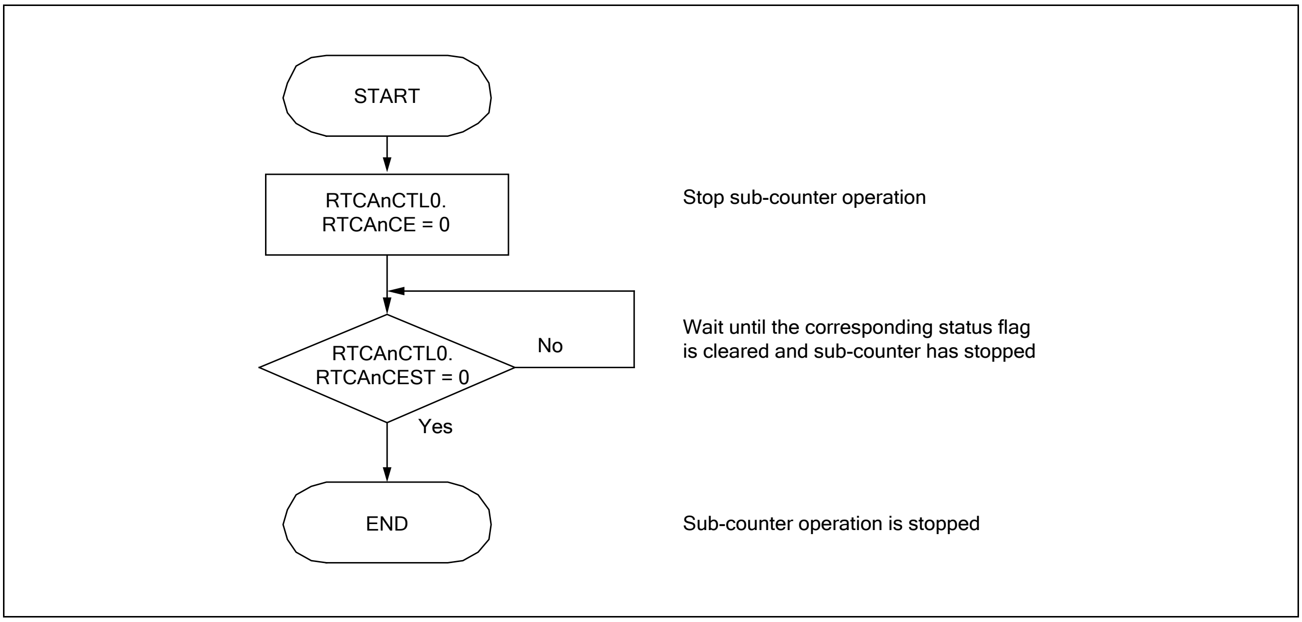


Figure 34.4 RTCA Stop Procedure

34.5.1.2 RTCA Initialization Procedure

Perform the initial setting of the RTCA according to the following flow:

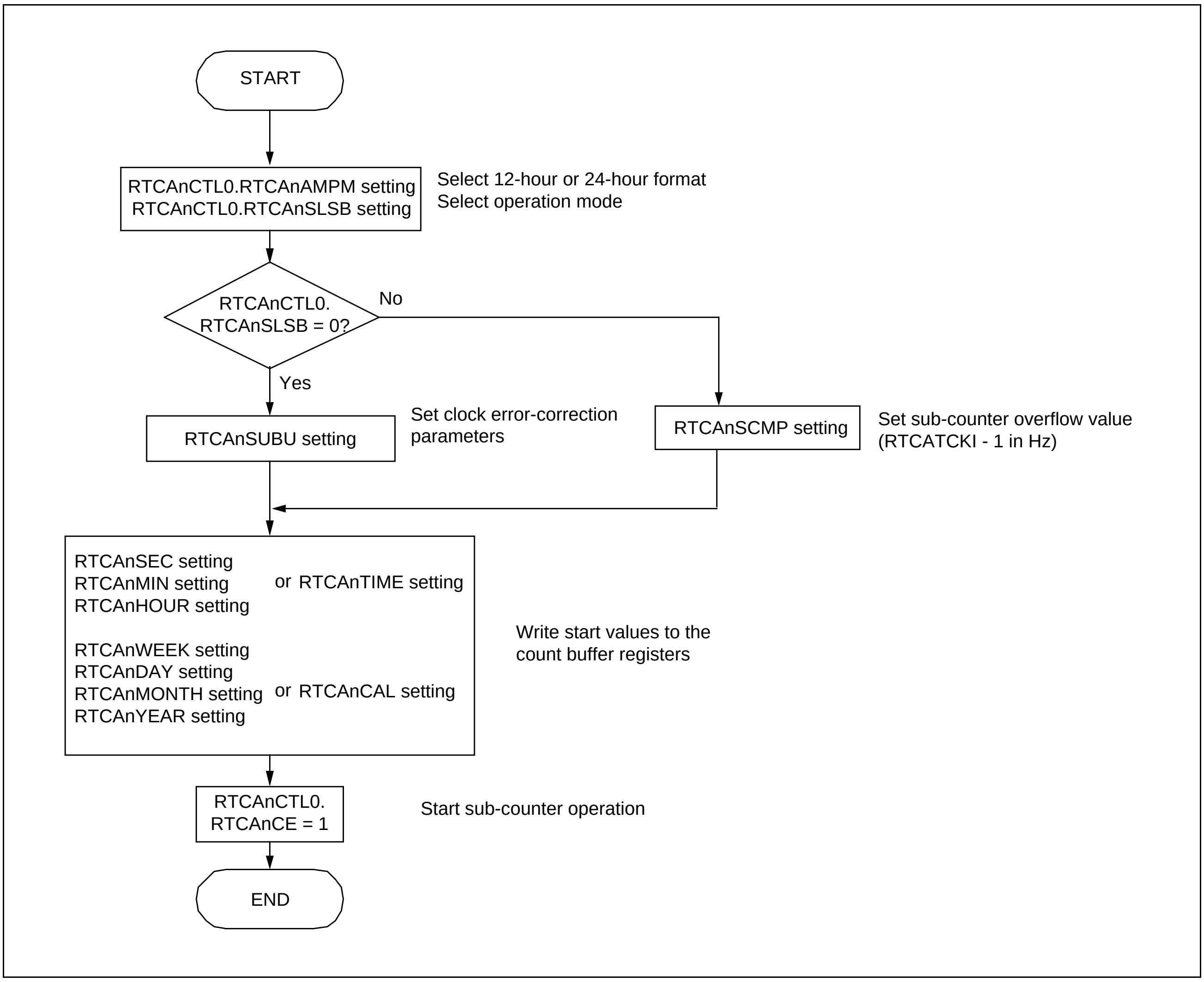


Figure 34.5 RTCA Initial Setup Procedure

CAUTION

The internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required before the clock counter starting behind END of the above flow.

Therefore, PCLK must be continuously supplied until the completion of the initial setting.

Check that RTCAnCTL0.RTCAnCEST = 1, when the supply of PCLK is stopped after setting the initial setting value of RTCA.

34.5.2 Updating Clock Counters

The clock counters RTCAnSECC to RTCAnYEARC can be stopped and updated while the sub-counter is running.

To update the clock counter when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

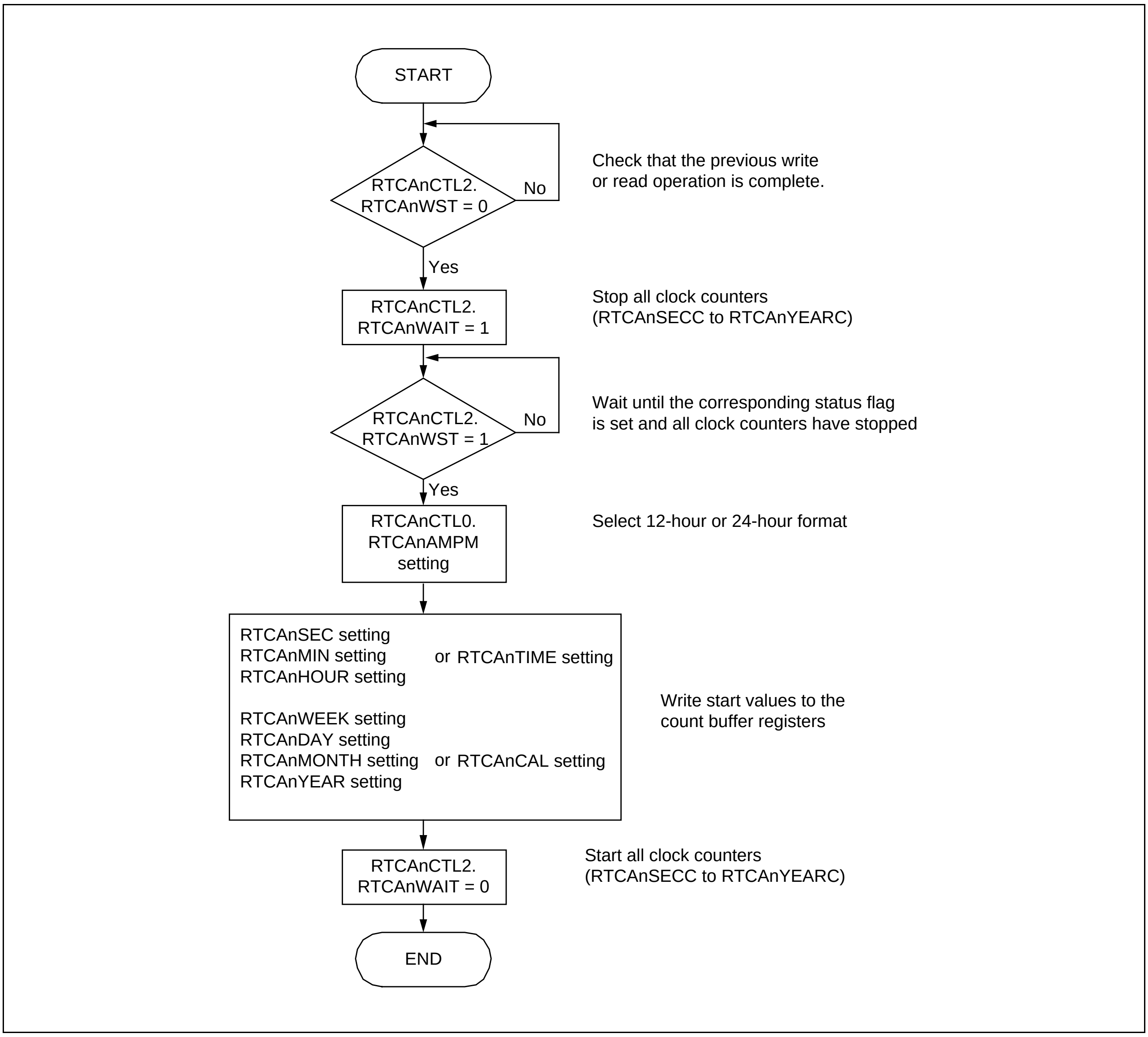


Figure 34.6 Updating Clock Counter Values

CAUTIONS

1. The internal clock counter is synchronized with RTCATCKI.
   In addition, two RTCATCKI periods are required before the clock counter updating behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the completion of the clock counter updating.
   Check that RTCAnCTL2.RTCAnWST = 0 before stopping the supply of PCLK after the completion of clock counter updating.
2. The update procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more:

3. Only one sub-counter overflow can be held internally and increment the seconds counter after restarting the clock counters if the value is held.
4. If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.

34.5.3 Reading Clock Counters

There are two methods to read the clock counters while sub-counter operation is enabled:

- Reading count buffer registers
- Reading counter registers

The advantages and disadvantages of the two methods are summarized in the following table.

Table 34.50 Comparison of the Two Read Methods

	Advantage	Disadvantage
Reading count buffer registers	It is unnecessary to read clock counters several times because the clock counters are read synchronously.	A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.
Reading count registers	Program wait state does not occur.	If the sub-counter increments, the clock counters must be read several times because they are read asynchronously to RTCATCKI.

34.5.3.1 Procedure for Reading Count Buffer Registers

The following operations are necessary:

1. Stop all clock counters (RTCAnCTL2.RTCAnWAIT = 1). The value of the clock counters is transferred to the corresponding count buffer registers.
2. Read the count buffer registers.

A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.

The maximum delay is three PCLK periods plus two RTCATCKI periods. For example, if the RTCA operates with PCLK = 40 MHz and RTCATCKI = 32.768 kHz, the delay is about 61 µs.

To read the count buffer register when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1), follow the flowchart shown below.

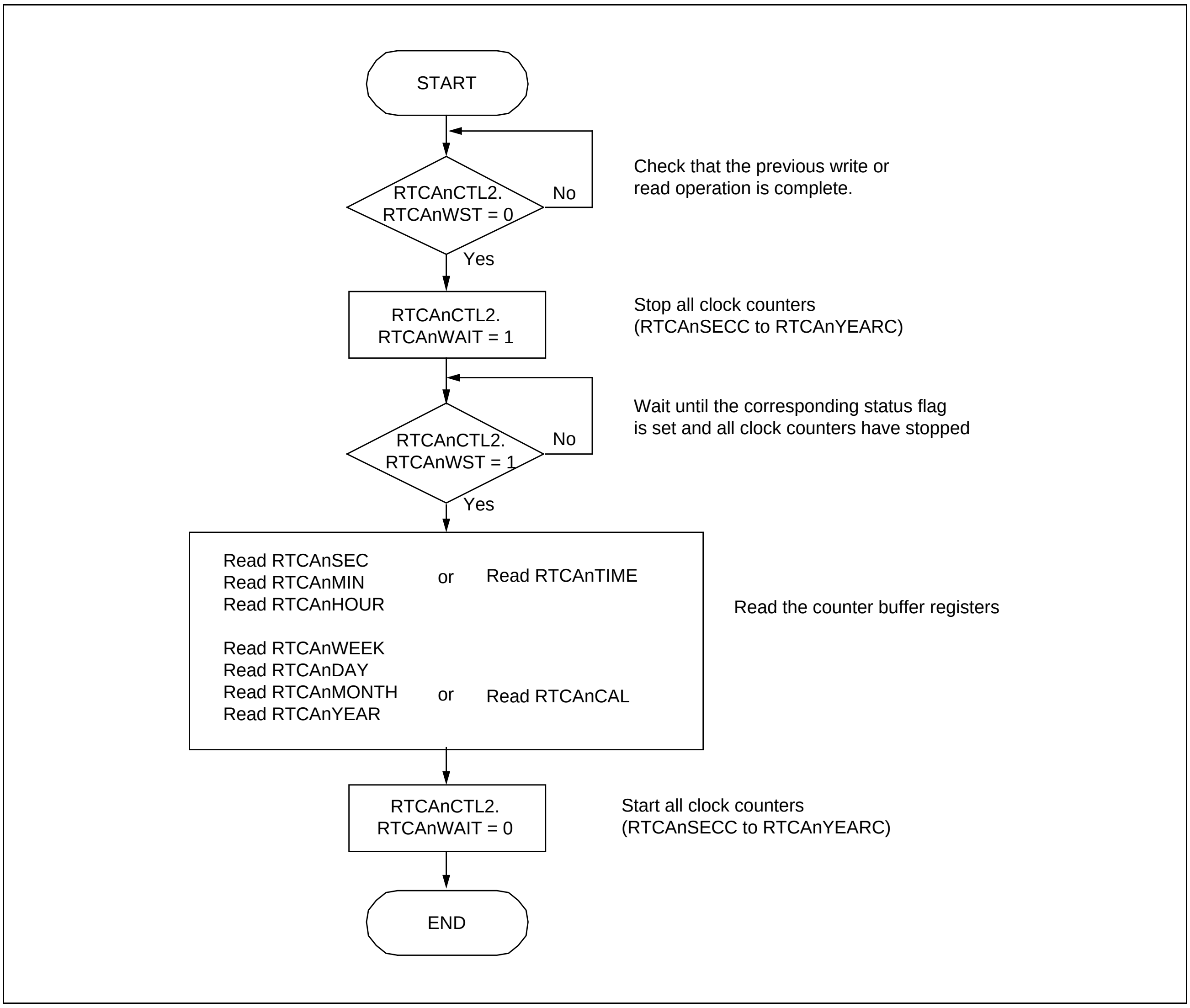


Figure 34.7 Reading Clock Count Buffer Registers

CAUTIONS

1. The internal clock counter is synchronized with RTCATCKI.
   In addition, two RTCATCKI periods are required before resuming counter behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the counter resuming.
   Check that RTCAnCTL0.RTCAnCEST = 1 first to stop the supply of PCLK after count buffer register reading.
2. The reading procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more.
3. Only one sub-counter overflow can be held internally. If there is a value held internally when the clock counter restarts, the seconds counter will be incremented by 1.
4. If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.

34.5.3.2 Procedure for Reading Counter Registers Directly

To ensure that the sub-counter did not overflow while reading the counters, the seconds counter RTCAnSECC must be read twice in the beginning and at the end of the procedure. The first read value is compared with the second read value.

- First read value = second read value:
  No overflow of sub-counter occurred during counter read operation.
- First read value ≠ second read value:
  Overflow of the sub-counter occurred during counter read operation. The counters must be read again to get the current counter values.

To read the counter register directly when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

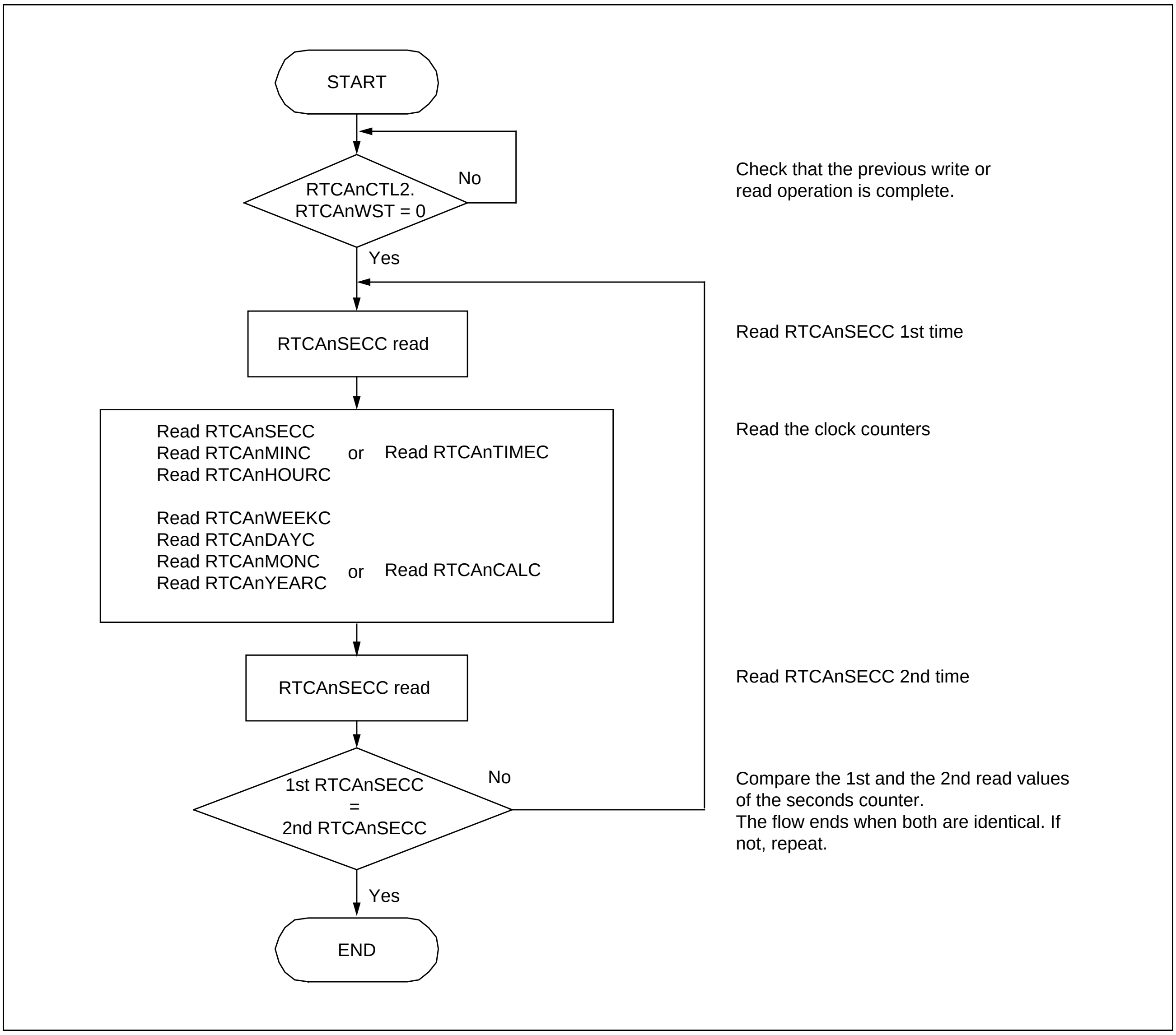


Figure 34.8 Reading Clock Counter Registers

NOTE

The procedure must be completed within one second.

34.5.4 Reading RTCAnSRBU

RTCAnSRBU is the read buffer register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), read RTCAnSRBU according to the following flow.

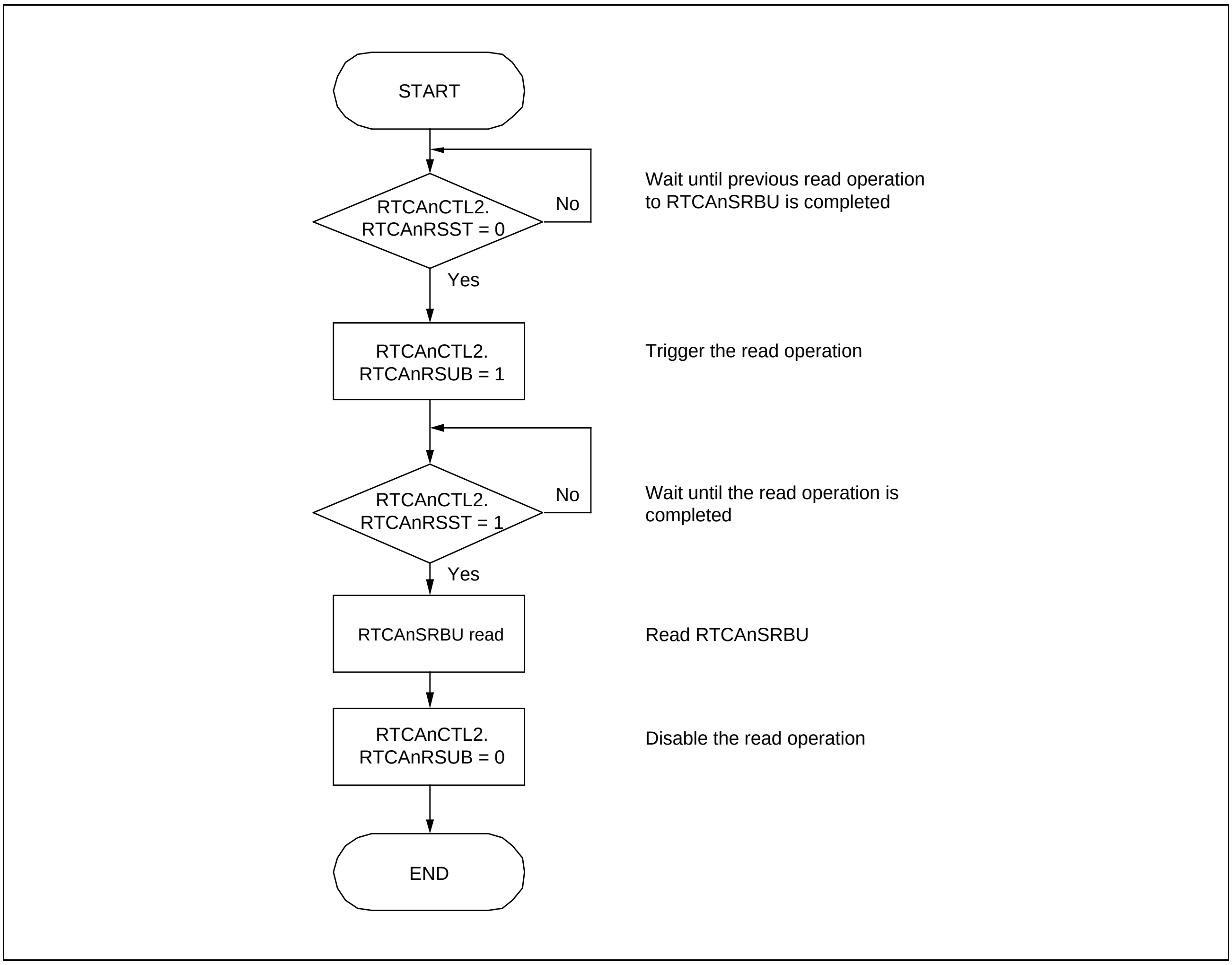


Figure 34.9 Reading the RTCAnSRBU Register

34.5.5 Writing to RTCAnSUBU

RTCAnSUBU is the clock error correction register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSUBU according to the flow described below.

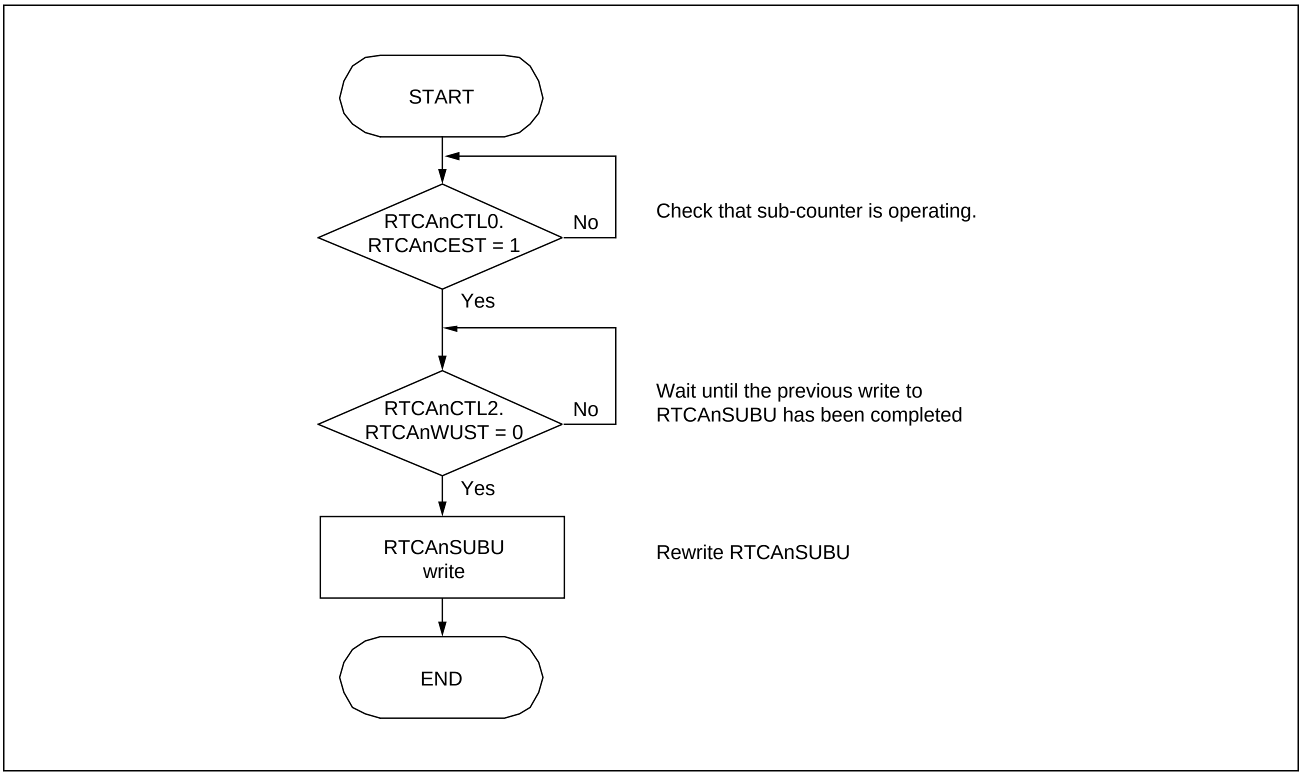


Figure 34.10 Writing to the RTCAnSUBU Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWUST is set when RTCAnSUBU is written to. It is cleared when the write operation to RTCAnSUBU is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWUST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWUST = 1 at the beginning of this flow).

34.5.6 Writing to RTCAnSCMP

RTCAnSCMP is the sub-counter compare register.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSCMP according to the flow described below.

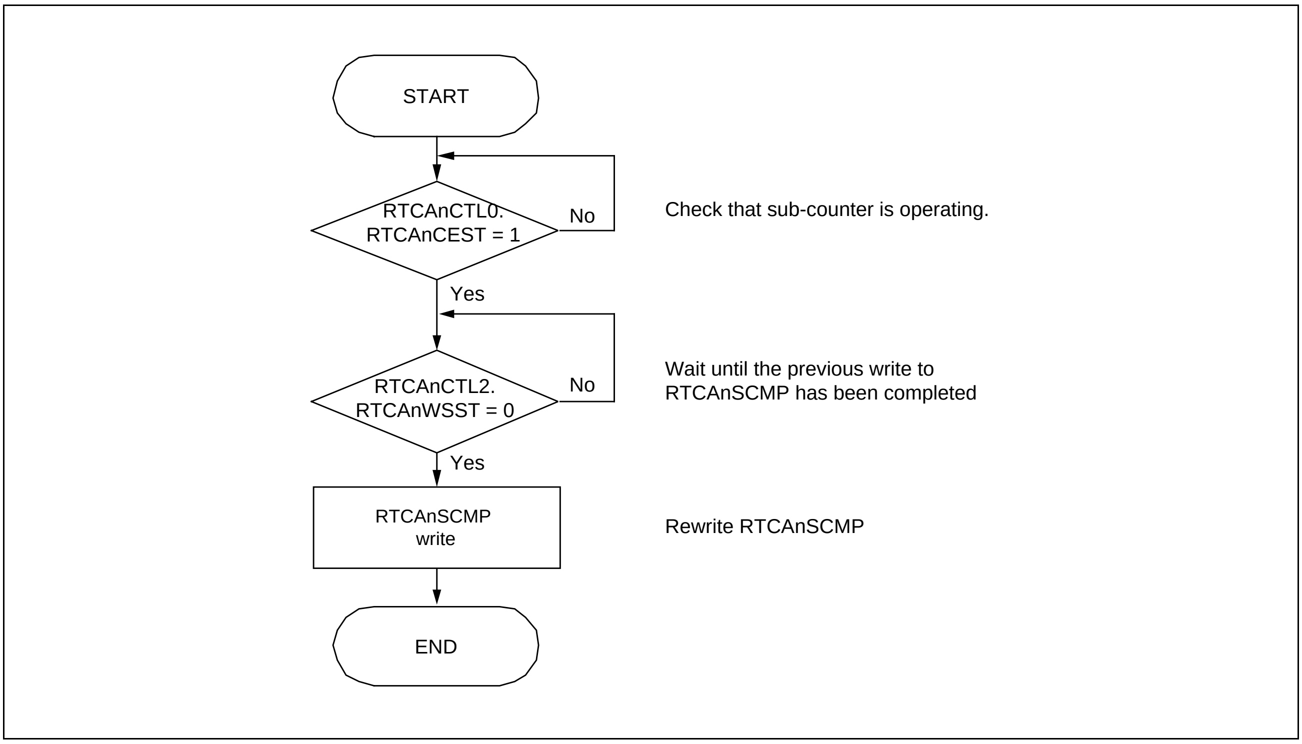


Figure 34.11 Writing the RTCAnSCMP Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWSST is set when RTCAnSCMP is written to. It is cleared when the write operation to RTCAnSCMP is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWSST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWSST = 1 at the beginning of this flow).

34.6 Timing Diagrams

34.6.1 Timing of Counter Start

The following diagram illustrates the counter start after setting the time in the buffer registers.

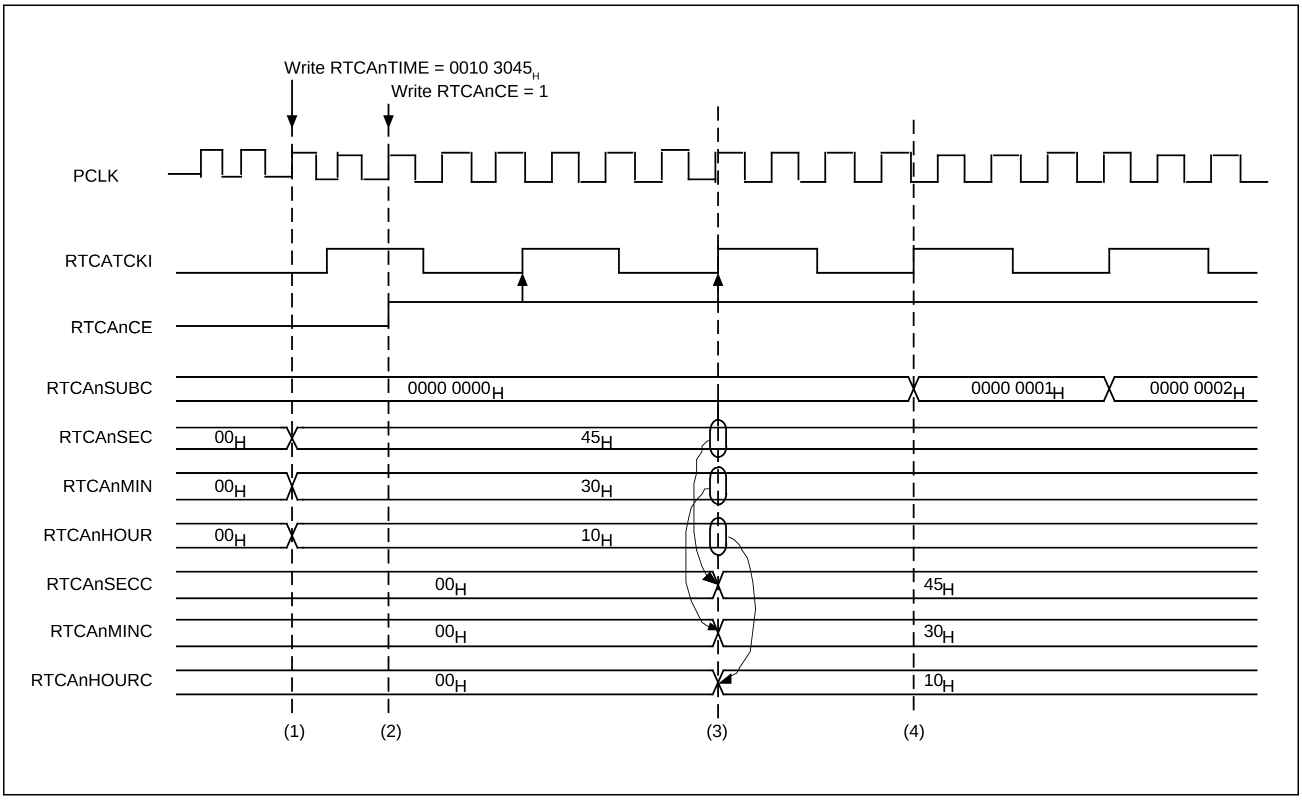


Figure 34.12 Counter Start Timing

The timing diagram above shows the following:

(1) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME = 0010 3045_H. Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(2) Sub-counter operation is started by setting RTCAnCTL0.RTCAnCE = 1.

(3) When the second rising edge of RTCATCKI occurs, the buffer register values are loaded to the corresponding count registers.

(4) When the next rising edge of RTCATCKI occurs, count up of the sub-counter starts.

34.6.2 Timing of Clock Counter Update while Counter Is Enabled

The following diagram illustrates the counter restart after setting the time in the buffer registers.

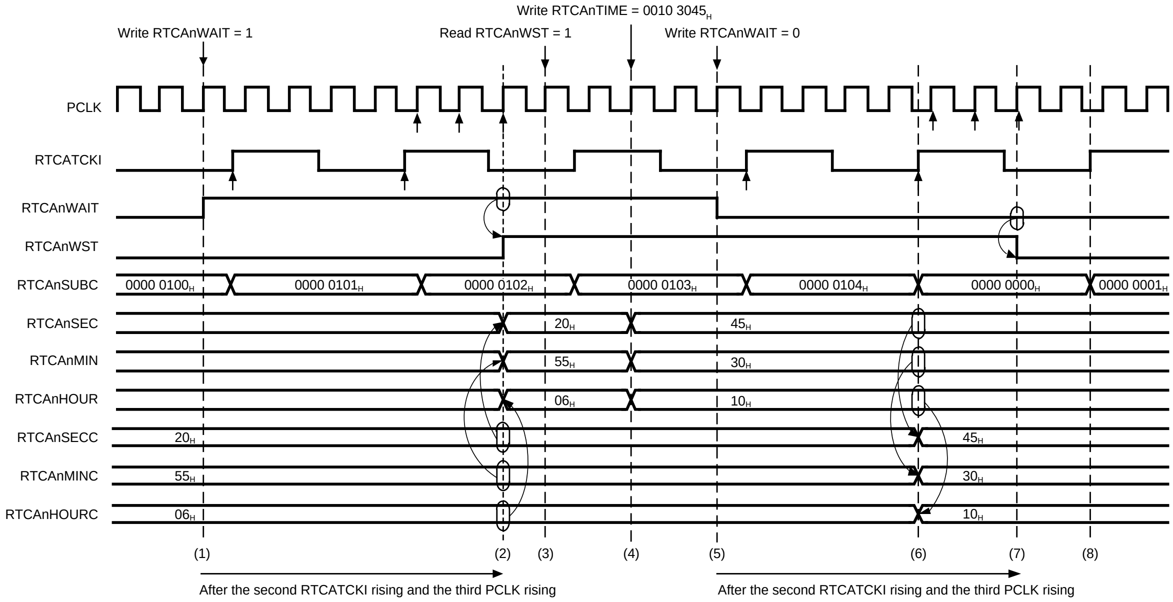


Figure 34.13 Clock Counter Update Timing

The timing diagram above shows the following:

(1) Trigger the clock counters stop (RTCAnCTL2.RTCAnWAIT = 1).

(2) RTCAnCTL2.RTCAnWST is set to 1 after the second rising edge of RTCATCKI and the third rising edge of PCLK, and the counter clock stops. The sub-counter continues counting.

(3) RTCAnCTL2.RTCAnWST = 1 can be readable.

(4) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME to 0010 3045_H. Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(5) Trigger the clock counters restart (RTCAnCTL2.RTCAnWAIT = 0).

(6) When the second rising edge of RTCATCKI occurs, the values of the buffer registers are loaded to the corresponding count registers. Write operation to RTCAnSECC is performed and RTCAnSUBC is cleared.

(7) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnWST is set to 0.

(8) Clock counter operation is resumed.

34.6.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled

The following diagram illustrates the timing when reading the sub-counter read buffer RTCAnSRBU.

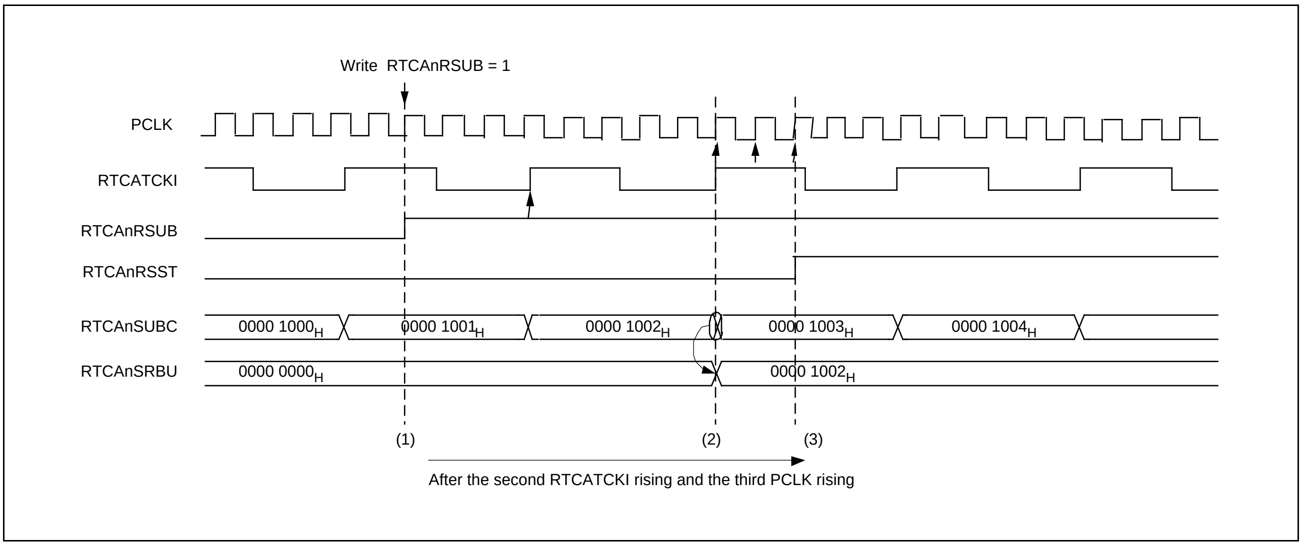


Figure 34.14 Timing when Reading the Sub-Counter Read Buffer Register Value

The timing diagram above shows the following:

(1) Setting RTCAnRSUB = 1 triggers loading of the sub-counter value to RTCAnSRBU.

(2) When the second rising edge of RTCATCKI occurs, the value of RTCAnSUBC is loaded to RTCAnSRBU.

(3) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnRSST is set to 1 and RTCAnSRBU can be read.

Section 35 Encoder Timer (ENCA)

This section contains a generic description of the Encoder Timer (ENCA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of ENCA.

35.1 Features of RH850/F1KH, RH850/F1KM ENCA

35.1.1 Number of Units

This microcontroller has the following number of ENCA units.

Table 35.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)

Table 35.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)

Table 35.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)	ENCAn (n = 0)

Table 35.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual ENCA units are identified by the index “n”; for example, ENCAnCTL (n = 0) is the ENCAn control register.

35.1.2 Register Base Address

ENCAn base address is listed in the following table.

ENCAn register addresses are given as an offset from the base address.

Table 35.5 Register Base Address (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<ENCA0_base>	FFE8 0000_H

35.1.3 Clock Supply

The ENCAn clock supply is shown in the following table.

Table 35.6 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
ENCAn	PCLK	CKSCLK_IPERI1	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock

35.1.4 Interrupt Requests

ENCAn interrupt requests are listed in the following table.

Table 35.7 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
ENCA0			
ENCATIOV	Overflow interrupt	85	—
ENCATIUD	Underflow interrupt	86	—
ENCATINT0	Capture/compare match interrupt 0	87	—
ENCATINT1	Capture/compare match interrupt 1	88	—
ENCATIEC	Encoder clear interrupt	89	—

35.1.5 Reset Sources

ENCAn reset sources are listed in the following table. ENCAn is initialized by these reset sources.

Table 35.8 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
ENCAn	All reset sources (ISORES)

35.1.6 External Input/Output Signals

External input/output signals of ENCAn are listed below.

Table 35.9 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
ENCA0		
ENCATTIN0	ENCAn capture trigger input 0*[1]	ENCA0TIN0
ENCATTIN1	ENCAn capture trigger input 1*[1]	ENCA0TIN1
ENCAnE0	ENCAn encoder input 0*[1]	ENCA0E0
ENCAnE1	ENCAn encoder input 1*[1]	ENCA0E1
ENCAnEC	ENCAn encoder clear input*[1]	ENCA0EC

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector** and **Section 2C.12, Noise Filter & Edge/Level Detector**.

35.1.7 Internal Input/Output Signals

Input/output signals to be connected between ENCA and PIC are listed below.

Table 35.10 Internal Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
ENCATSST	Simultaneous start trigger	PIC
ENCATTIN1	ENCAn capture trigger input 1	PIC

35.2 Overview

35.2.1 Functional Overview

- Generation of the counter control signal from the encoder input signal, and count operation in synchronization with PCLK
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow and output of error flags and error occurrence interrupts
- Five interrupts: two capture/compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt

35.2.2 Block Diagram

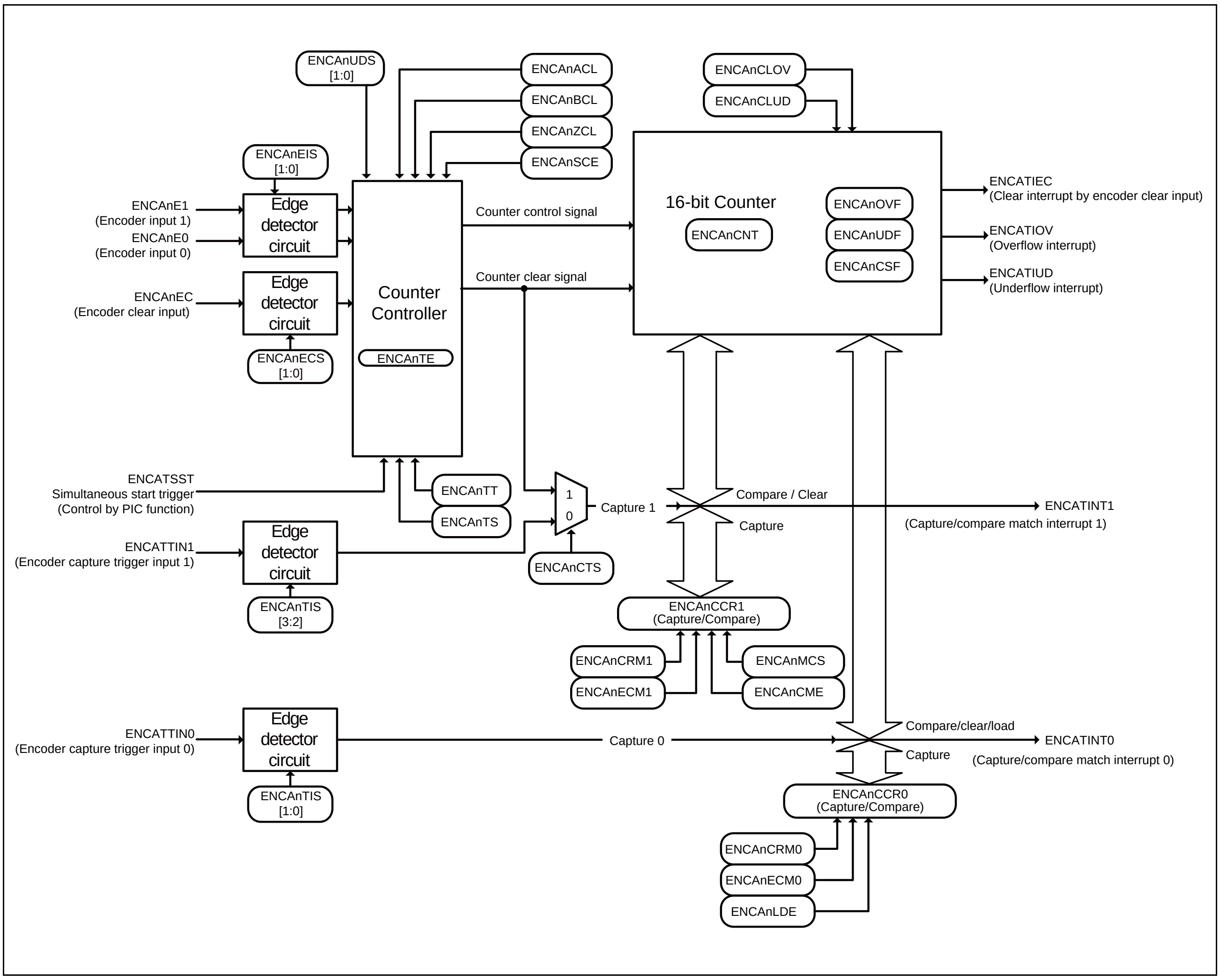


Figure 35.1 ENCA Block Diagram

35.3 Registers

35.3.1 List of Registers

ENCA registers are listed in the following table.

<ENCAn_base> is defined in **Section 35.1.2, Register Base Address**.

Table 35.11 List of Registers

Module Name	Register Name	Symbol	Address
ENCAn	ENCAn capture/compare register 0	ENCAnCCR0	<ENCAn_base>
	ENCAn capture/compare register 1	ENCAnCCR1	<ENCAn_base> + 04_H
	ENCAn counter register	ENCAnCNT	<ENCAn_base> + 08_H
	ENCAn status flag register	ENCAnFLG	<ENCAn_base> + $0C_H$
	ENCAn status flag clear register	ENCAnFGC	<ENCAn_base> + 10_H
	ENCAn timer enable status register	ENCAnTE	<ENCAn_base> + 14_H
	ENCAn timer start trigger register	ENCAnTS	<ENCAn_base> + 18_H
	ENCAn timer stop trigger register	ENCAnTT	<ENCAn_base> + $1C_H$
	ENCAn I/O control register 0	ENCAnIOC0	<ENCAn_base> + 20_H
	ENCAn control register	ENCAnCTL	<ENCAn_base> + 40_H
	ENCAn I/O control register 1	ENCAnIOC1	<ENCAn_base> + 44_H
	ENCAn emulation register	ENCAnEMU	<ENCAn_base> + 48_H

35.3.2 ENCAnCTL — ENCAn Control Register

This register is used to configure various operation settings of the Encoder Timer.

Access:	This register can be read or written in 16-bit units. Writing to this register during operation is prohibited.
Address:	<ENCAn_base> + 40_H
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAn CME	ENCAn MCS	—	—	—	—	ENCAn CRM1	ENCAn CRM0	ENCAn CTS	—	—	ENCAn LDE	ENCAn ECM1	ENCAn ECM0	ENCAnUDS [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 35.12 ENCAnCTL Register Contents

Bit Position	Bit Name	Function
15	ENCAnCME	Encoder Clear Mask Enable This bit is used to enable/disable masking of compare-match interrupt detection when the compare function is used. 0: Disables the compare-match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register 1: Enables the compare-match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 0. When this bit is set to “1”, setting ENCAnECM1 to “1” is prohibited.
14	ENCAnMCS	Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare-match interrupt detection ENCATINT1 when the compare function is used. This bit is valid only when ENCAnCRM1 = 0. 0: Masking of compare-match interrupt detection is canceled when the ENCAnCCR1 register is written. 1: Masking of compare match interrupt detection is canceled when one of the following three operations is performed. – Timer counter clear operation accompanying encoder clear input – Timer counter clear operation upon compare-match between ENCAnCNT and ENCAnCCR0 when ENCAnECM0 = 1 – Loading from ENCAnCCR0 to the timer counter upon underflow detection when ENCAnLDE = 1
13 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	ENCAnCRM1	ENCAnCCR1 Register Mode 0: ENCAnCCR1 used as compare register. 1: ENCAnCCR1 used as capture register.
8	ENCAnCRM0	ENCAnCCR0 Register Mode 0: ENCAnCCR0 used as compare register. 1: ENCAnCCR0 used as capture register.
7	ENCAnCTS	ENCAnCCR1 Capture Trigger Select This is a trigger selection bit for the capture operation to the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 1. 0: Uses ENCATTIN1 of capture trigger 1 signal as the capture trigger for the ENCAnCCR1 register. 1: The counter clear signal selected with ENCAnSCE is used as the capture trigger for the ENCAnCCR1 register.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 35.12 ENCAnCTL Register Contents

Bit Position	Bit Name	Function
4	ENCAnLDE	ENCAn Counter Load Enable This bit is used to enable/disable setting value loading to the counter upon underflow occurrence. This bit is valid only when ENCAnCRM0 = 0. When ENCAnCRM0 = 1, loading of the ENCAnCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit. 0: Disables loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow. 1: Enables loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow.
3	ENCAnECM1	Encoder Clear Mode 1 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR1 setting value. This bit is valid only when ENCAnCRM1 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value if the next count is a down-count.
2	ENCAnECM0	Encoder Clear Mode 0 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR0 setting value. This bit is valid only when ENCAnCRM0 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value if the next count is an up-count.
1, 0	ENCAnUDS[1:0]	Up/down Count Selection 1 and 0 These are the counter up/down control bits using ENCAnE0 and ENCAnE1. 00: Upon detection of valid edge of ENCAnE0, - down-count when ENCAnE1 = H, - up-count when ENCAnE1 = L 01: Upon detection of valid edge of ENCAnE0, up-count, Upon detection of valid edge of ENCAnE1, down-count 10: At rising edge of ENCAnE0, down-count At falling edge of ENCAnE0, up-count However, count operation is performed only when ENCAnE1 = L. 11: Detection of both edges of ENCAnE0, ENCAnE1. The count operation is determined based on the combination of the detected edge and level.

35.3.3 ENCAnIOC0 — ENCAn I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCATTIN0, ENCATTIN1).

Access: This register can be read or written in 8-bit units.

Address: <ENCAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	ENCAnTIS[3:2]		ENCAnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 35.13 ENCAnIOC0 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	ENCAnTIS[3:2]	Input Edge Selection for Capture Trigger 1 These bits are valid only when ENCAnCTL.ENCAnCRM1 = 1 and ENCAnCTL.ENCAnCTS = 0. All other settings of ENCAnCRM1 and ENCAnCTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAnTIS[1:0]	Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCAnCTL.ENCAnCRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

35.3.4 ENCAnIOC1 — ENCAn I/O Control Register 1

This register is used to perform the clear condition setting and edge selection for input from the encoder.

Access: This register can be read or written in 8-bit units.
Writing to this register during operation is prohibited.

Address: <ENCAn_base> + 44_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS[1:0]		ENCAnEIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.14 ENCAnIOC1 Register Contents

Bit Position	Bit Name	Function
7	ENCAnSCE	Encoder Special-Clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCAnUDS1, ENCAnUDS0 to 10_B or 11_B. The operation is not guaranteed if this bit is set to 1 with ENCAnUDS1, ENCAnUDS0 set to 00_B or 01_B. 0: Clears the counter upon detection of ENCAnEC valid edge (set with ENCAnECS1 and ENCAnECS0). 1: Clears the counter upon detection of input level condition of ENCAnE0, ENCAnE1 and ENCAnEC (set with ENCAnZCL bit, ENCAnBCL bit, and ENCAnACL bit).
6	ENCAnZCL	Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCAnEC) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCAnBCL	Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCAnE1) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
4	ENCAnACL	Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 (ENCAnE0) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCAnECS[1:0]	Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCAnSCE = 0; they are invalid when ENCAnSCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

Table 35.14 ENCAnIOC1 Register Contents

Bit Position	Bit Name	Function
1, 0	ENCAnEIS[1:0]	Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCAnUDS1, ENCAnUDS0 = 00_B or 01_B, and are invalid when ENCAnUDS1, ENCAnUDS0 = 10_B or 11_B. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

35.3.5 ENCAnFLG — ENCAn Status Flag Register

This register holds the status flags of the timer counter of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENCAnCSF	ENCAnUDF	ENCAnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.15 ENCAnFLG Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	ENCAnCSF	Counter Status Flag This bit reflects the current timer counter operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCAnUDF	Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared to 0 at the start of count operation. 0: This flag is cleared upon any of the following events: – “1” is written to ENCAnFGC.ENCAnCLUD – The flag is cleared to 0 by setting ENCAnTS bit to “1” when ENCAnTE = 0 or by setting the simultaneous start trigger input (ENCATSST signal) to “High”. 1: This flag is set to “1” upon occurrence of an underflow during the encoder timer count operation.
0	ENCAnOVF	Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared to 0 at the start of count operation. 0: This flag is cleared upon any of the following events: – “1” is written to ENCAnFGC.ENCAnCLOV – The flag is cleared to 0 by setting ENCAnTS bit to “1” when ENCAnTE = 0 or by setting the simultaneous start trigger input (ENCATSST signal) to “High”. 1: This flag is set to “1” upon occurrence of an overflow during the encoder timer count operation.

35.3.6 ENCAnFGC — ENCAn Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAnFLG.

Access: This register is a write-only register that can be written in 8-bit units.

This register always returns 0 when read.

Address: <ENCAn_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ENCAnCLUD	ENCAnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 35.16 ENCAnFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	ENCAnCLUD	Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCAnUDF of the ENCAnFLG register (clears underflow detection).
0	ENCAnCLOV	Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCAnOVF of the ENCAnFLG register (clears overflow detection).

35.3.7 ENCAnCCR0 — ENCAn Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

Access: This register can be read or written in 16-bit units.

When this register functions as a capture register, only reading is possible. Write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

Address: <ENCAn_base> + 00_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.17 ENCAnCCR0 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR0[15:0]	Capture/Compare Register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAnCTL.ENCAnLDE setting. See the description of the ENCAnLDE bit in ENCA control register ENCAnCTL for details. ● If ENCAnCTL.ENCAnCRM0 = 0: ENCAnCCR0 is a compare register. Set the value to be compared with the timer counter value. ● If ENCAnCTL.ENCAnCRM0 = 1: ENCAnCCR0 is a capture register. The captured timer counter value is stored.

35.3.8 ENCAnCCR1 — ENCAn Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

Access: This register can be read or written in 16-bit units.

When this register functions as a capture register, only reading is possible. Write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

Address: <ENCAn_base> + 04_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.18 ENCAnCCR1 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR1[15:0]	Capture/Compare Register 1 During capture operation, the capture trigger to this register differs according to the ENCAnCTL.ENCAnCTS setting. See the description of the ENCAnCTS bit in ENCA control register ENCAnCTL for details. ● If ENCAnCTL.ENCAnCRM1 = 0: ENCAnCCR1 is a compare register. Set the value to be compared with the timer counter value. ● If ENCAnCTL.ENCAnCRM1 = 1: ENCAnCCR1 is a capture register. The captured timer counter value is stored.

35.3.9 ENCAnCNT — ENCAn Counter Register

This register is the 16-bit timer counter register.

Access: This register can be read or written in 16-bit units.
This register can be written only when the operation is stopped.

Address: <ENCAn_base> + 08_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.19 ENCAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCNT[15:0]	Counter Register ● ENCAnTE.ENCAnTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. ● ENCAnTE.ENCAnTE status: 0 → 1 (operation start): Count operation start Up/down count operation is started with the set arbitrary value. ● ENCAnTE.ENCAnTE status: 1 (operating): Counting Up/down count operation is performed. ● ENCAnTE.ENCAnTE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped.

35.3.10 ENCAnTE — ENCAn Timer Enable Status Register

This register indicates the operating status of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.20 ENCAnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ENCAnTE	Timer Status Enable This is a status bit that indicates the operation enabled/stopped status of ENCAn. This bit is cleared to 0 when “1” is written to ENCAnTT.ENCAnTT. This bit is set to “1” when “1” is written to ENCAnTS.ENCAnTS, or when the input signal of ENCATSST is set to High level. 0: Operation stopped status 1: Operation enabled status

35.3.11 ENCAnTS — ENCAn Timer Start Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation enabled state.

Access: This register is a write-only register that can be written in 8-bit units.

It is always read as 00_H. This register can be written only when ENCAnTE.ENCAnTE is 0.

Address: <ENCAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.21 ENCAnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTS	Timer Start Trigger This is the trigger bit that sets the ENCAn to the operation enabled state. 0: Writing is ignored. 1: The ENCAn is set to the operation enabled state by setting ENCAnTE.ENCAnTE = 1.

35.3.12 ENCAnTT — ENCAn Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation stopped state.

Access: This register is a write-only register that can be written in 8-bit units.

It is always read as 00_H.

Address: <ENCAn_base> + $1C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.22 ENCAnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTT	Timer Stop Trigger This is the trigger bit that sets the ENCAn to the operation stopped state. 0: Writing is ignored. 1: Clears ENCAnTE.ENCAnTE to “0”, to set the ENCAn to the count operation stopped state.

35.3.13 ENCAnEMU — ENCAn Emulation Register

This register controls operations by SVSTOP.

Access: This register can be read or written in 8-bit units.

Writing to this register should be performed in the counter operation stopped status (ENCAnTE.ENCAnTE = 0 and EPC.SVSTOP = 0).

Address: <ENCAn_base> + 48_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 35.23 ENCAnEMU Register Contents

Bit Position	Bit Name	Function
7	ENCAnSVSDIS	● When EPC.SVSTOP bit = 0: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.), regardless of the value of this bit (1 or 0). ● When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger assumes control of the microcontroller (at a break point, etc.). 1: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

35.4 Operation

The ENCAn operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCAnCCR0 and ENCAnCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

35.4.1 Timer Counter Operation

The timer counter operations of the ENCAn are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

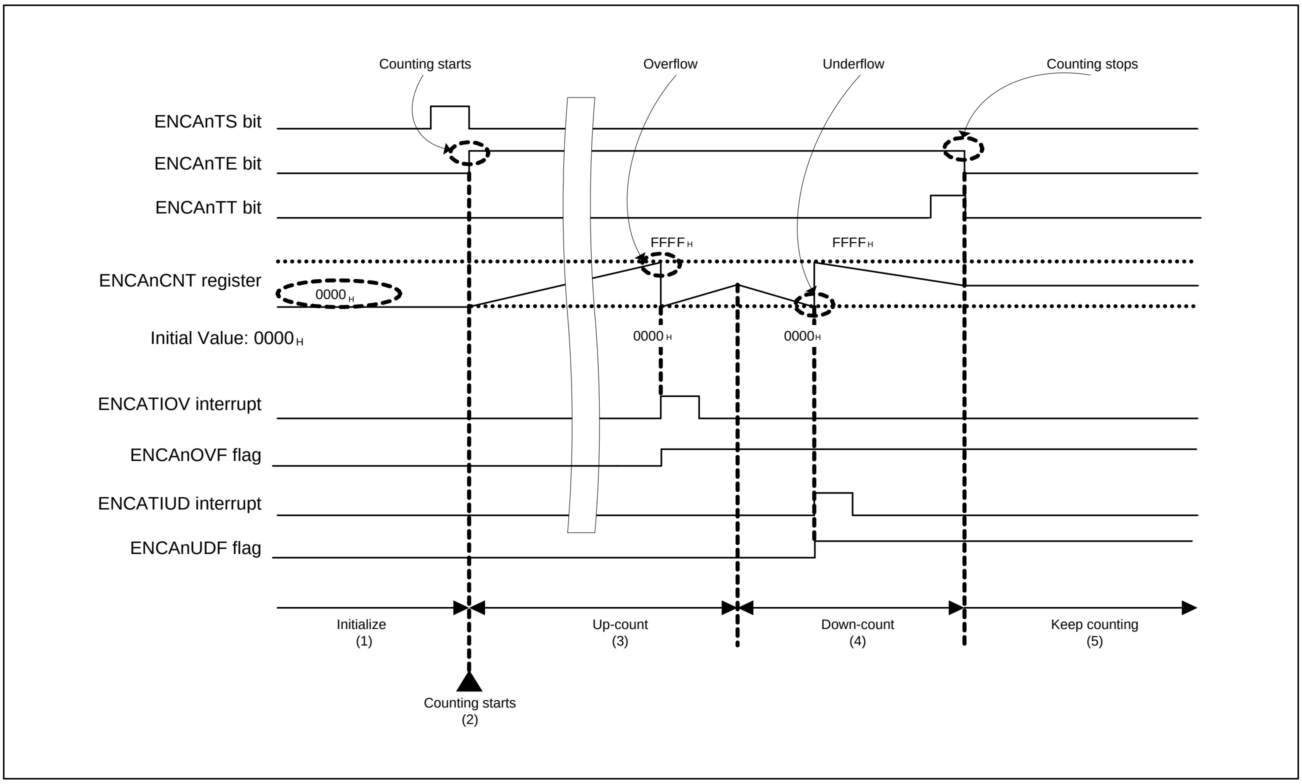


Figure 35.2 Timer Counter Initial Value Setting/Start/Stop

(1) Timer Counter Initial Value Setting

The initial value of the ENCAn counter register (ENCAnCNT) can be set in the counter operation stopped status (ENCAnTE = 0).

(2) Timer Counter Startup

By writing “1” to the timer start trigger bit (ENCAnTS), the timer status enable bit (ENCAnTE) is set to “1”, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

(3) Overflow Operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. If the counter value changes from $FFFF_H$ to 0000_H, an overflow interrupt (ENCATIOV) is generated, and the overflow flag (ENCAnOVF) is set to “1”. The overflow flag (ENCAnOVF) is cleared to “0” when “1” is set to the overflow flag clear bit (ENCAnCLOV). For details about the operation, see **Section 35.6.1, Overflow Occurrence and Overflow Flag Clear Operation**.

(4) Underflow Operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. If the counter value changes from 0000_H to $FFFF_H$, an underflow interrupt (ENCATIUD) is generated, and the underflow flag (ENCAnUDF) is set to “1”. The underflow flag (ENCAnUDF) is cleared to “0” when “1” is set to the underflow flag clear bit (ENCAnCLUD). For details about the operation, see **Section 35.6.2, Underflow Occurrence and Underflow Flag Clear Operation**.

(5) Timer Counter Stop

By writing “1” to the timer stop trigger bit (ENCAnTT), the timer status enable bit (ENCAnTE) is cleared to “0”, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stop.

35.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCAnE0, ENCAnE1) according to the settings of ENCAnUDS1, ENCAnUDS0.

35.4.2.1 When the ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

Table 35.24 When ENCAnUDS1, ENCAnUDS0 Bits = 00_B

		Operation Description		
ENCAnUDS1	ENCAnUDS0	ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
0	0	Rising edge	High level	Down
		Falling edge		
		Rising and falling edges		
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		

The valid edge for ENCAnE0 is specified by setting ENCAnEIS1, ENCAnEIS0.

Up/down count operation is performed when the valid edges and levels of ENCAnE0 and ENCAnE1 match.

The following timing chart shows the count operation when ENCAnUDS1, ENCAnUDS0 bits = 00_B.

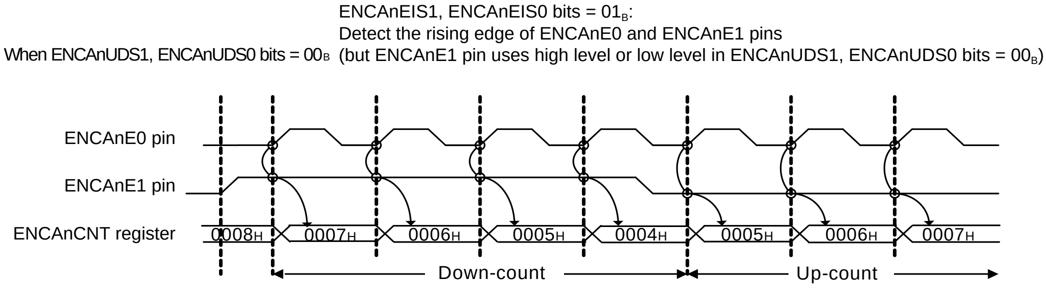


Figure 35.3 Count Operation when the ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

35.4.2.2 When the ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

Table 35.25 When the ENCAnUDS1, ENCAnUDS0 Bits = 01_B

		Operation Description		
ENCAnUDS1	ENCAnUDS0	ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
0	1	Low level	Rising edge	Down
			Falling edge	
			Rising and falling edges	
		High level	Rising edge	
			Falling edge	
			Rising and falling edges	
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		
		Rising edge	High level	
		Falling edge		
		Rising and falling edges		
		Simultaneous input		Hold

The valid edges for ENCAnE0 and ENCAnE1 are specified by setting ENCAnEIS1, ENCAnEIS0.

Up/down count operation is performed when the valid edges and levels of the ENCAnE0 and ENCAnE1 pins match, and the count is held when the valid edges overlap.

The following timing chart shows the count operation when ENCAnUDS1, ENCAnUDS0 bits = 01_B.

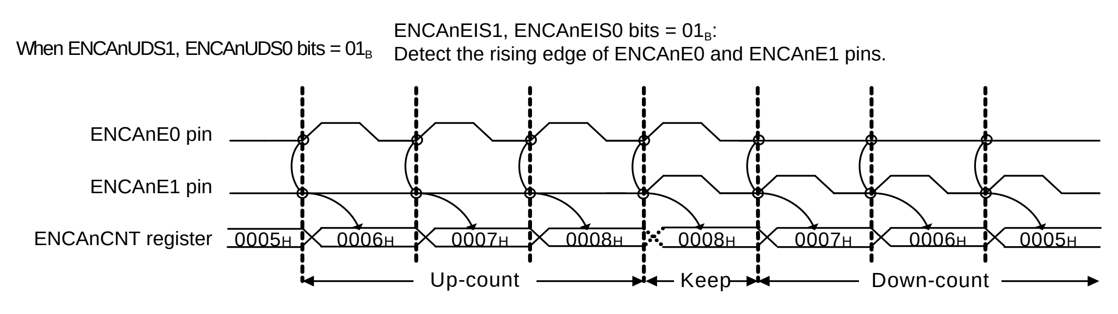


Figure 35.4 Count Operation when the ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

35.4.2.3 When the ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 10_B

Table 35.26 When the ENCAnUDS1, ENCAnUDS0 Bits = 10_B

		Operation Description		
ENCAnUDS1	ENCAnUDS0	ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

The valid edge specification for ENCAnE0 and ENCAnE1 (settings of ENCAnEIS1, ENCAnEIS0) is invalid.

The following timing chart shows the count operation when the ENCAnUDS1, ENCAnUDS0 bits = 10_B.

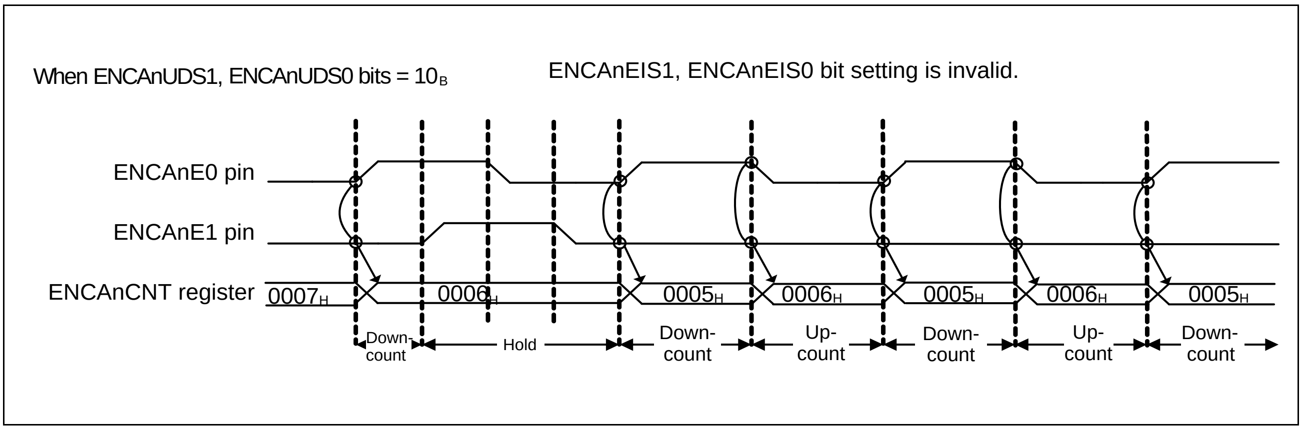


Figure 35.5 Count Operation when ENCAnUDS1, ENCAnUDS0 Bits in ENCAnCTL Register = 10_B

35.4.2.4 When ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

Table 35.27 When ENCAnUDS1, ENCAnUDS0 Bits = 11_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Hold

Valid edge specification for ENCAnE0 and ENCAnE1 (settings of ENCAnEIS1, ENCAnEIS0) is invalid.

The counter value is held when the valid edges of ENCAnE0 and ENCAnE1 overlap.

The following timing chart shows the count operation when ENCAnUDS1, ENCAnUDS0 bits = 11_B.

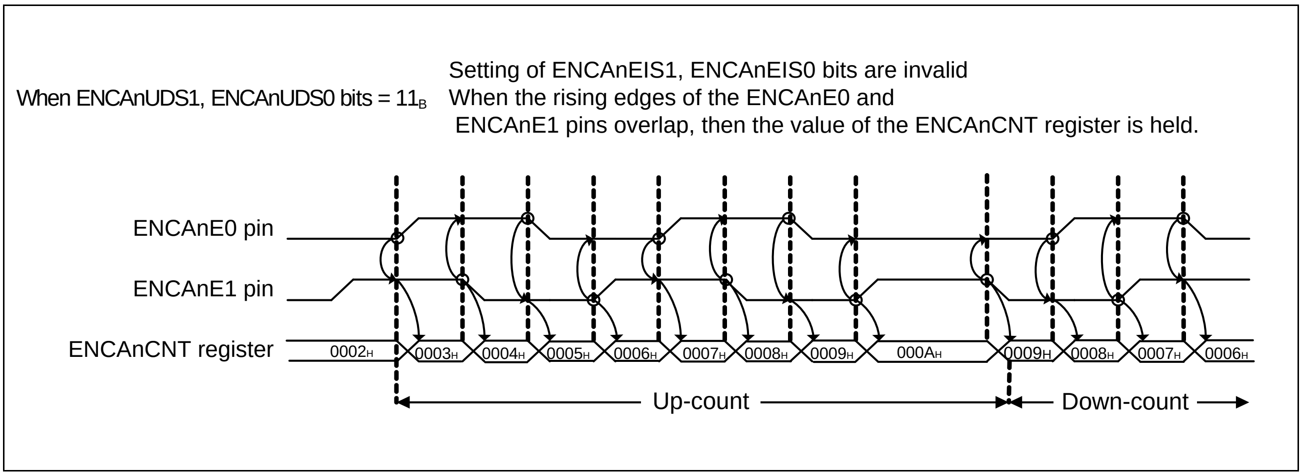


Figure 35.6 Count Operation when ENCAnUDS1, ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

35.4.3 Timer Counter Clear Control by Encoder Input

The timer counter is cleared to 0000_H by encoder clear input (ENCAnEC).

Two types of clearing methods can be selected by controlling the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS1, and ENCAnECS0 bits of the ENCAnIOC1 register.

Table 35.28 Timer Counter Clear Control by Encoder Input

Clearing method	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS1, ENCAnECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

35.4.3.1 Clearing Method when ENCAnSCE = 0

- Upon detection of the valid edge of ENCAnEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCAnEC is specified by the setting of the ENCAnECS1 and ENCAnECS0 bits.
- The settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCAnSCE = 0, see the timing chart in **Section 35.6.19, Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0**.

35.4.3.2 Clearing Method when ENCAnSCE = 1

- When the clear levels of the ENCAnEC, ENCAnE1, ENCAnE0 inputs are detected, the timer counter is cleared to 0000_H in synchronization with the operating clock.
- Specify the clear levels of the ENCAnEC, ENCAnE1, ENCAnE0 inputs by the settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits.
- The settings of the ENCAnECS1 and ENCAnECS0 bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCAnZCL, ENCAnBCL, and ENCAnACL settings are listed in the table below.

Table 35.29 Clearing Conditions of the Timer Counter

Counter Clear Condition Setting			Encoder Pin Input Level		
ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnEC	ENCAnE1	ENCAnE0
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

35.4.4 Functions of ENCAnCCR0

35.4.4.1 Compare Function

- When ENCAnCRM0 = 0, the ENCAnCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR0 setting value, a compare 0 match interrupt (ENCATINT0) is output.
- When ENCAnECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 35.30 Compare Function of ENCAnCCR0

ENCAnCCR0 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAnCCR0
ENCAnCRM0	ENCAnECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000_H.
		Down-count	Does not clear (continues count operation).

When ENCAnLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (ENCATIUD) is output.

NOTE

For the timing chart when ENCAnLDE = 1, see **Section 35.6.8, Using the ENCAnLDE Function Immediately after Startup** to **Section 35.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

35.4.4.2 Capture Function

- When ENCAnCRM0 = 1, the ENCAnCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCATTIN0), the value of the timer counter is stored into ENCAnCCR0.
- A capture 0 interrupt (ENCATINT0) is output during capture operation.

NOTE

For details about capture operation for ENCAnCCR0, see the timing charts in **Section 35.6.14, Capture Operation between Count Clocks (ENCAnCCR0)** and **Section 35.6.17, Encoder Operation when Compare Match Clear Control is Disabled**.

35.4.5 Functions of ENCAnCCR1

35.4.5.1 Compare Function

- When ENCAnCRM1 = 0, the ENCAnCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR1 setting value, a compare 1 match interrupt (ENCATINT1) is output.
- When ENCAnECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 35.31 Compare Function of ENCAnCCR1

ENCAnCCR1 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000_H.

Compare match interrupt mask function

- When ENCAnCME = 1, the compare 1 match interrupt mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAnCME = 1 and ENCAnMCS = 0, a compare 1 match interrupt is output once upon the first compare match by writing to the ENCAnCCR1 register (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAnCME = 1 and ENCAnMCS = 1, a compare 1 match interrupt is output once upon the first compare match by a timer counter clear operation accompanying encoder clear input or by a timer counter clear operation upon match between the ENCAnCCR0 register value and the timer counter value (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAnCME = 1, ENCAnMCS = 1 and ENCAnLDE = 1, a compare 1 match interrupt is output once upon the first compare match by a loading operation of the ENCAnCCR0 register to the timer counter upon underflow detection (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- Setting ENCAnECM1 to “1” is prohibited when enabling the compare 1 match interrupt mask function.

Table 35.32 Compare Match Interrupt Mask Function of ENCAnCCR1

ENCAnCCR1 Function	Compare 1 Match Interrupt Mask	Interrupt Mask Cancel Trigger	Compare 1 Match Interrupt Output upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnCME	ENCAnMCS	
0 (Compare)	0 (Mask function disabled)	— (Setting invalid)	Outputs compare 1 match interrupt upon each compare match.
	1 (Mask function enabled)	0 (Write operation to ENCAnCCR1)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)
		1 (Timer counter clear operation) (Loading from ENCAnCCR0 to the timer counter upon underflow occurrence when ENCAnLDE = 1)	

35.4.5.2 Capture Function

When ENCAnCRM1 = 1, the ENCAnCCR1 register functions as a dedicated capture register.

NOTE

For details about capture operation to ENCAnCCR1, see the timing chart in **Section 35.6.13, Capture Operation between Count Clocks (ENCAnCCR1)**.

The operations for each of the ENCAnCTS settings are shown in the table below.

Table 35.33 Operations for Each of the ENCAnCTS Settings

ENCAnCCR1 Function	Capture Trigger Selection			
ENCAnCRM1	ENCAnCTS	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input (set with ENCAnSCE)	Clears timer counter.	(1) Capture 1 interrupt (ENCATINT1) (2) Encoder clear interrupt (ENCATIEC)

NOTE

For details about the timing chart when ENCAnCTS = 0 or ENCAnCTS = 1, see the following:

Section 35.6.3, Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin), **Section 35.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 35.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 35.6.11, Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)** and **Section 35.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

35.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the value of the timer counter and the ENCAnCCR0/1 setting value, according to the settings of the ENCAnECM1 and ENCAnECM0 bits in the ENCAnCTL register, is detailed in the following table.

Table 35.34 Timer Counter Clearing Operation upon Compare Register Match

ENCAnECM1 and ENCAnECM0	Next Count Operation	Timer Counter Clearing upon Compare Match with ENCAnCCR1	Timer Counter Clearing upon Compare Match with ENCAnCCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).

35.4.6 Startup/Stop of Timer Counter

35.4.6.1 Startup of Timer

The timer operation can be started by setting the ENCAnTS bit to “1”.

PIC setting enables simultaneous start with other timers. For details, see **Section 36.8, Simultaneous Start Trigger Function**.

35.4.6.2 Stop of Timer

When the ENCAnTT bit is set to “1”, the ENCAnTE bit becomes “0” and the timer stops.

35.5 ENCAn Setting Sequences

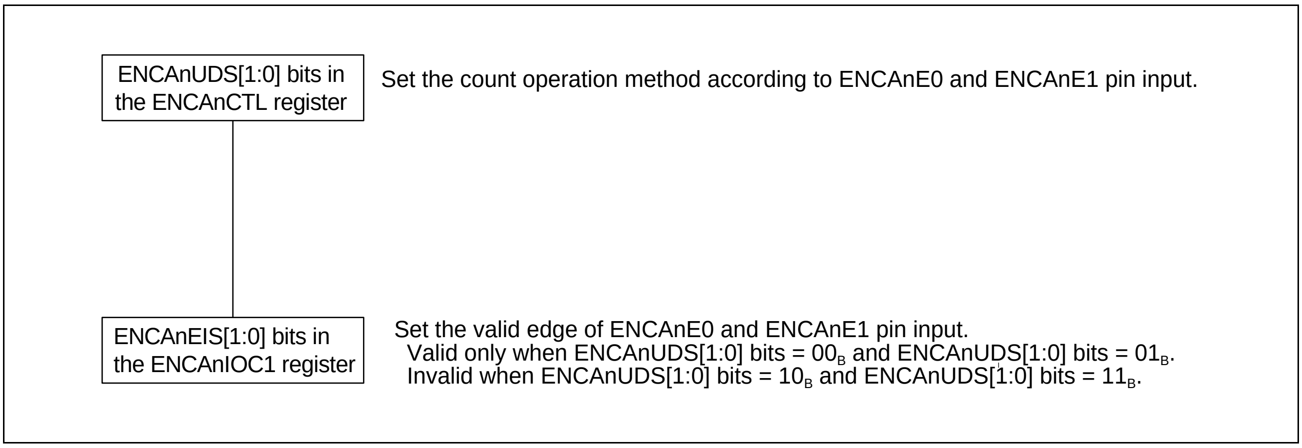
35.5.1 ENCAn Setting Procedure

The ENCAn setting procedure is described below.

Table 35.35 ENCAn Setting Procedure

Initial Setting	Action	Setting status
Initial setting	Reset deassertion	Power-on status, operation stopped status. (Writing to each register is enabled)
ENCAn initial setting	Perform the following initial settings. ● Setting for counter ● Setting for counter clear ● Setting for ENCAnCCR0 register ● Setting for ENCAnCCR1 register	This is the count operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
	Perform the counter initial value settings. ● Set any 16-bit value to ENCAnCNT register. (When, after setting this register, the ENCAnTS bit is set to “1”, the counter operation starts from the set count value.)	The set value is set as the initial value of the counter register.
Operation start	Perform the counter operation start setting. ● Set the ENCAnTS bit to “1”.	This is the counter operation starts status. The value of the ENCAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. ● ENCAnCCR0 register setting. ● ENCAnCCR1 register setting. ● ENCAnIOC0 register setting.	The count operation set with the initial setting is performed, and up/down counting is performed according to ENCAnE0 and ENCAnE1 pins.
Operation stop	Perform the counter operation stop setting during operation. ● Set the ENCAnTT bit to “1”.	This is the counter operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
ENCAn stop	Reset	The setting registers are initialized.

35.5.1.1 Initial Setting Procedure for the Counter

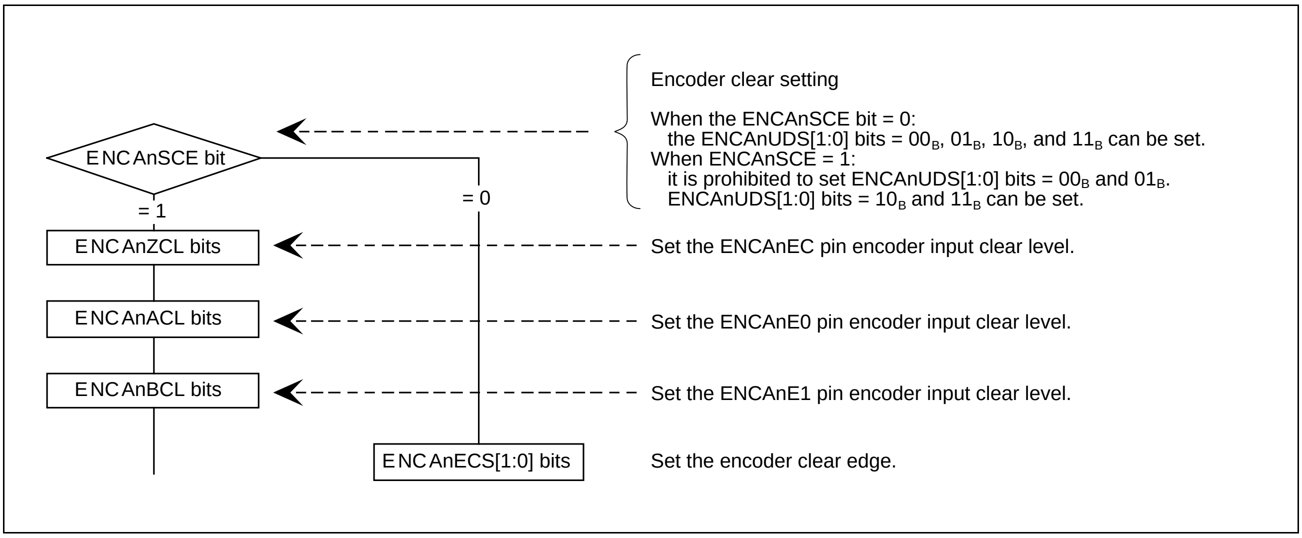


Set the count operation method according to ENCAnE0 and ENCAnE1 pin input.

Set the valid edge of ENCAnE0 and ENCAnE1 pin input.
Valid only when ENCAnUDS[1:0] bits = 00_B and ENCAnUDS[1:0] bits = 01_B.
Invalid when ENCAnUDS[1:0] bits = 10_B and ENCAnUDS[1:0] bits = 11_B.

Figure 35.7 Initial Setting Procedure for the Counter

35.5.1.2 Initial Setting Procedure for Counter Clear



Encoder clear setting

When the ENCAnSCE bit = 0:
the ENCAnUDS[1:0] bits = 00_B, 01_B, 10_B, and 11_B can be set.
When ENCAnSCE = 1:
it is prohibited to set ENCAnUDS[1:0] bits = 00_B and 01_B.
ENCAnUDS[1:0] bits = 10_B and 11_B can be set.

Set the ENCAnEC pin encoder input clear level.

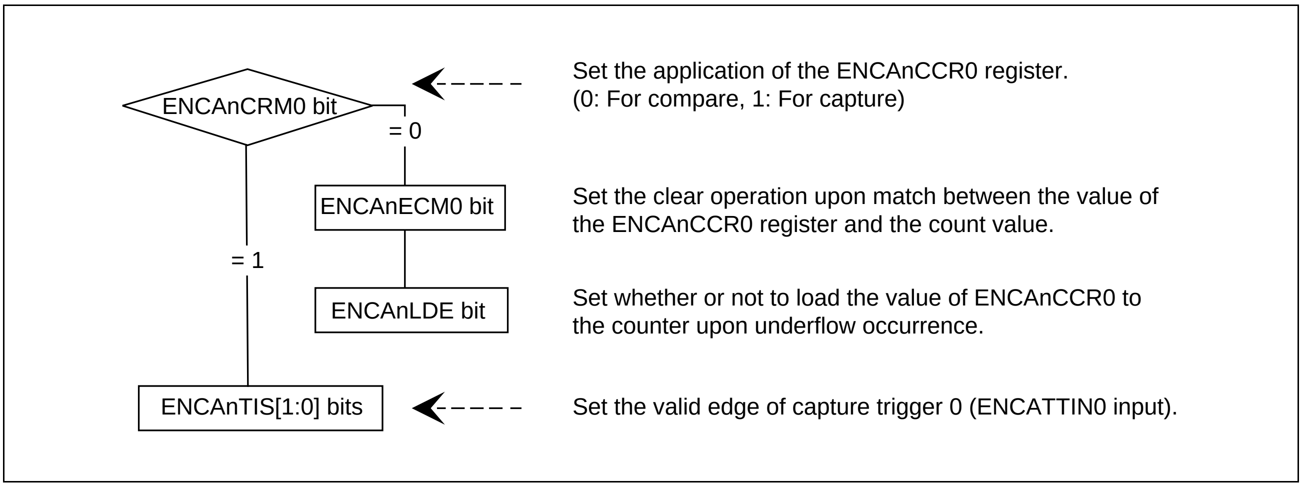
Set the ENCAnE0 pin encoder input clear level.

Set the ENCAnE1 pin encoder input clear level.

Set the encoder clear edge.

Figure 35.8 Initial Setting Procedure for Counter Clear

35.5.1.3 Setting Procedure for ENCAnCCR0 Register



Set the application of the ENCAnCCR0 register.
(0: For compare, 1: For capture)

Set the clear operation upon match between the value of the ENCAnCCR0 register and the count value.

Set whether or not to load the value of ENCAnCCR0 to the counter upon underflow occurrence.

Set the valid edge of capture trigger 0 (ENCATTIN0 input).

Figure 35.9 Setting Procedure for ENCAnCCR0 Register

35.5.1.4 Setting Procedure for ENCAnCCR1 Register

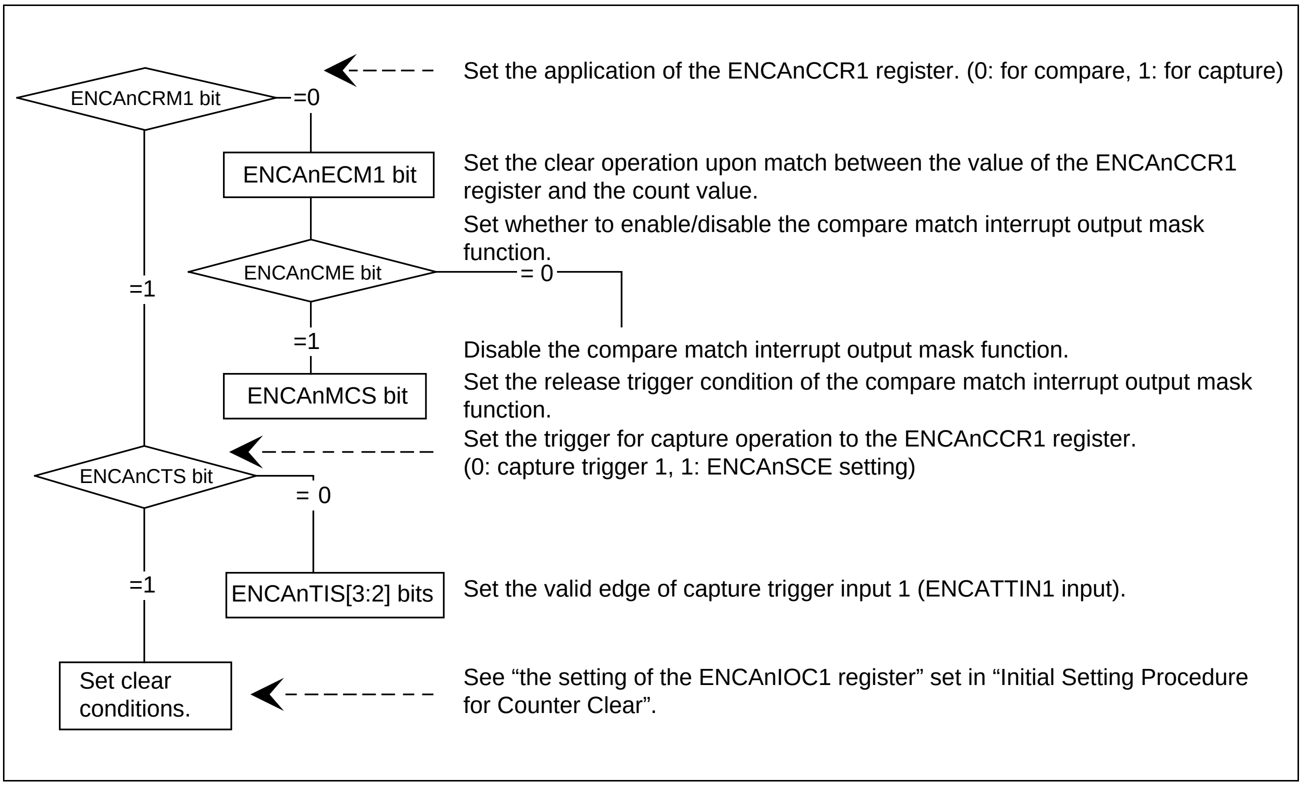


Figure 35.10 Setting Procedure for ENCAnCCR1 Register

35.6 Timing Chart

35.6.1 Overflow Occurrence and Overflow Flag Clear Operation

An overflow occurs when up-counting is performed when the counter value is FFFF_H. Once an overflow occurs, an overflow interrupt (ENCATIOV) is output and the overflow flag (ENCAnOVF) is set to 1. When the overflow clear bit (ENCAnCLOV) is set to 1, the overflow flag (ENCAnOVF) is cleared to 0.

The operations of overflow occurrence and overflow flag clearing are described below.

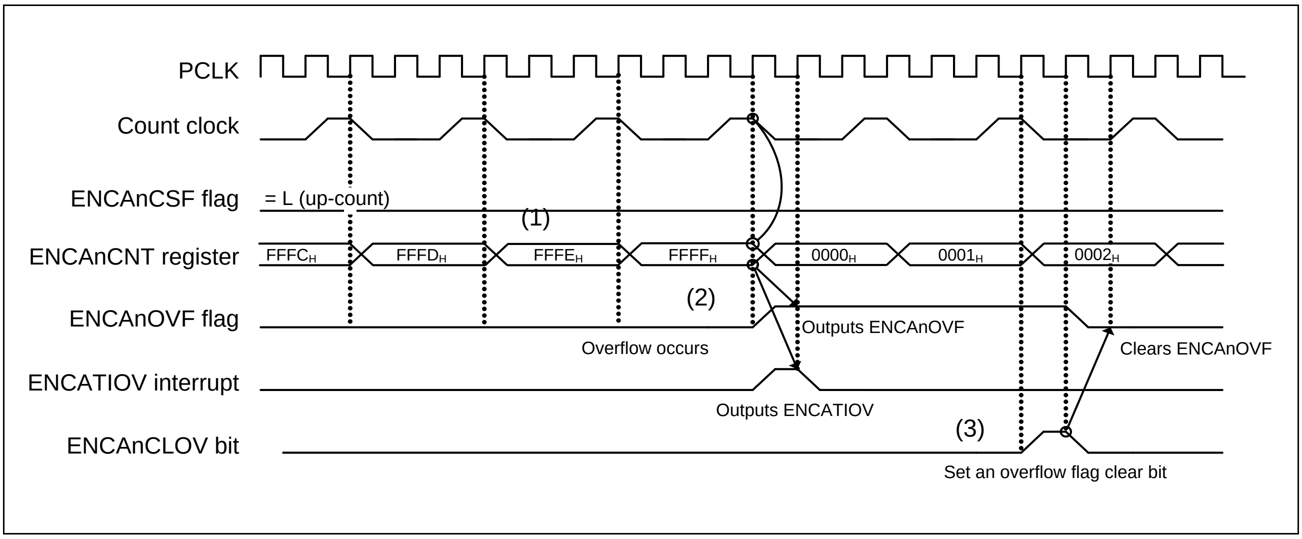


Figure 35.11 Settings of Overflow Occurrence and Overflow Flag Clear

(1) The count value is counted up from FFFE_H to FFFF_H.

(2) When the count value changes from FFFF_H to 0000_H, an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.

(3) By setting the ENCAnCLOV bit in the ENCAnFGC register to 1 by the overflow flag clearing method, the overflow flag is cleared to 0. The overflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or setting the input signal of ENCATSST (simultaneous start trigger input) to “High”.

35.6.2 Underflow Occurrence and Underflow Flag Clear Operation

An underflow occurs when down-counting is performed when the counter value is 0000_H.
Once an underflow occurs, an underflow interrupt (ENCATIUD) is output and the underflow flag (ENCAnUDF) is set to 1. When the underflow clear bit (ENCAnCLUD) is set to 1, the underflow flag (ENCAnUDF) is cleared to 0.

The operations of underflow occurrence and underflow flag clearing are described below.

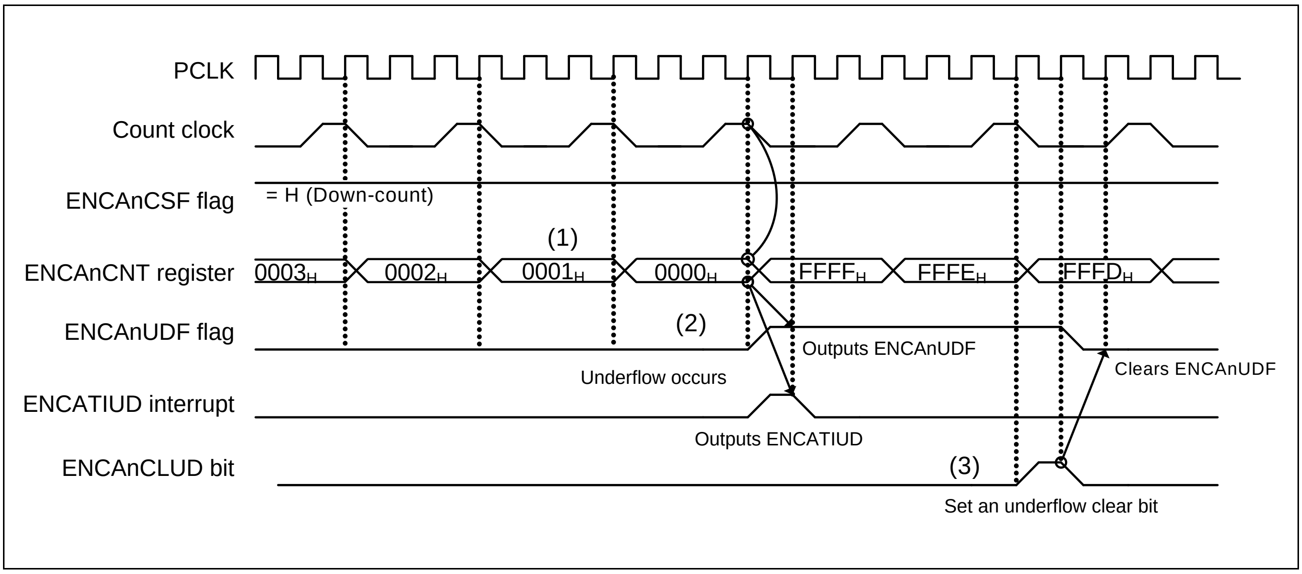


Figure 35.12 Settings of Underflow Occurrence and Underflow Flag Clear

(1) The count value is counted down from 0001_H to 0000_H.

(2) When the count value changes from 0000_H to $FFFF_H$, an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.

(3) By setting the ENCAnCLUD bit in the ENCAnFGC register to 1 by the underflow flag clearing method, the underflow flag is cleared to 0. The underflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or by setting the input signal of ENCATSST (simultaneous start trigger) to “High”.

35.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

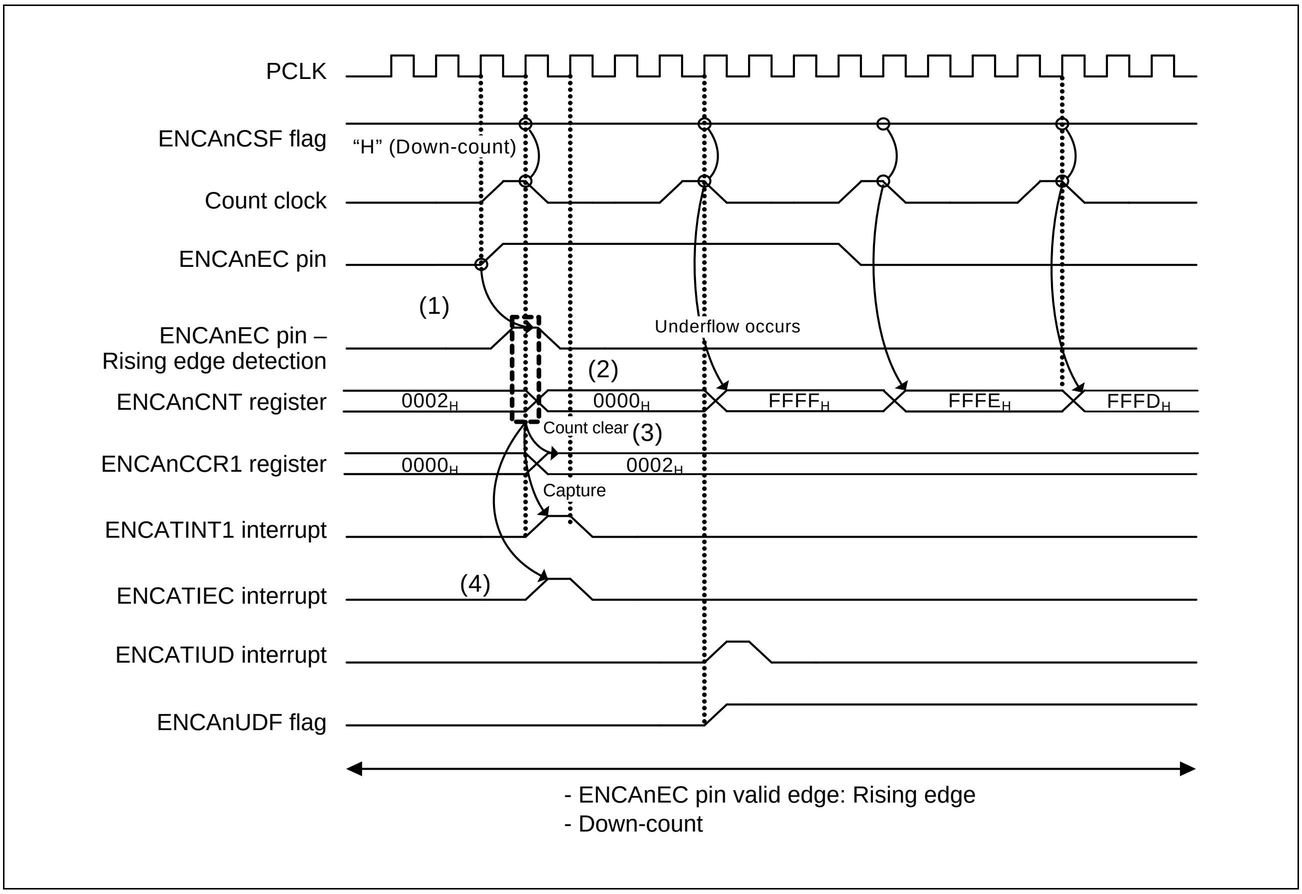


Figure 35.13 Timing Chart of Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

Setting conditions

- ENCAnCRM1 bit in the ENCAnCTL register = 1
  (Select the ENCAnCCR1 register as capture.)
- ENCAnCTS bit in the ENCAnCTL register = 1
  (Select the ENCAnEC pin input as capture trigger input.)
- ENCAnECS1 and ENCAnECS0 bits in the ENCAnIOC1 register = 01_B
  (Select the ENCAnEC pin input as rising edge detection.)

(1) Capture operation is performed by the rising edge of the ENCAnEC pin input trigger.

(2) Clearing is performed by the ENCAnEC pin input and the count value is set to 0000_H.

(3) The counter value (0002_H) is captured in the ENCAnCCR1 register by the rising edge of the ENCAnEC pin input.

(4) At the same time, a clear interrupt (ENCATIEC) and capture interrupt (ENCATINT1) due to the ENCAnEC pin input are output.

35.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

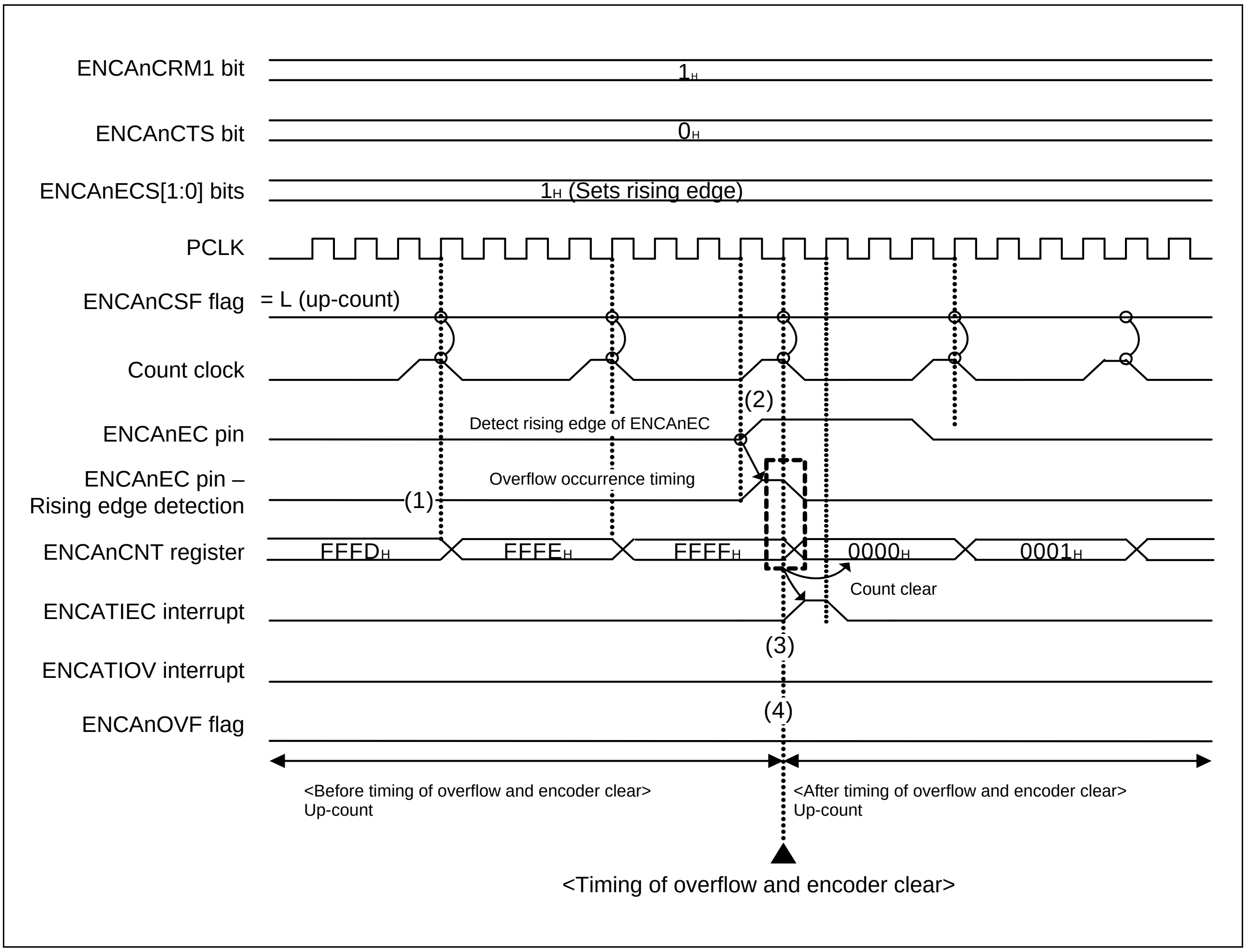


Figure 35.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

(1) An up-count from $FFFD_H$ is continuously performed.

(2) When an overflow occurs if the count value is $FFFF_H$, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The counter value is cleared to 0000_H.

(3) When the counter value is cleared by the encoder clear input, a clear interrupt (ENCATIEC) by encoder clear input is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the encoder clear input).

(4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

35.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

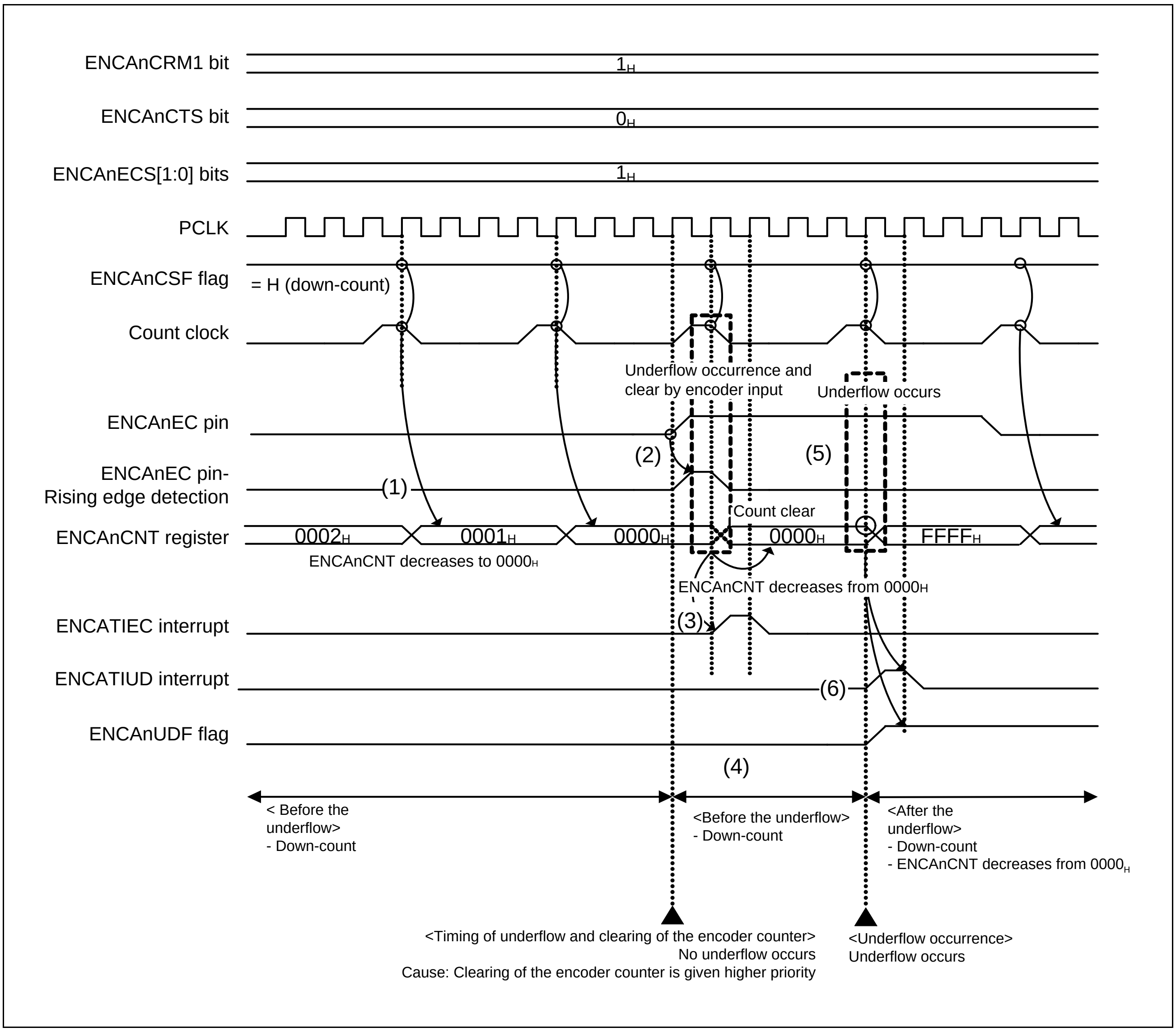


Figure 35.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

(1) A down-count from 0002_H is continuously performed.

(2) When an underflow occurs if the count value is 0000_H, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.

(3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (ENCATIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the encoder clear input).

(4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.

(5) When a further down-count is performed after the counter value changes to 0000_H by clear operation by the encoder clear input, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs.

(6) When an underflow occurs, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

35.6.6 Overflow Operation Immediately after Startup

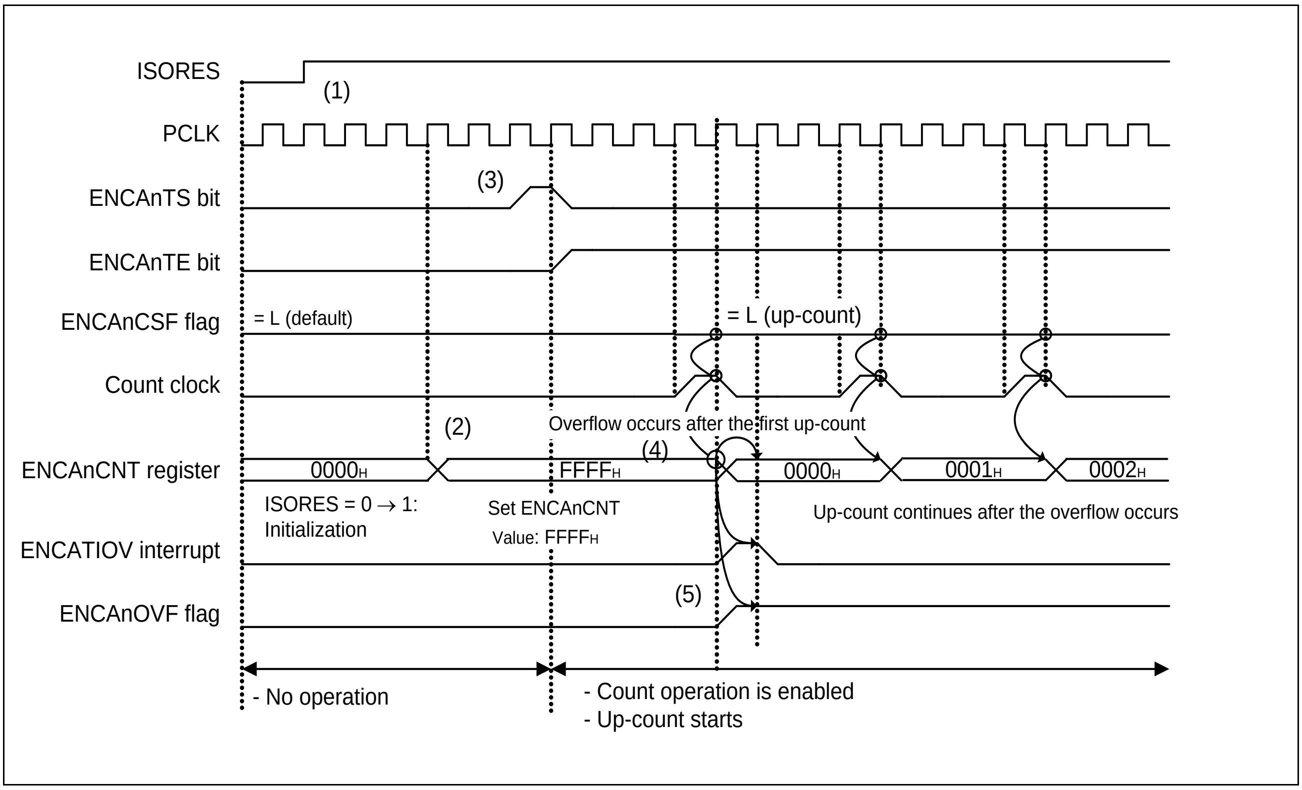


Figure 35.16 Overflow Operation Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.

(2) The timer counter is set to $FFFF_H$ as the initial value.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When an up-count is performed from $FFFF_H$ which is the initially set count value, the counter value changes from $FFFF_H$ to 0000_H, and an overflow occurs immediately after operation starts.

(5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (ENCATIOV) is output, and the overflow flag (ENCAnOVF) is set.

35.6.7 Underflow Operation Immediately after Startup

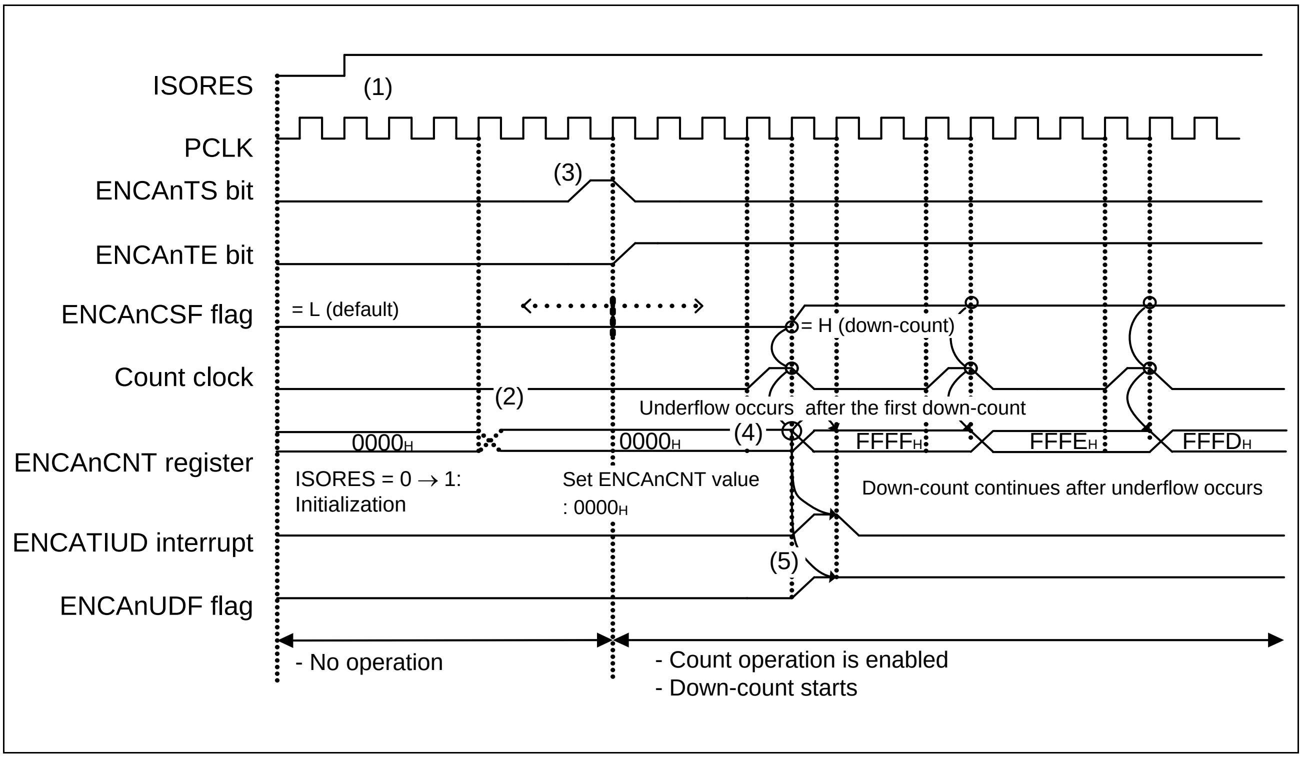


Figure 35.17 Underflow Operation Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.

(2) The timer counter is set to 0000_H as the initial value.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When a down-count is performed from 0000_H which is the initially set count value, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs immediately after operation starts.

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

35.6.8 Using the ENCAnLDE Function Immediately after Startup

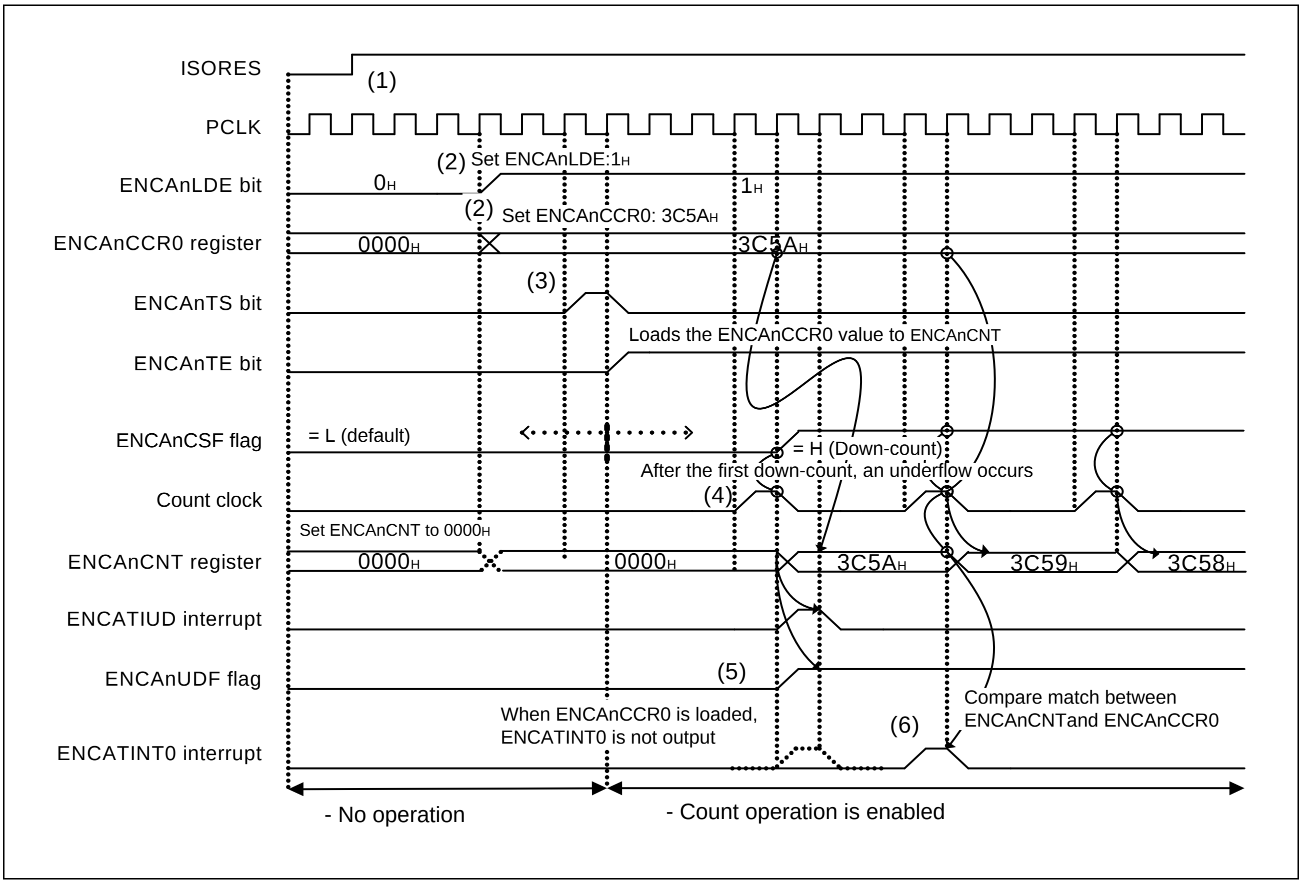


Figure 35.18 Using the ENCAnLDE Function Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.

(2) The load enable bit (ENCAnLDE) is set to “1”, capture/compare register 0 (ENCAnCCR0) is set to 3C5A$_H$, and the timer counter is set to the initial value 0000$_H$.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When a down-count is performed from 0000$_H$ which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCAnLDE is set to “1”, the ENCAnCCR0 value, 3C5A$_H$, is loaded to the timer counter (ENCATINT0 is not output during loading).

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set (after an underflow occurs, down-count operation from the loaded value (3C5A$_H$) continues).

(6) After the ENCAnCCR0 value is loaded to ENCAnCNT, a match with ENCAnCCR0 is detected, and ENCATINT0 is output.

35.6.9 ENCAnLDE Function (Loading Count Value)

(1) <When ENCAnLDE = 0>

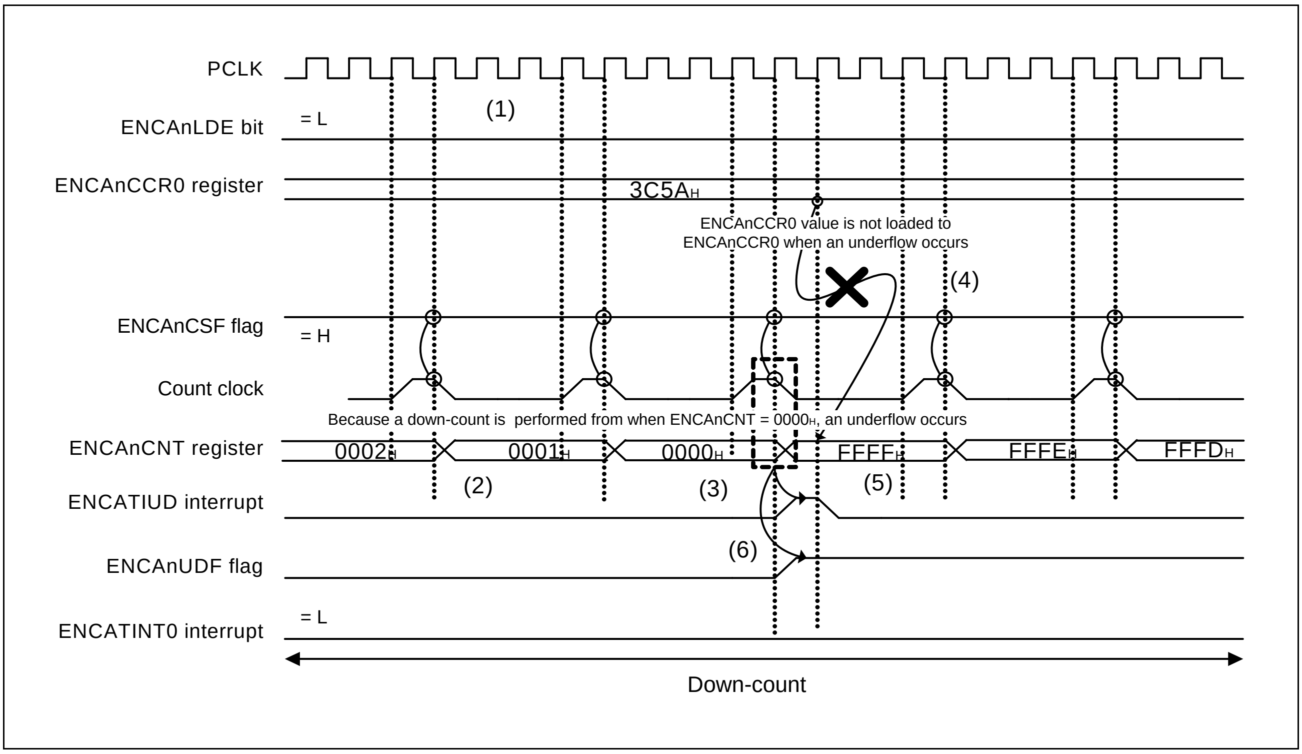


Figure 35.19 ENCAnLDE Function (when ENCAnLDE = 0)

(1) ENCAnLDE is set to “0” (even if an underflow occurs, the ENCAnCCR0 value is not loaded).

(2) A down-count is performed: 0002_H → 0001_H → 0000_H

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) Because ENCAnLDE is set to “0”, the setting value of the ENCAnCCR0 register is not loaded to the counter even if an underflow occurs.

(5) Operation changes to underflow operation (counter value: 0000_H → $FFFF_H$).

(6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(2) <When ENCAnLDE = 1>

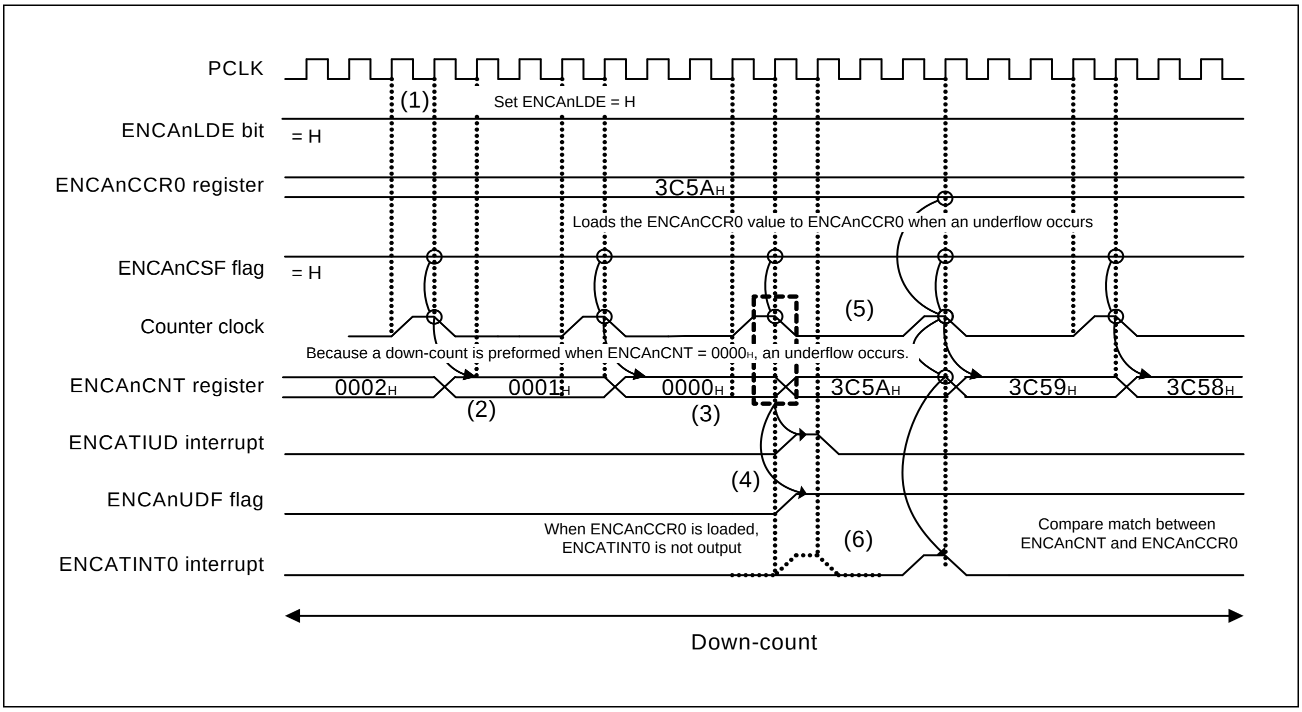


Figure 35.20 ENCAnLDE Function (when ENCAnLDE = 1)

(1) ENCAnLDE is set to “1” (if an underflow occurs, the ENCAnCCR0 value is loaded to the counter).

(2) A down-count is performed: 0002_H → 0001_H → 0000_H

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) An underflow interrupt is output, and the underflow flag is set.

(5) Because ENCAnLDE is set to “1”, the setting value of the ENCAnCCR0 register is loaded to the counter if an underflow occurs. ENCAnCNT is set to $3C5A_H$.

(6) After the ENCAnCCR0 value is set to ENCAnCNT, if the ENCAnCNT value matches the ENCAnCCR0 value on a count clock, a compare match interrupt (ENCATINT0) is output.

35.6.10 Conflict between ENCAnLDE Function (Loading Counter Value) and Rewrite of ENCAnCCR0 Register

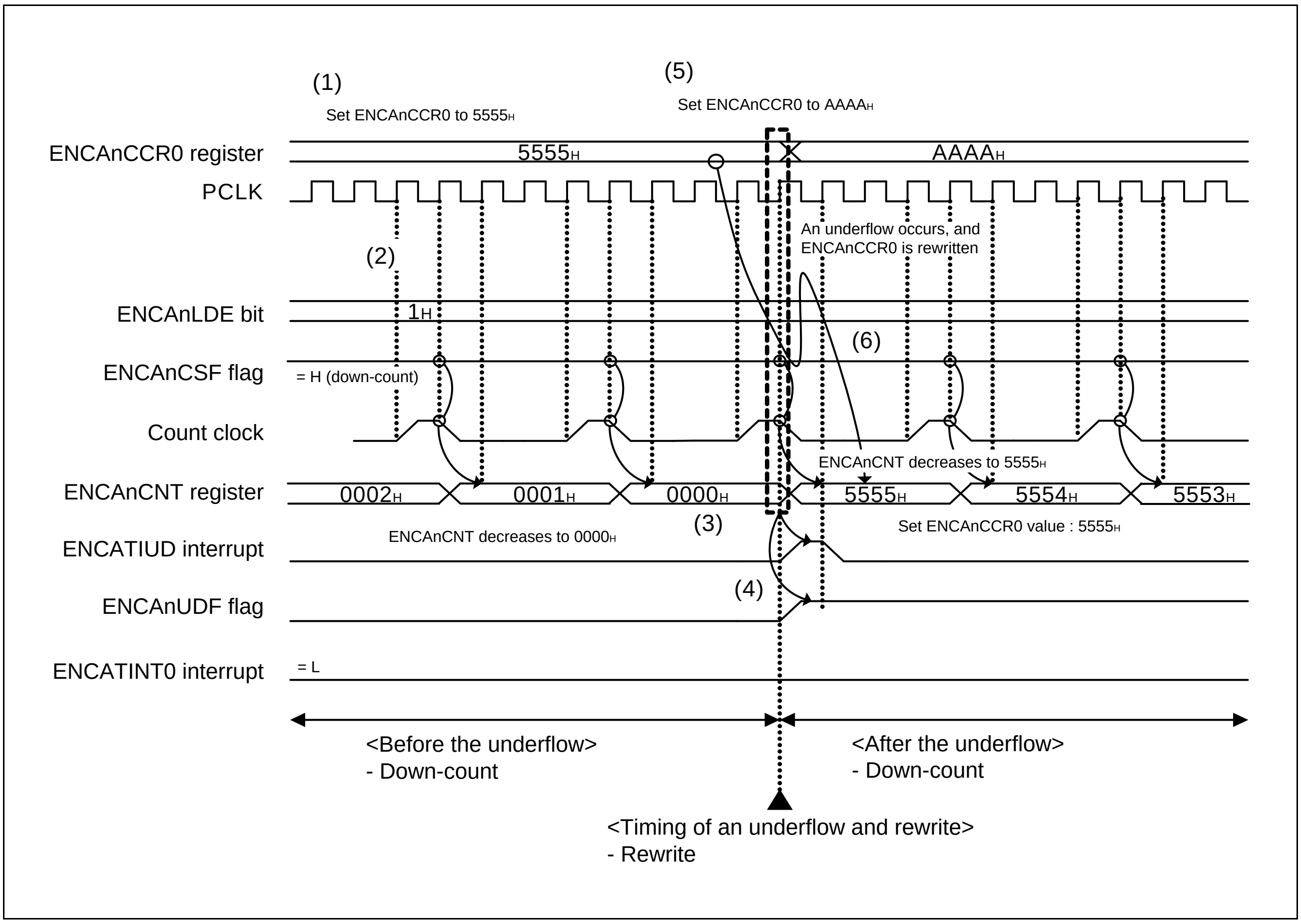


Figure 35.21 Conflict between ENCAnLDE Function and Rewrite of ENCAnCCR0 Register

(1) The ENCAnCCR0 register is currently set to 5555_H.

(2) ENCAnLDE is currently set to “1”.

(3) A down-count is performed ($0002_H \rightarrow 0001_H \rightarrow 0000_H$), and an underflow occurs.

(4) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(5) When an underflow occurs, the ENCAnCCR0 register value is changed from 5555_H to $AAAA_H$.

(6) Additionally, when an underflow occurs, the ENCAnCCR0 value before the rewrite was performed (5555_H) is set in ENCAnCNT.

35.6.11 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

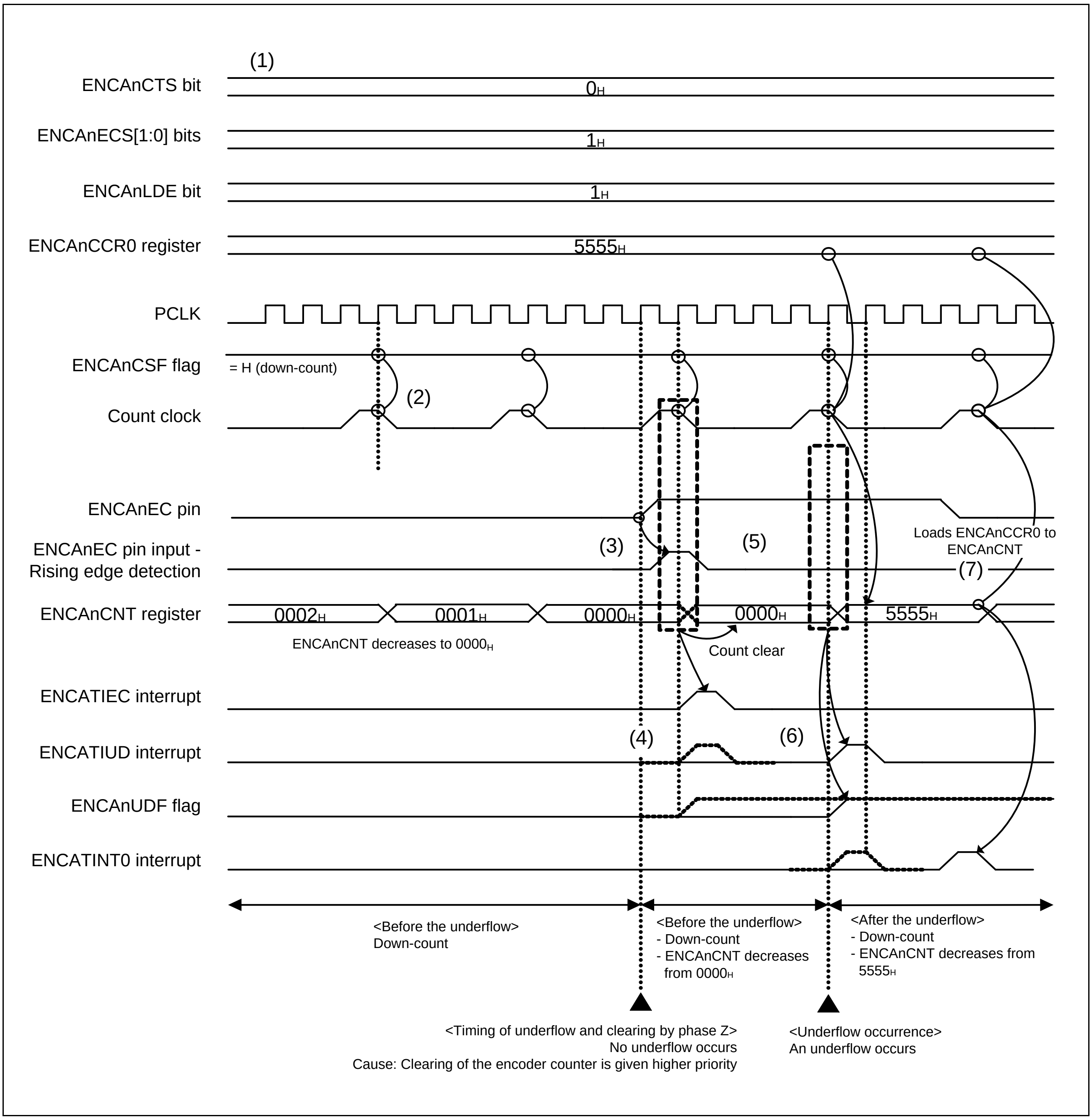


Figure 35.22 Conflict between ENCAnLDE Function and Clear Operation by Encoder Clear Input

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC pin is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when

the count value is 0000_H. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, a down-count is performed and an underflow occurs.

(6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(7) Because ENCAnLDE = “1”, if an underflow occurs, the ENCAnCCR0 value is loaded to ENCAnCNT.

(8) After the ENCAnCCR0 value is set to ENCAnCNT, a compare match is detected according to the count clock. If the ENCAnCNT value matches the ENCAnCCR0 value, a compare match interrupt (ENCATINT0) is output.

35.6.12 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

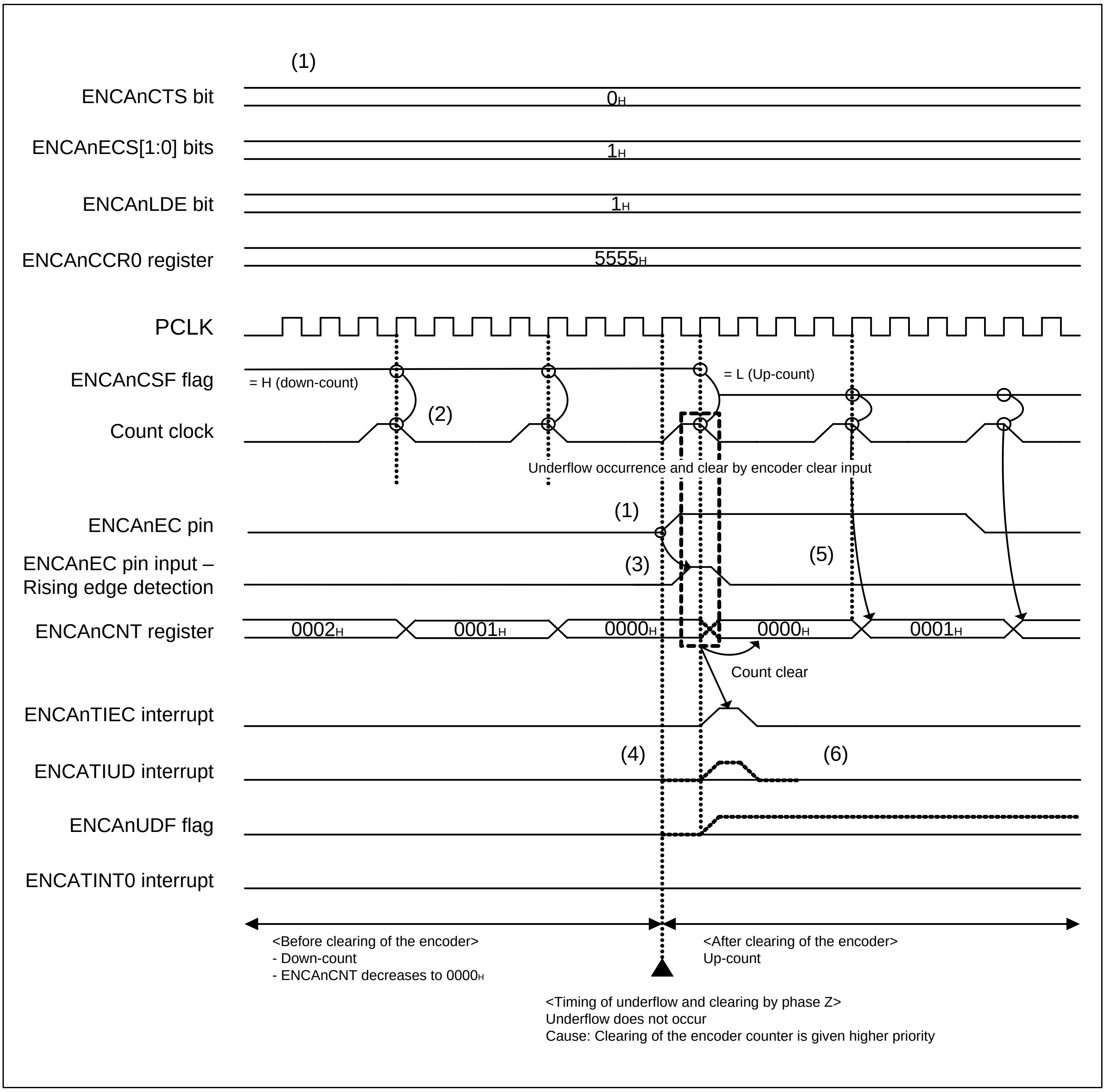


Figure 35.23 Up-count after Conflict between ENCAnLDE Function and Encoder Clear

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC pin is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when

the count value is 0000_H. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, an up-count is performed.

(6) An underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

35.6.13 Capture Operation between Count Clocks (ENCAnCCR1)

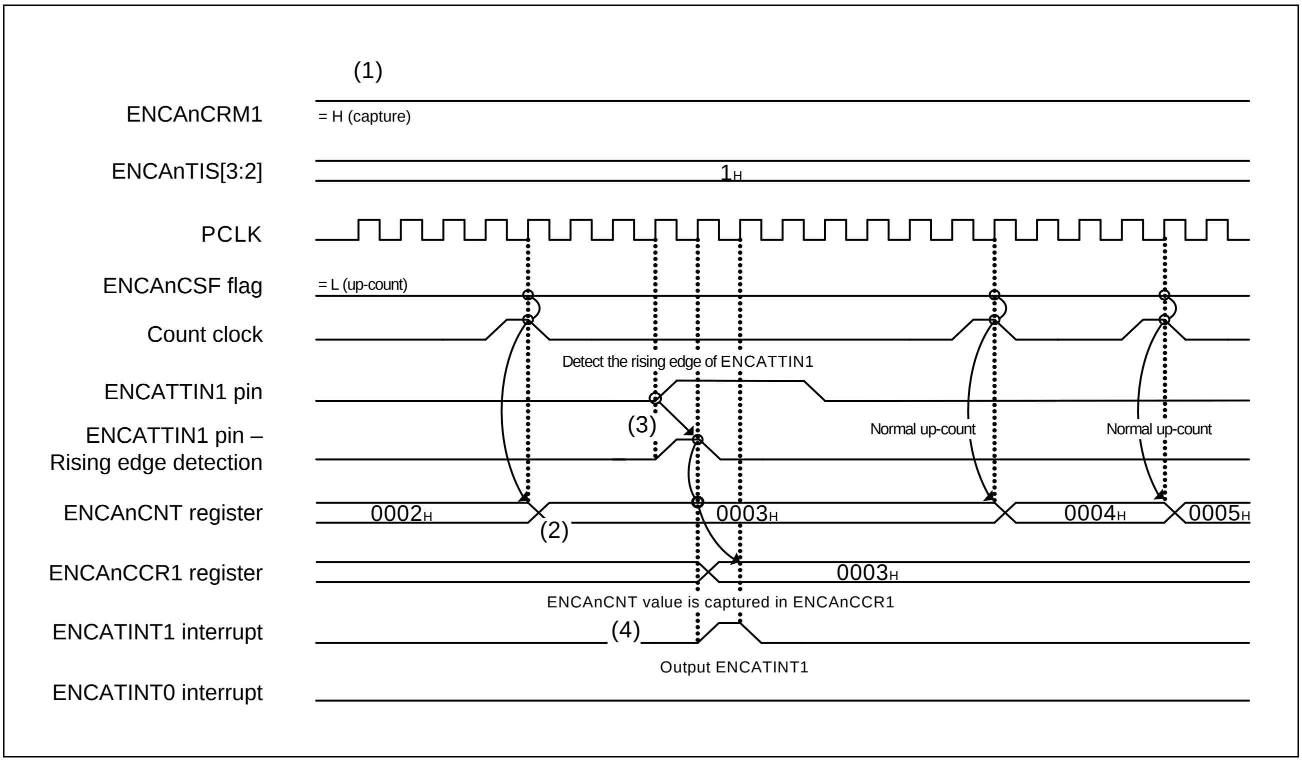


Figure 35.24 Capture Operation between Count Clocks (ENCAnCCR1)

(1) The values are set as follows: ENCAnCRM1 = 1, and ENCAnTIS[3:2] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCATTIN1 input is detected, and the count value is captured in ENCAnCCR1.

(4) An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register is output.

35.6.14 Capture Operation between Count Clocks (ENCAnCCR0)

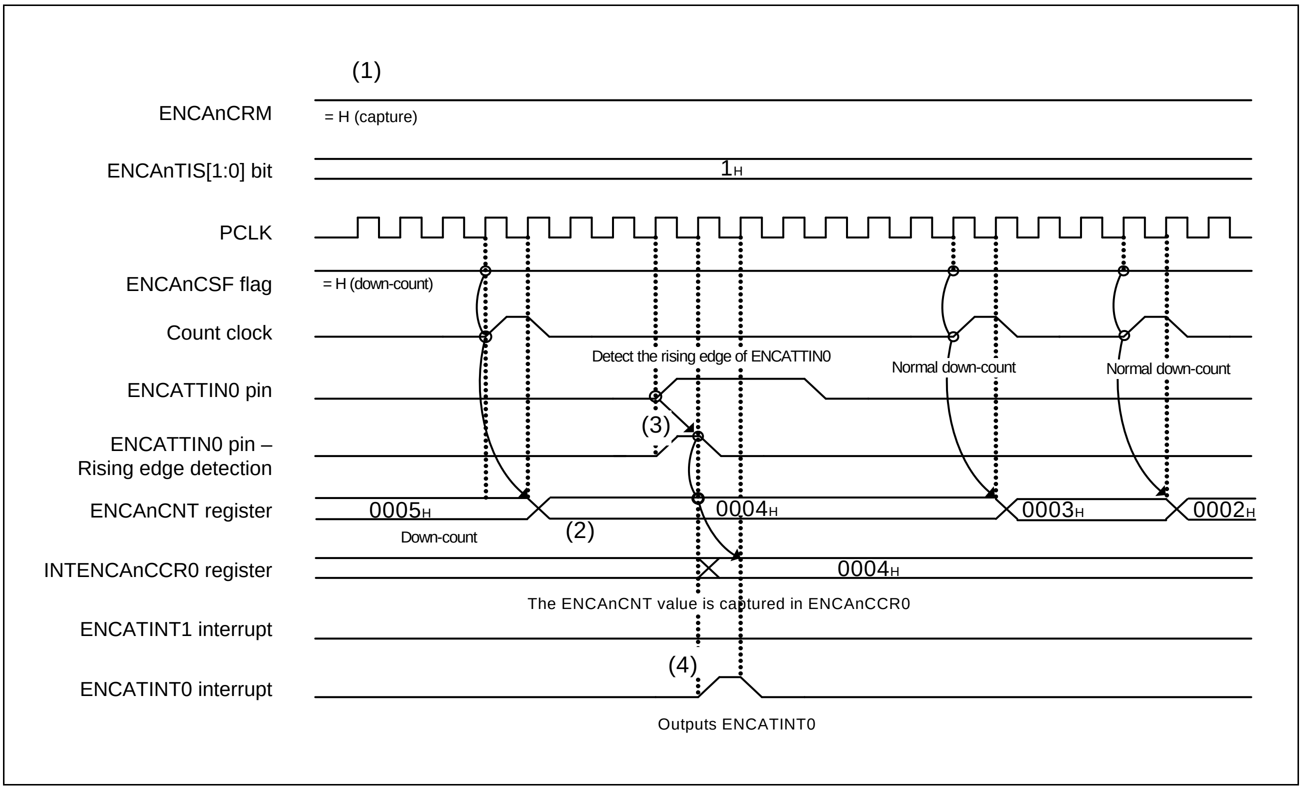


Figure 35.25 Capture Operation between Count Clocks (ENCAnCCR0)

(1) The values are set as follows: ENCAnCRM0 = 1, and ENCAnTIS[1:0] = 01_B.

(2) A down-count is performed.

(3) The rising edge of the ENCATTIN0 input is detected, and the count value is captured in ENCAnCCR0.

(4) An interrupt (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.

35.6.15 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

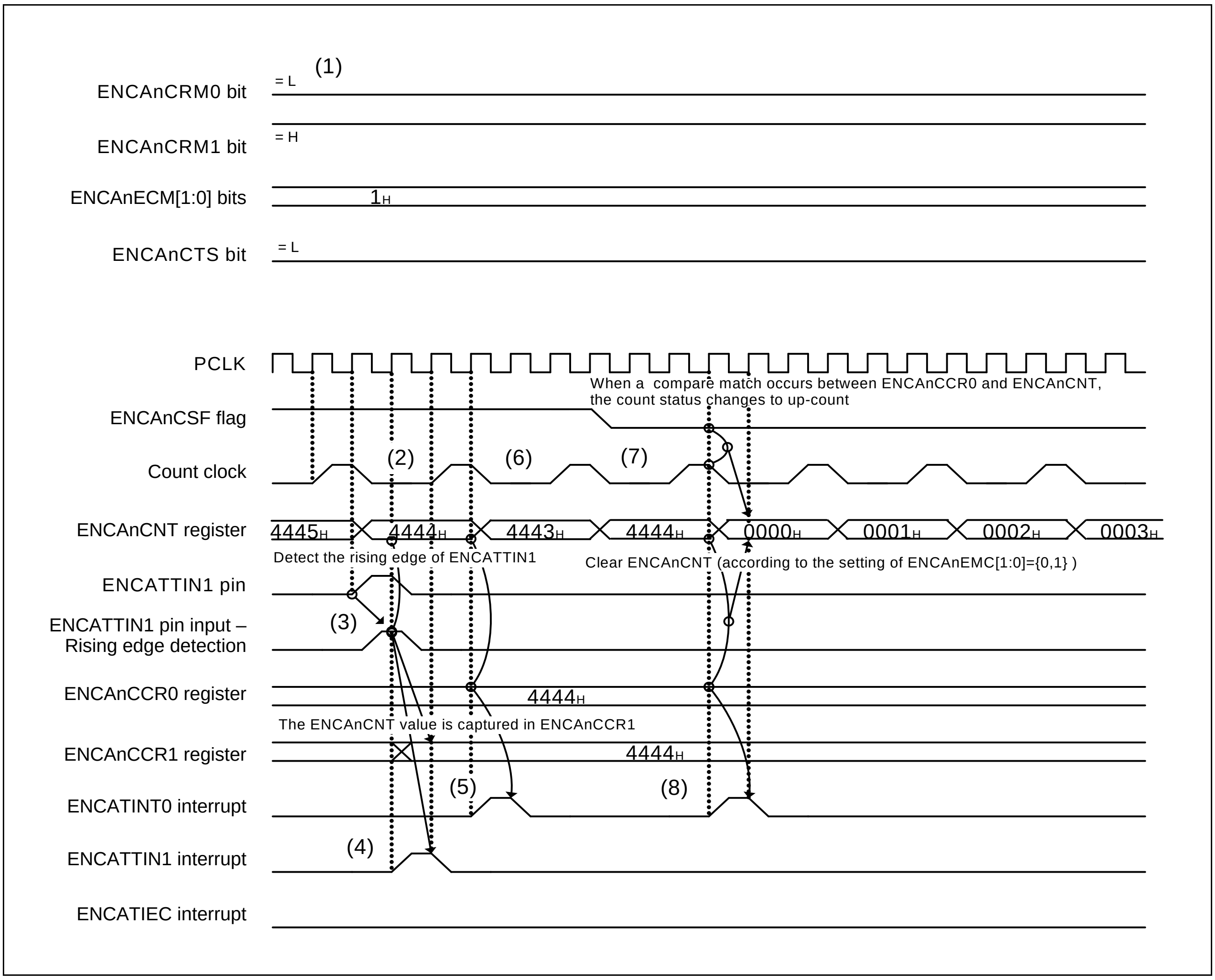


Figure 35.26 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) The rising edge of the ENCATTIN1 input is detected, and the ENCAnCNT value (4444_H) is captured in the ENCAnCCR1 register.

(4) An interrupt signal (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register is output.

(5) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

(6) The count operation changes to up-count.

(7) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(8) When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

35.6.16 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

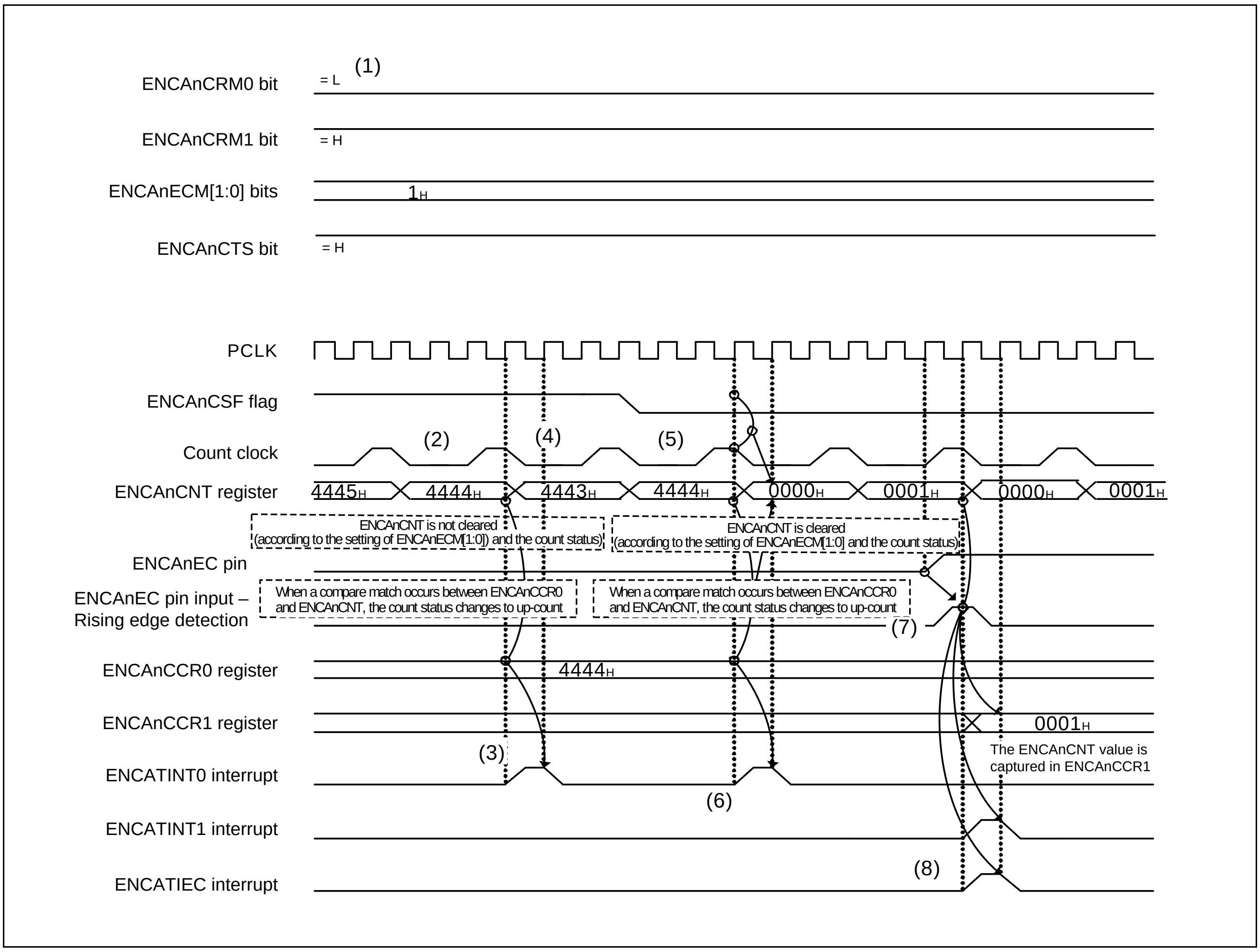


Figure 35.27 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 1.

(2) A down-count is performed.

(3) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare/capture interrupt (ENCATINT0) is output.

(4) The count operation changes to up-count.

(5) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(6) When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

(7) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCAnCNT value (0001_H) is captured in ENCAnCCR1 by detecting the rising edge of the ENCAnEC signal, and the counter is cleared to 0000_H.

(8) An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCAnEC are output.

35.6.17 Encoder Operation when Compare Match Clear Control is Disabled

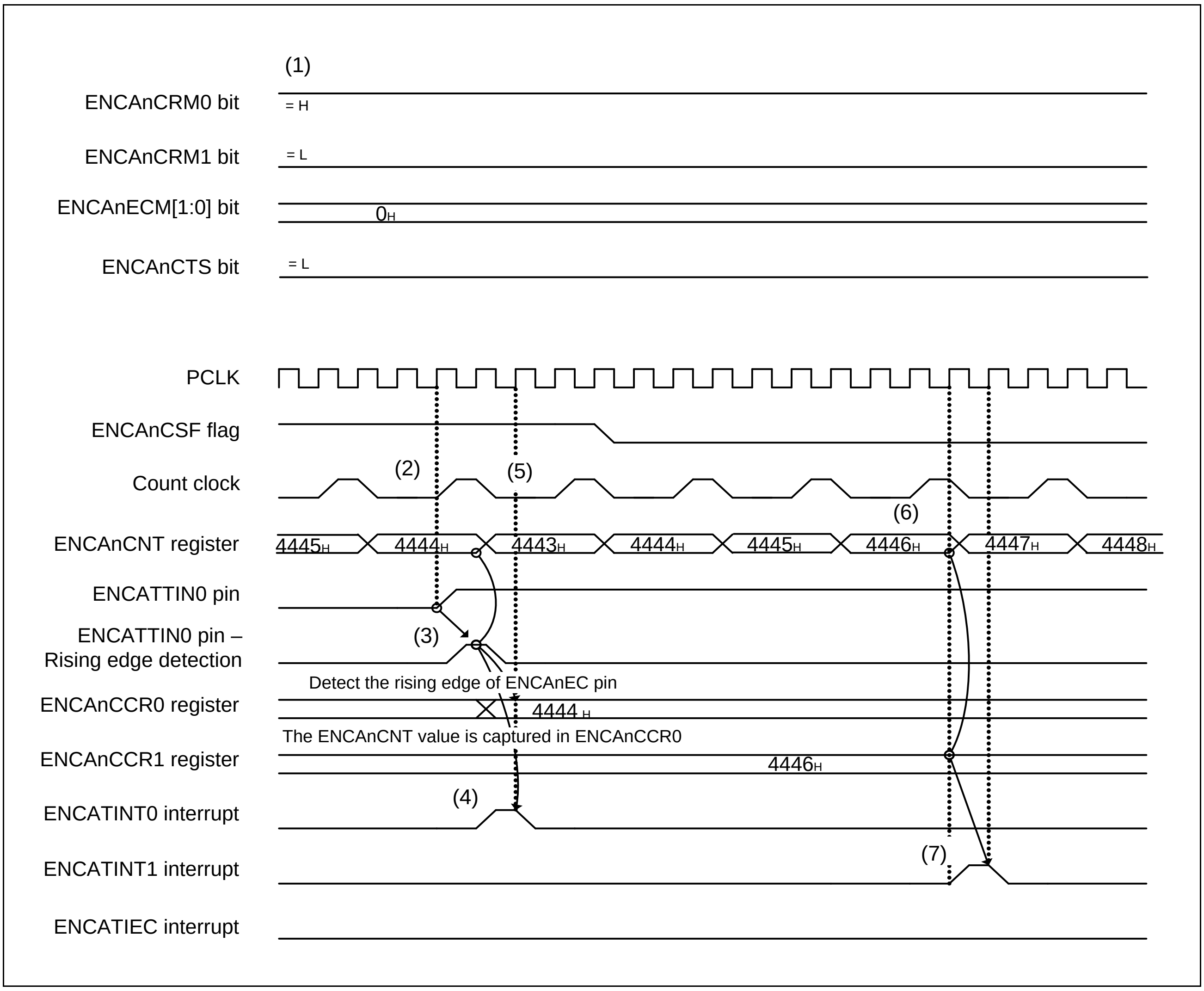


Figure 35.28 Encoder Operation when Compare Match Clear Control is Disabled

(1) The values are set as follows: ENCAnCCR1 = 4446_H, ENCAnCRM0 = 1, ENCAnCRM1 = 0, ENCAnECM[1:0] = 00_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) When the rising edge of ENCATTIN0 is detected, the ENCAnCNT value (4444_H) is captured in ENCAnCCR0.

(4) An interrupt signal (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.

(5) The count operation changes to up-count.

(6) When ENCAnCNT changes to 4446_H, a compare match with ENCAnCCR1 is detected.

(7) A compare match interrupt (ENCATINT1) with ENCAnCCR1 is output.

35.6.18 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, ENCAnE1 when ENCAnSCE = 1

35.6.18.1 Accompanying Capture Operation

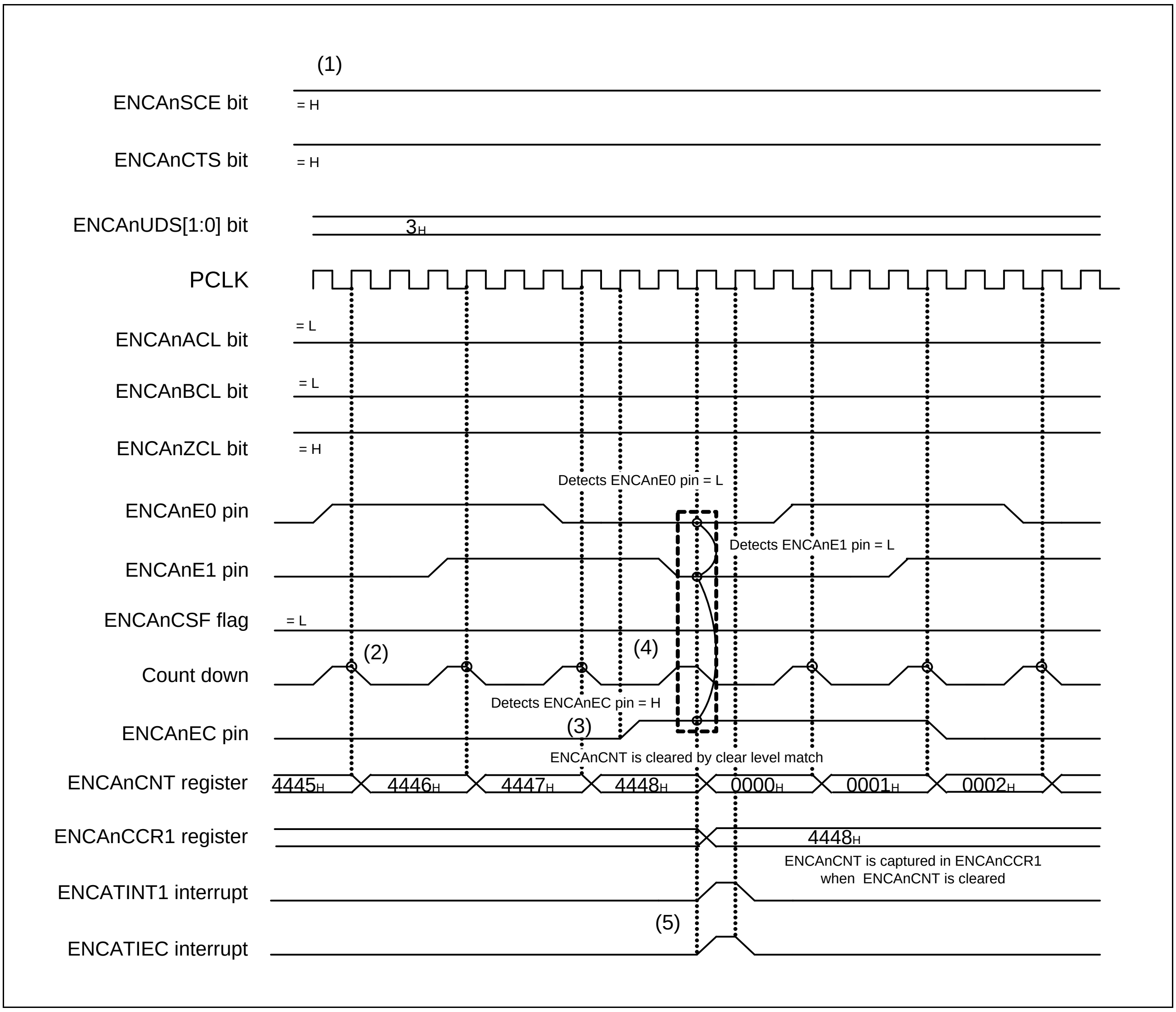


Figure 35.29 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, ENCAnE1 when ENCAnSCE = 1

(1) The values are set as follows: ENCAnSCE = 1, ENCAnCTS = 1, ENCAnUDS[1:0] =11_B, ENCAnACL = 0, ENCAnBCL = 0, and ENCAnZCL = 1.

(2) An up-count is performed.

(3) The count value is not cleared upon the rising edge of ENCAnEC.

(4) When ENCAnE0, ENCAnE1 and ENCAnEC reach the set clear level, the count value is cleared. The count value is captured in ENCAnCCR1 at the time of the clearing.

(5) At the time of the clearing, an interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCAnEC are output.

35.6.18.2 When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

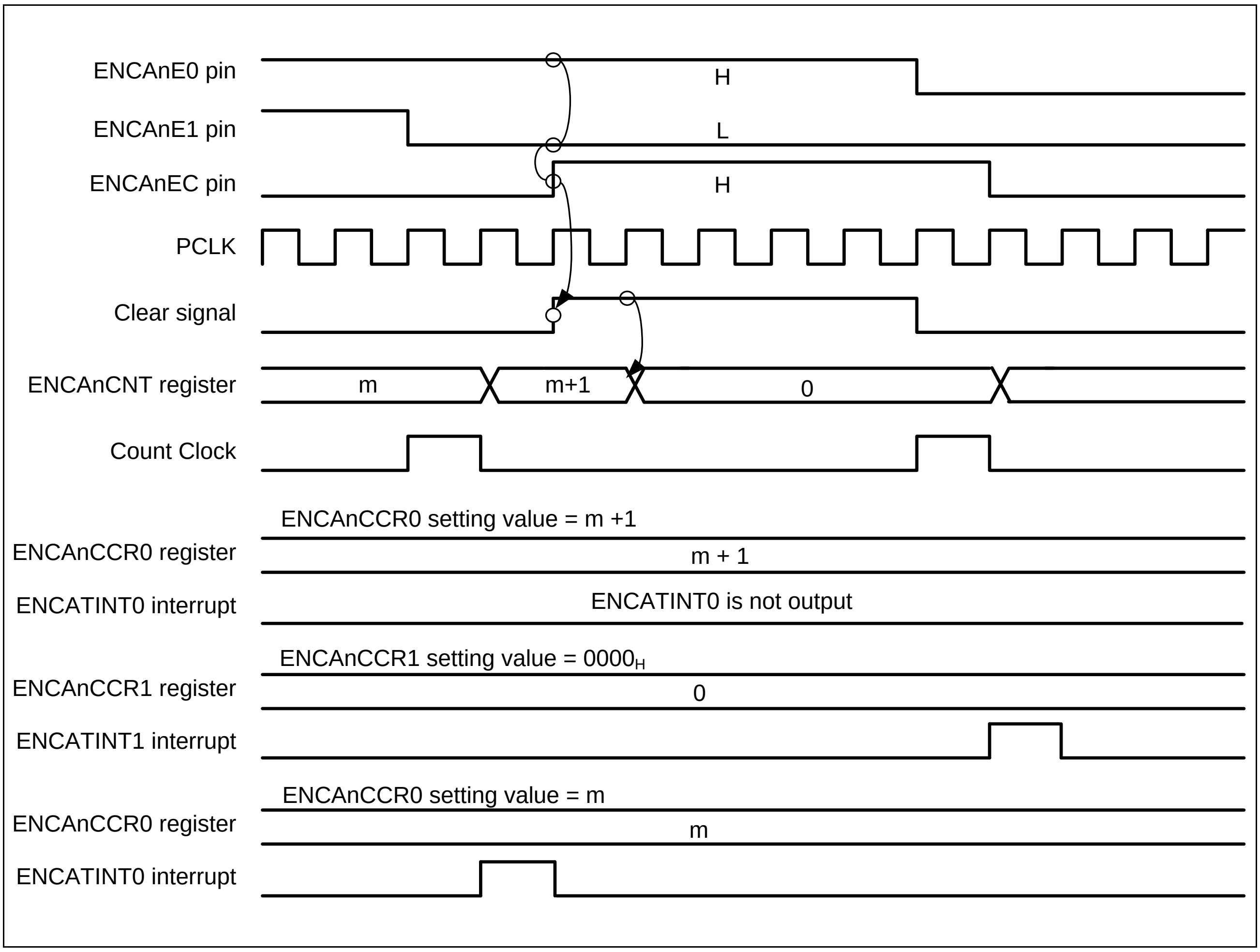


Figure 35.30 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count

35.6.18.3 When the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

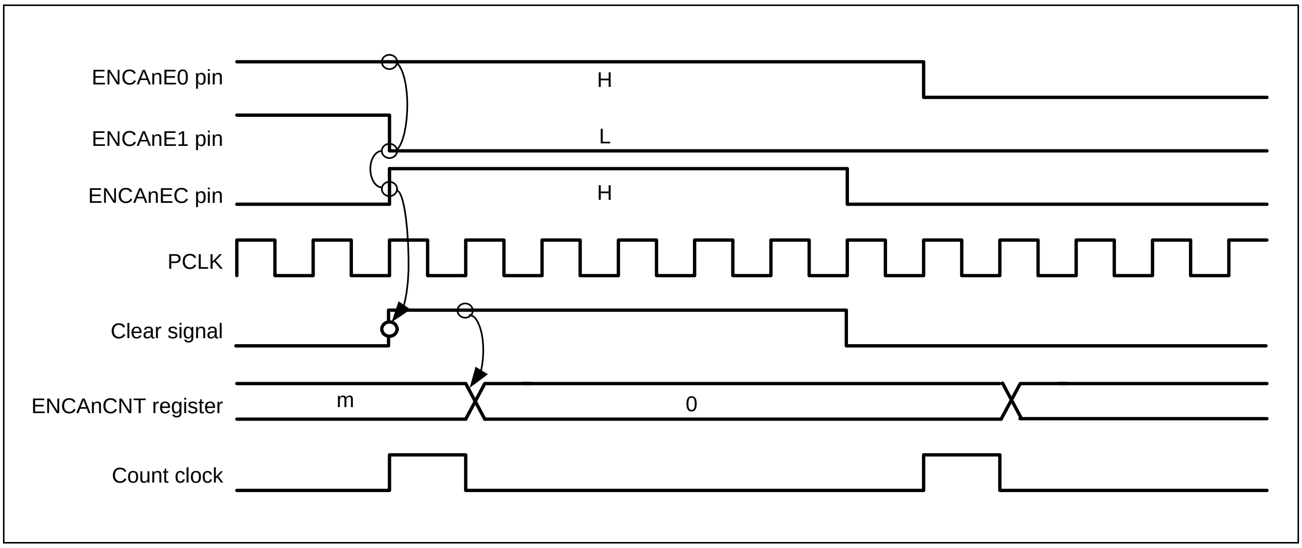


Figure 35.31 Clearing Timing for when the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count

35.6.18.4 When the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

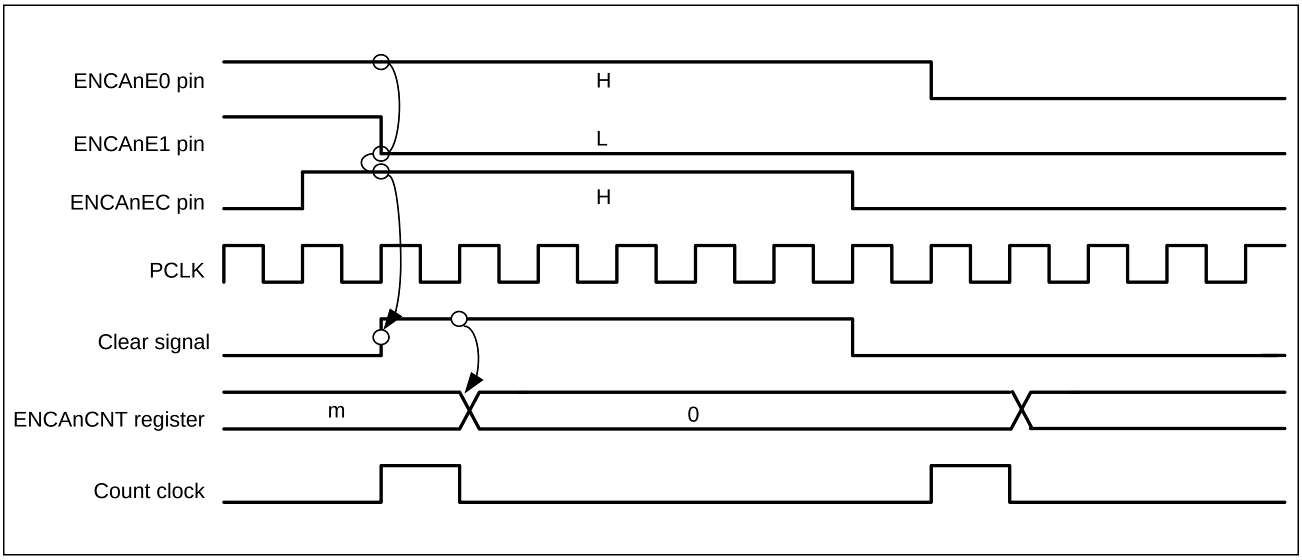


Figure 35.32 Clearing Timing for when the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count

35.6.18.5 When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

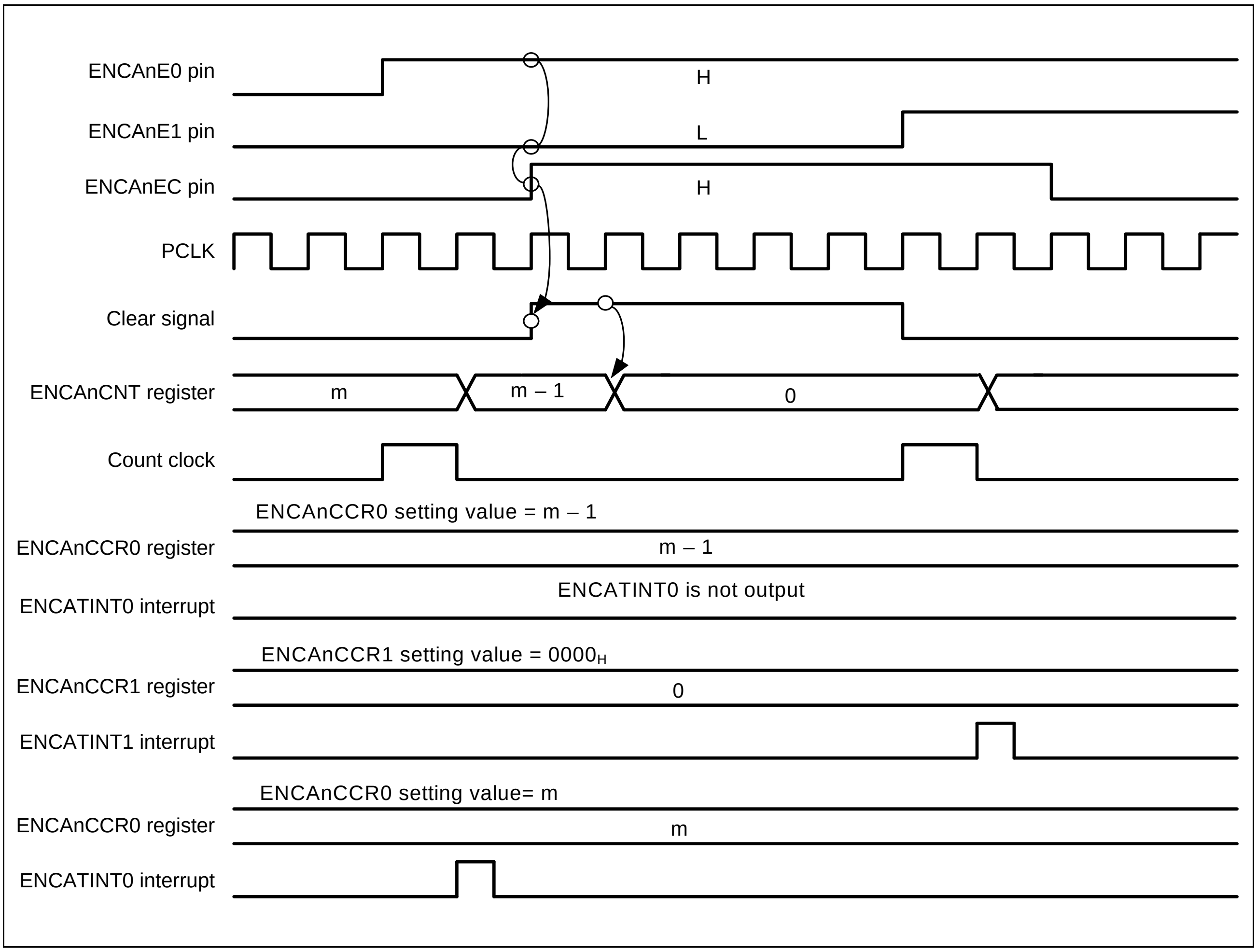


Figure 35.33 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count

35.6.19 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

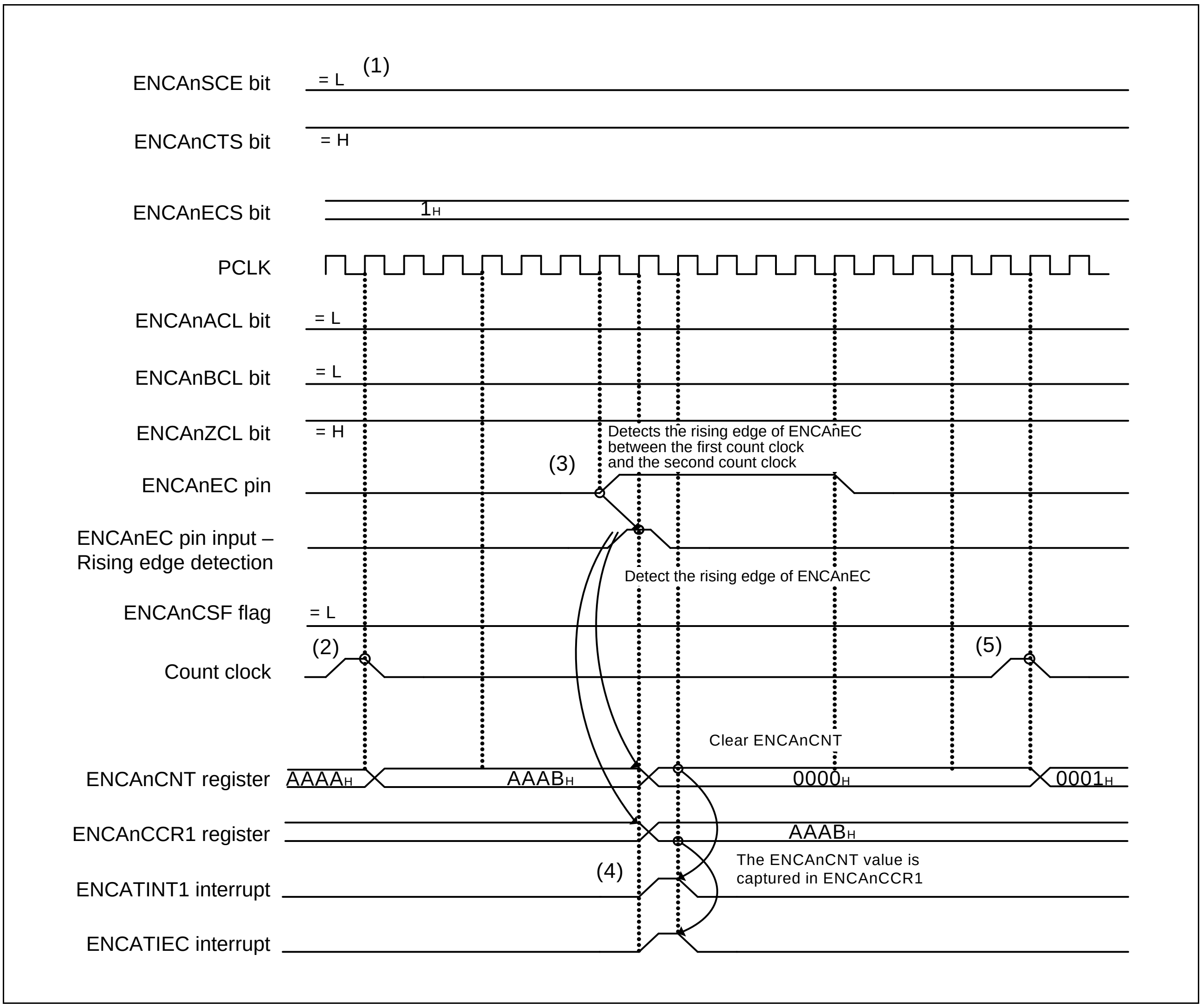


Figure 35.34 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

(1) The values are set as follows: ENCAnSCE = 0, ENCAnCTS = 1, and ENCAnECS[1:0] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCAnEC input is detected, and the ENCAnCNT value ($AAAB_H$) is captured in the ENCAnCCR1 register. Concurrently, clear operation by ENCAnEC is performed, and ENCAnCNT is cleared to 0000_H.

(4) A capture interrupt 1 (ENCATINT1) to the ENCAnCCR1 register and an encoder clear interrupt (ENCATIEC) by ENCAnEC are output.

(5) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

Section 36 Motor Control

This section contains a generic description of the Motor Control.

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the Motor Control.

36.1 Features of RH850/F1KH, RH850/F1KM Motor Control

36.1.1 Number of Units

Motor control function comprises the timer motor control units (TAPA) and the peripheral interconnection unit (PIC) to connect the TAPA unit to peripheral timers, and generates motor control waveforms by using a combination of peripheral timers and A/D converters.

This microcontroller has the following number of TAPA and PIC units.

Table 36.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
TAPA			
Number of Units	1	1	1
Name	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)
PIC			
Number of Units	1	1	1
Name	PIC0	PIC0	PIC0

Table 36.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
TAPA					
Number of Units	1	1	1	1	1
Name	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)
PIC					
Number of Units	1	1	1	1	1
Name	PIC0	PIC0	PIC0	PIC0	PIC0

Table 36.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
TAPA				
Number of Units	1	1	1	1
Name	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)	TAPAn (n = 0)
PIC				
Number of Units	1	1	1	1
Name	PIC0	PIC0	PIC0	PIC0

Table 36.4 Indices (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the unit of a timer and A/D converter used by TAPA and the motor control function is identified by the index “n”; for example, TAPAnCTL0 (n = 0) is TAPAn control register 0.
m	The channel of a used timer and A/D converter is identified by the index “m”. For example, the TAUDn channel is described as CHm.
x	The scan group of an A/D converter is identified by the index “x” (x = 1 to 3).
j	The scan trigger number of an A/D converter is identified by the index “j” (j = 0 to 2).

The following table lists the values indicated by the indices of each product.

Table 36.5 Indices of Products (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Indices of Each Product
All Products
m = 0 to 15 (e.g. TAUDn)
x = 1 to 3
j = 0 to 2

36.1.2 Register Base Addresses

Base addresses of TAPAn and PIC0 are listed in the following table.

Register addresses of TAPAn and PIC0 are given as offsets from the base addresses.

Table 36.6 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<TAPA0_base>	FFE9 0000$_H$
<PIC0_base>	FFDD 0000$_H$

36.1.3 Clock Supply

The TAPAn and PIC0 clock supplies are listed in the following table.

Table 36.7 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
TAPAn	PCLK	CKSCLK_IPERI1	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock
PIC0	PCLK	CKSCLK_IPERI1	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI1	Bus clock

36.1.4 Interrupt Requests

TAPAn interrupt requests are listed in the following table.

Table 36.8 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAPA0			
INTTAPA0IPEK0	Peak interrupt 0	16, 116	—
INTTAPA0IVLY0	Valley interrupt 0	17, 117	—

36.1.5 Reset Sources

Reset sources of TAPAn and PIC0 are listed in the following table. TAPAn and PIC0 are initialized by these reset sources.

Table 36.9 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
TAPAn	All reset sources (ISORES)
PIC0	All reset sources (ISORES)

36.1.6 External Input/Output Signals

External output signals of TAPAn and PIC0 are listed below.

Table 36.10 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
PIC0		
TOUTU	Motor control output U phase (positive)	TAPA0UP
TOUTUB	Motor control output U phase (negative)	TAPA0UN
TOUTV	Motor control output V phase (positive)	TAPA0VP
TOUTVB	Motor control output V phase (negative)	TAPA0VN
TOUTW	Motor control output W phase (positive)	TAPA0WP
TOUTWB	Motor control output W phase (negative)	TAPA0WN
TAPA0		
TAPATHASIN	Motor control output Hi-Z control input	TAPA0ESO

CAUTION

For the port pins that are used as TAPA0UP, TAPA0UN, TAPA0VP, TAPA0VN, TAPA0WP and TAPA0WN, set the output driver strength to high (PDSCn_m = 1).

36.1.7 Internal Output Signals

The internal output signals of TAPAn and PIC0 are listed below.

Table 36.11 Internal Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
TAPA0		
$\overline{\text{TAPATHZOUT0}}$	TAPA0UP/TAPA0UN output buffers Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT1}}$	TAPA0VP/TAPA0VN output buffers Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT2}}$	TAPA0WP/TAPA0WN output buffers Hi-Z control output*1	Port
TAPATADOUT0	A/D trigger signal 0 output*2	ADCA0 hardware trigger expansion
TAPATADOUT1	A/D trigger signal 1 output*2	ADCA0 hardware trigger expansion
PIC0		
TAPATHASIN	TAPA0 asynchronous Hi-Z control signal*1,*3	TAPA0
TAPATSIM0	TAUD master channel interrupt signal (TAUD0: INTTAUD0I0,INTTAUD0I2, INTTAUD0I8)	TAPA0
TAPATUDCM0	TAUD master up/down signal (TAUD0: TAUD0UDC0, TAUD0UDC2, TAUD0UDC8)	TAPA0
TAPATCDENS0	TAUD slave 0 match detect*4 (ADCA0 hardware trigger expansion: ADOPA1ADCATTIN00)	TAPA0
TAPATCDENS1	TAUD slave 1match detect*4 (ADCA0 hardware trigger expansion: ADOPA2ADCATTIN00)	TAPA0

Note 1. See **Section 36.4.6, TAPA0 Hi-Z Control Input Selection** for details.

Note 2. These signals can be used to as a trigger source to start the A/D converter. See **Table 38.78, List of A/D Conversion Hardware Triggers**.

Note 3. This input signal is passed through a noise filter. See **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector**, **Section 2C.12, Noise Filter & Edge/Level Detector**, **Section 2A.13, Description of Port Noise Filter & Edge/Level Detection**, **Section 2B.13, Description of Port Noise Filter & Edge/Level Detection**, and **Section 2C.13, Description of Port Noise Filter & Edge/Level Detection**.

Note 4. These signals are selected by the H/W trigger selection bit of the ADCA0 A/D converter. See **Table 38.78, List of A/D Conversion Hardware Triggers**.

36.2 Overview

36.2.1 Functional Overview

The motor control function provides the following functions by combining the motor control unit (TAPA) and Timer Array Unit D (TAUDn) or A/D (ADCAn):

- Asynchronous Hi-Z control function
  Hi-Z control for TAUDn output can be performed by using pin input or error signals.
- Interrupt signal output function
  Request signals for two types of interrupts, peak interrupts and valley interrupts, can be output by the INTn signals output by TAUDn.
- A/D conversion start trigger selection function
  An A/D conversion start trigger can be output by the INTn signals output by TAUDn.

Additionally, the motor control function provides also the following functions by combining the motor control unit (TAPA) and the peripheral interconnection (PIC):

- Timer simultaneous start trigger function
  The respective channels of TAUD0 and TAUJ1, and the ENCAn timer can be started simultaneously.
- Trigger and pulse width measuring function
  Measurement of trigger periods can be performed by inputting ENCAn interrupt signals to TAUDn or TAUJ1.
- A/D trigger encoder capture function
  The value of the ENCAn counter can be captured at the A/D conversion start trigger timing.
- Three-phase PWM output with dead time / High-accuracy triangle PWM output with dead time
  Three-phase PWM output with dead time can be performed by TAUDn.
- Delay pulse output with dead time
  Delay pulses (with dead time) for the cycle timing can be output.

36.2.2 Basic Structure of Motor Control

The peripheral block configuration of the motor control function is shown below.

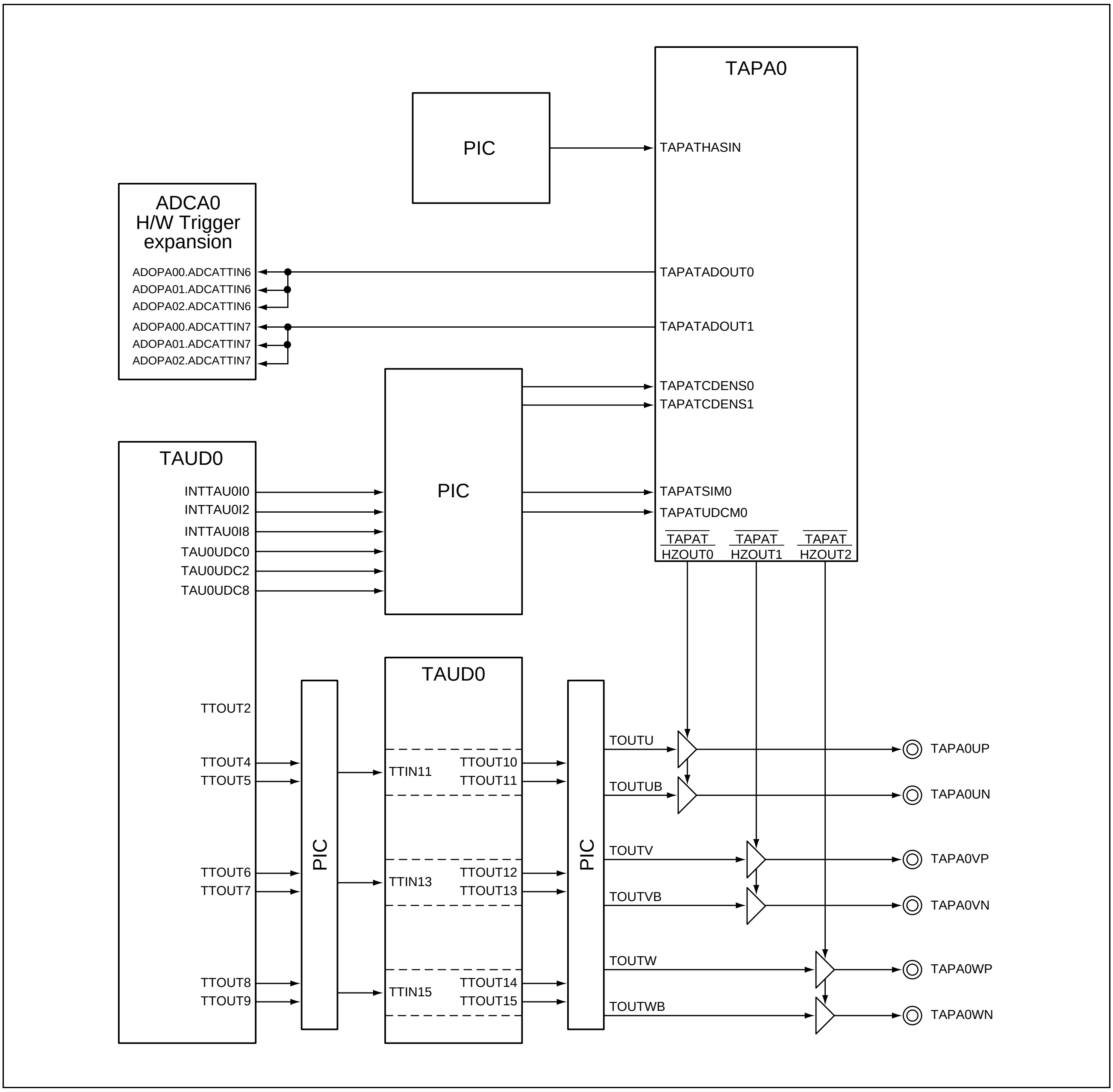


Figure 36.1 Configuration of Motor Control

TAUDn and PIC are used to generate the motor control output signals (three-phase PWM output signals with dead time).

The timer control unit (TAPA) performs Hi-Z control for the motor control output.

Additionally, PIC can provide functions specific to the motor by combining respective channels of TAUDn and TAUJ1, ENCAn, and TAPA.

36.2.3 Block Diagram

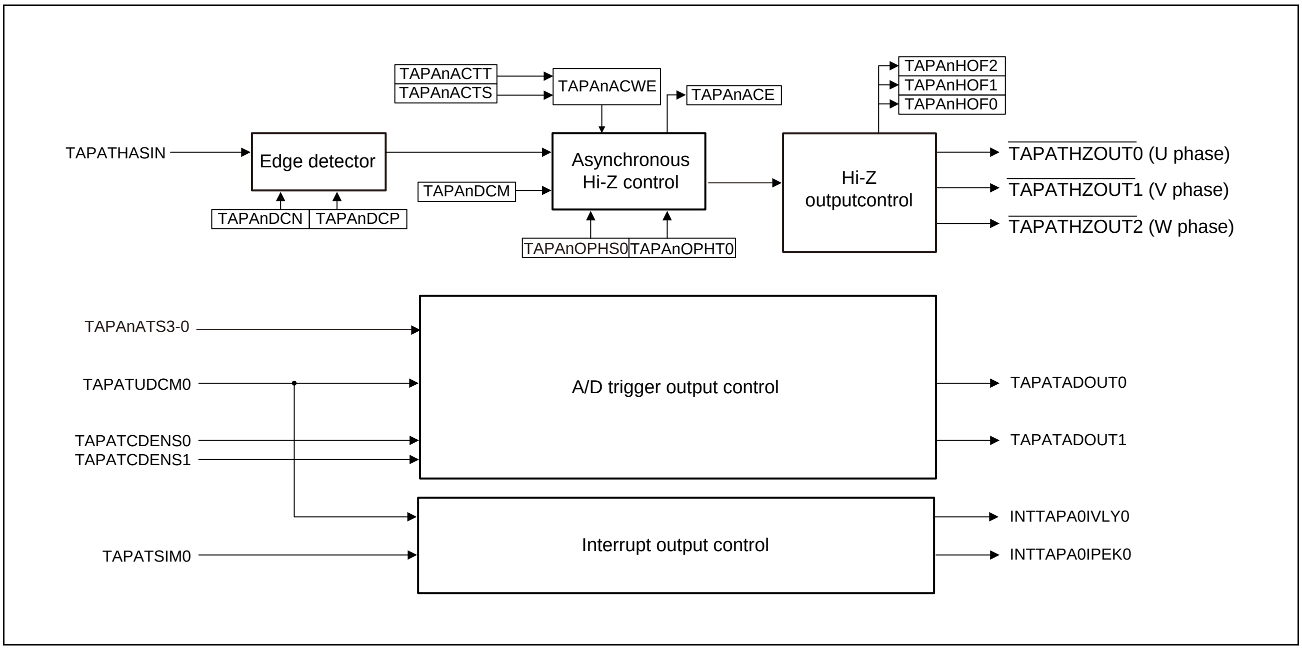


Figure 36.2 TAPA Peripheral Block Diagram

NOTE

For the PIC peripheral block diagram, see the respective section describing each function.

36.2.4 Definition of Terms

Peak and valley interrupts - Peak and valley of timer counter

In this document, the period from a TAUD counting-up status to generation of INT from the master channel is defined as a peak period, and this INT is defined as a peak interrupt.

In contrast, the period from a TAUD counting-down status to generation of INT from the master channel is defined as a valley period, and this INT is defined as a valley interrupt.

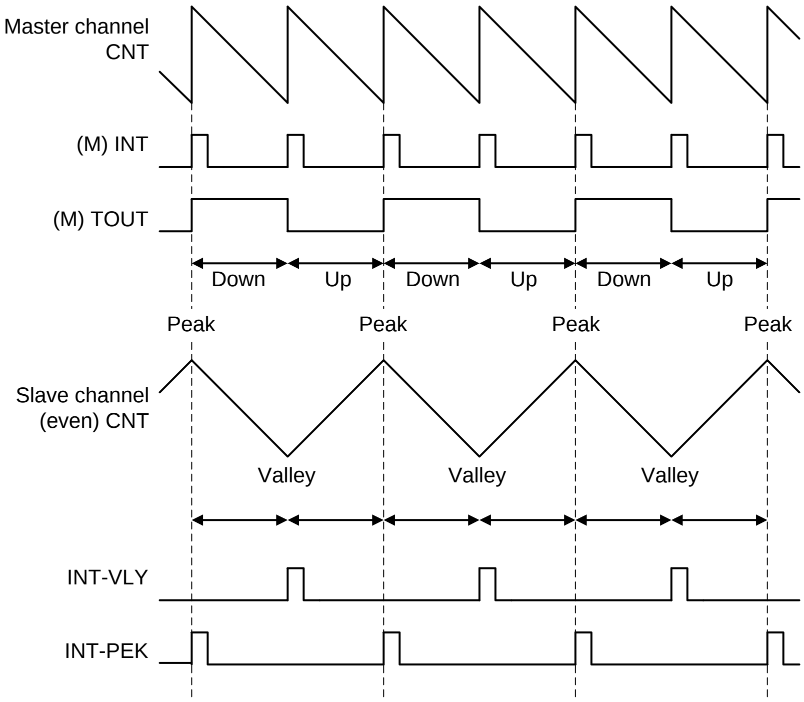


Figure 36.3 Peak and Valley Interrupts

36.3 Registers

36.3.1 List of Registers

The registers of TAPAn and PIC0 are listed in the following table.

For details about <TAPAn_base> and <PIC0_base>, see **Section 36.1.2, Register Base Addresses**.

Table 36.12 List of Registers

Module Name	Register Name	Symbol	Address
TAPAn	TAPA control register 0	TAPAnCTL0	<TAPAn_base> + 20_H
	TAPA control register 1	TAPAnCTL1	<TAPAn_base> + 24_H
	TAPA flag register	TAPAnFLG	<TAPAn_base> + 00_H
	TAPA asynchronous Hi-Z control write enable register	TAPAnACWE	<TAPAn_base> + 04_H
	TAPA asynchronous Hi-Z control start trigger register	TAPAnACTS	<TAPAn_base> + 08_H
	TAPA asynchronous Hi-Z control stop trigger register	TAPAnACTT	<TAPAn_base> + $0C_H$
	TAPA Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base> + 14_H
	TAPA Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base> + 18_H
	TAPA emulation register	TAPAnEMU	<TAPAn_base> + 28_H
PIC0	Simultaneous start trigger control register	PIC0SST	<PIC0_base> + 04_H
	Simultaneous start control register 0	PIC0SSER0	<PIC0_base> + 10_H
	Simultaneous start control register 2	PIC0SSER2	<PIC0_base> + 18_H
	Hi-Z output control register 0	PIC0HIZCEN0	<PIC0_base> + 80_H
	A/D conversion trigger output control register 400	PIC0ADTEN400	<PIC0_base> + 90_H
	A/D conversion trigger output control register 401	PIC0ADTEN401	<PIC0_base> + 94_H
	A/D conversion trigger output control register 402	PIC0ADTEN402	<PIC0_base> + 98_H
	Timer I/O control register 200	PIC0REG200	<PIC0_base> + $C0_H$
	Timer I/O control register 201	PIC0REG201	<PIC0_base> + $C4_H$
	Timer I/O control register 202	PIC0REG202	<PIC0_base> + $C8_H$
	Timer I/O control register 203	PIC0REG203	<PIC0_base> + CC_H
	Timer I/O control register 30	PIC0REG30	<PIC0_base> + $E8_H$
	Timer I/O control register 31	PIC0REG31	<PIC0_base> + EC_H

NOTE

For details about PIC-related registers, see the respective section describing each function.

36.3.2 TAPAnCTL0 — TAPA Control Register 0

This register is used to set up the asynchronous Hi-Z control function.

The values of this register can be rewritten only when TAPAnFLG.TAPAnACE is 0 and TAUDnTEm for the corresponding TAUD's master channel is 0.

Access: This register can be read or written in 16-bit units.

Address: <TAPAn_base> + 20_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TAPAn DCM	TAPAn DCN	TAPAn DCP	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 36.13 TAPAnCTL0 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 5</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>4</td><td>TAPAnDCM</td><td>Clear Condition Configuration
This control bit specifies the clear conditions for Hi-Z control output.
0: Enables manipulation of TAPAnOPHT0 regardless of the TAPATHASIN signal input level.
1: Enables manipulation of TAPAnOPHT0 only if the TAPATHASIN signal input is inactive.</td></tr>
<tr><td>3, 2</td><td>TAPAnDCN,
TAPAnDCP</td><td>Hi-Z Input Edge Selection
These are control bits that specify the valid edge of TAPATHASIN.
<table>
<tr><th>TAPAn DCN</th><th>TAPAn DCP</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Does not detect valid edges.</td></tr>
<tr><td>0</td><td>1</td><td>Detects a rising edge as the valid edge (active level = high).</td></tr>
<tr><td>1</td><td>0</td><td>Detects a falling edge as the valid edge (active level = low).</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
</table>

36.3.3 TAPAnCTL1 — TAPA Control Register 1

This register is used to specify the A/D conversion trigger.

Access: This register can be read or written in 8-bit units.

Address: <TAPAn_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAPAnATS3	TAPAnATS2	TAPAnATS1	TAPAnATS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 36.14 TAPAnCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	TAPAnATS3, TAPAnATS2	A/D Conversion Trigger 1 Selection These are control bits that specify A/D conversion trigger output 1 (TAPATADOUT1).

TAPAn ATS3	TAPAn ATS2	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 signal

Bit Position	Bit Name	Function
1, 0	TAPAnATS1, TAPAnATS0	A/D Conversion Trigger 0 Selection These are control bits that specify the A/D conversion trigger output 0 (TAPATADOUT0).

TAPAn ATS1	TAPAn ATS0	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt INTTAPA0IVLY0 signal while the triangle wave is rising (counting up) or falling (counting down)

36.3.4 TAPAnFLG — TAPA Flag Register

This flag register is for asynchronous Hi-Z control.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAPAn_base> + 00_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TAPAn HOF2	TAPAn HOF1	TAPAn HOF0	—	—	—	—	—	—	—	TAPAn ACE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.15 TAPAnFLG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned.
10	TAPAnHOF2	W phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT2 is high level 1: The present output TAPAnTHZOUT2 is low level.
9	TAPAnHOF1	V phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT1 is high level 1: The present output TAPAnTHZOUT1 is low level.
8	TAPAnHOF0	U phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT0 is high level 1: The present output TAPAnTHZOUT0 is low level.
7 to 1	Reserved	When read, the value after reset is returned.
0	TAPAnACE	Asynchronous Hi-Z Control Enable 0: Indicates that the asynchronous Hi-Z control is stopped. 1: Indicates that the asynchronous Hi-Z control is enabled. The conditions for setting or clearing this bit are as follows: Clear condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1 Set condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1

36.3.5 TAPAnACWE — TAPA Asynchronous Hi-Z Control Write Enable Register

This register is used to enable writing for asynchronous Hi-Z control.

Access: This register can be read or written in 8-bit units.

Address: <TAPAn_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACWE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 36.16 TAPAnACWE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TAPAnACWE	Asynchronous Control Write Enable This is a write enable bit for asynchronous Hi-Z control. After 1 is written, this bit is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

36.3.6 TAPAnACTS — TAPA Asynchronous Hi-Z Control Start Trigger Register

This register is used to enable the start trigger for asynchronous Hi-Z control.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.17 TAPAnACTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTS	Asynchronous Hi-Z Control Start Trigger This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Enables asynchronous Hi-Z control when TAPAnACWE is 1.

36.3.7 TAPAnACTT — TAPA Asynchronous Hi-Z Control Stop Trigger Register

This bit enables the stop trigger for asynchronous Hi-Z control.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.18 TAPAnACTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTT	Asynchronous Hi-Z Control Stop Trigger This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Disables asynchronous Hi-Z control when TAPAnACWE is 1.

36.3.8 TAPAnOPHS — TAPA Hi-Z Start Trigger Register

This software trigger register is used to start Hi-Z control for motor control output pins.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.19 TAPAnOPHS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHS0	Hi-Z Control Start Trigger This bit starts Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Starts Hi-Z control.

36.3.9 TAPAnOPHT — TAPA Hi-Z Stop Trigger Register

This software trigger register is used to stop Hi-Z control for motor control output pins.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.20 TAPAnOPHT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHT0	Hi-Z Control Stop Trigger This bit stops Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Stops Hi-Z control. Whether the setting of this bit is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

36.3.10 TAPAnEMU — TAPA Emulation Register

This register controls SVSTOP for emulation.

Access: This register can be read or written in 8-bit units. (when EPC.SVSTOP = 0, rewritten only)

Address: <TAPAn_base> + 28_H

Value after reset: Reading this register returns always 00_H.

Bit	7	6	5	4	3	2	1	0
	TAPAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 36.21 TAPAnEMU Register Contents

Bit Position	Bit Name	Function
7	TAPAnSVSDIS	This bit is used to control disabling of SVSTOP. 0: SVSTOP is valid. (Sets Hi-Z control output to low level when SVSTOP = 1 is input). 1: SVSTOP is invalid. (Hi-Z control output level does not change according to the level of SVSTOP input).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

36.4 Asynchronous Hi-Z Control Function

If the operation of the timer motor control function controlled by the MCU becomes abnormal, the rotation of the external motor also becomes abnormal. This function can forcibly set the motor control output to the Hi-Z state upon detection of abnormal motor operation, independent of MCU control.

36.4.1 Overview

This function forcibly stops TAPAn output through asynchronous Hi-Z control.

- When the TAPATHASIN signal becomes active, the levels of the motor control output pins are set to Hi-Z, and motor control output is forcibly stopped.
- Motor control output in a Hi-Z state can be resumed by writing the Hi-Z stop trigger register (TAPAnOPHT0).
- The Hi-Z state of motor control output can also be specified by writing the Hi-Z control start trigger register (TAPAnOPHS).
- Setting PIC can enable or disable Hi-Z control input when an error occurs.

36.4.2 System Configuration Example

A system configuration example is shown below, where an external error detection signal (the TAPA0ESO signal) is used for Hi-Z control of the motor control outputs (TAPA0UP / TAPA0UN / TAPA0VP / TAPA0VN / TAPA0WP / TAPA0WN).

When valid edges of the external error detection signal are detected, the level of the motor control outputs is set to Hi-Z.

Because the microcontroller might freeze when an error occurs, external error detection signals are continuously processed so that the motor control timer outputs can be set to Hi-Z even if no clock is supplied.

Note that an error is detected only when the valid edge of the error detection signal is detected. Therefore, no error is detected if the output level is fixed and the signal level does not change.

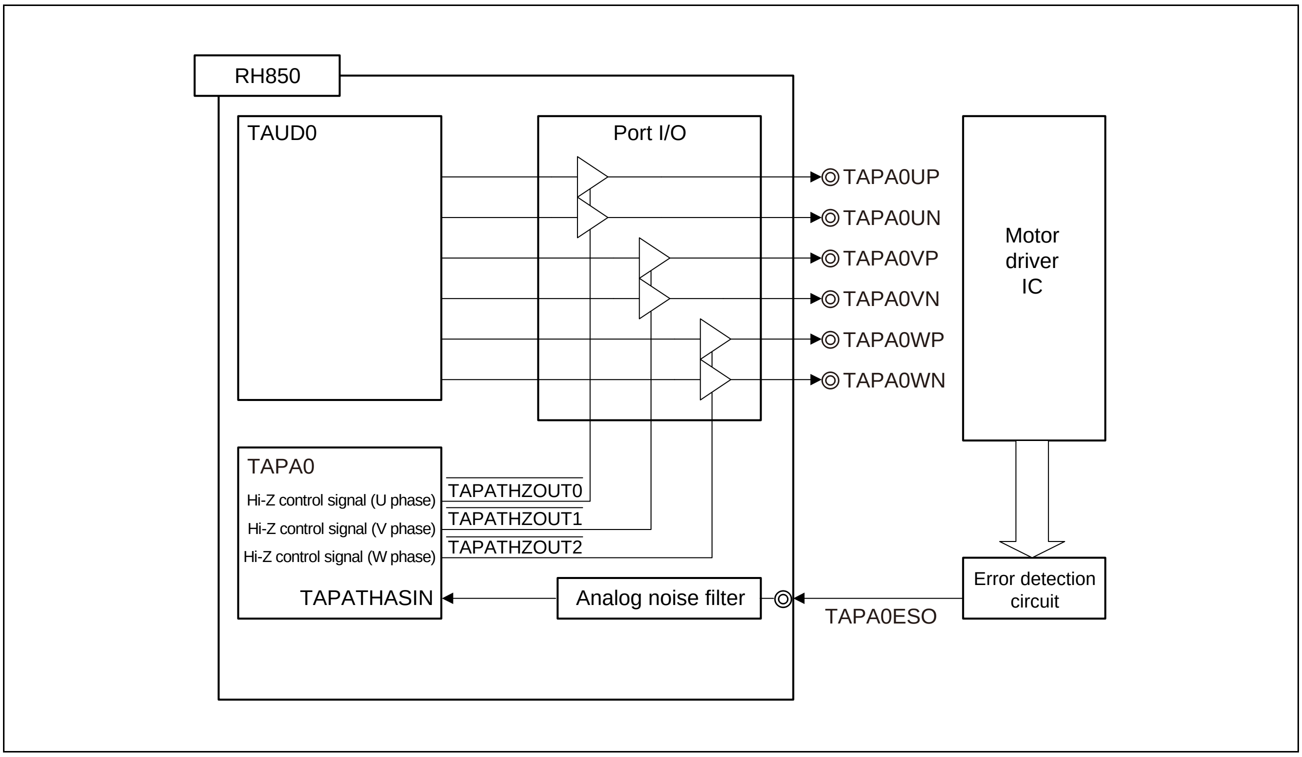


Figure 36.4 System Configuration Example of Asynchronous Hi-Z Control for Pin Input

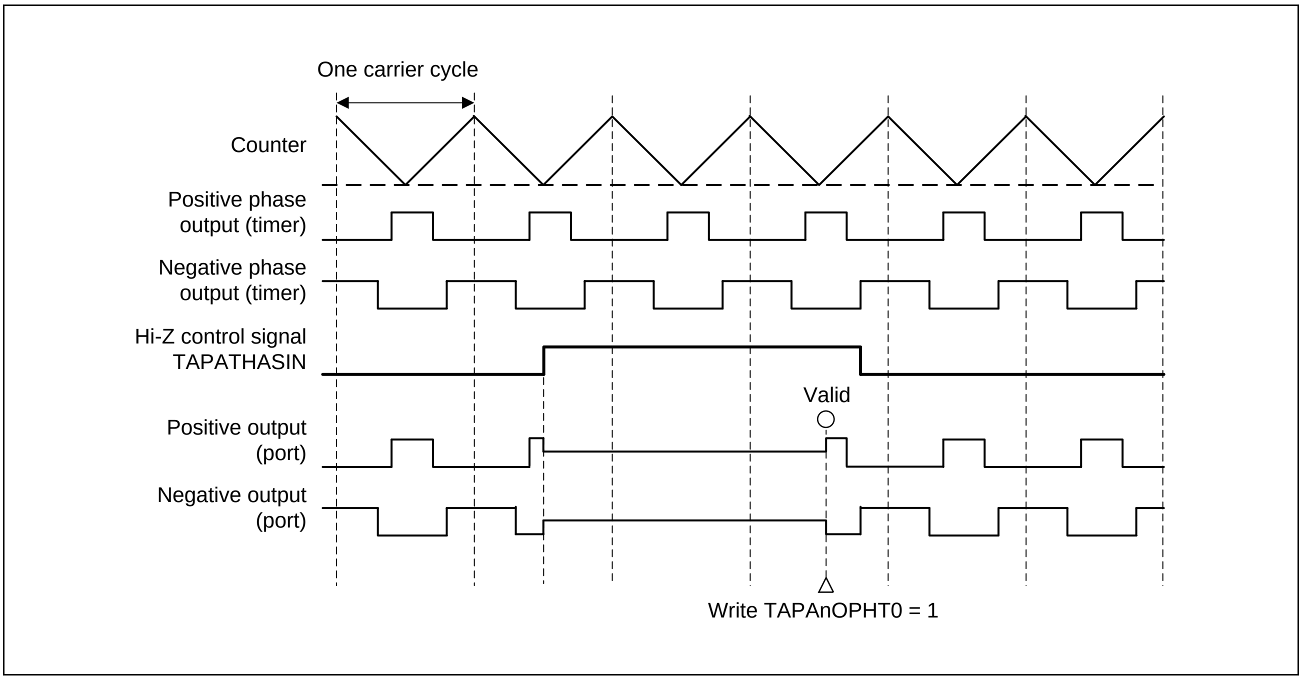
36.4.3 Basic Operation

Hi-Z control for motor control output pins can be started as follows:

- Detecting the valid edge of asynchronous Hi-Z control signal (TAPATHASIN)
- Setting the start trigger bit TAPAnOPHS.TAPAnOPHS0 of the Hi-Z control signal

The levels of the motor control output pins are set to Hi-Z until the stop trigger bit of the Hi-Z control signal (TAPAnOPHT.TAPAnOPHT0) is set. Note that whether the setting of TAPAnOPTH0 is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

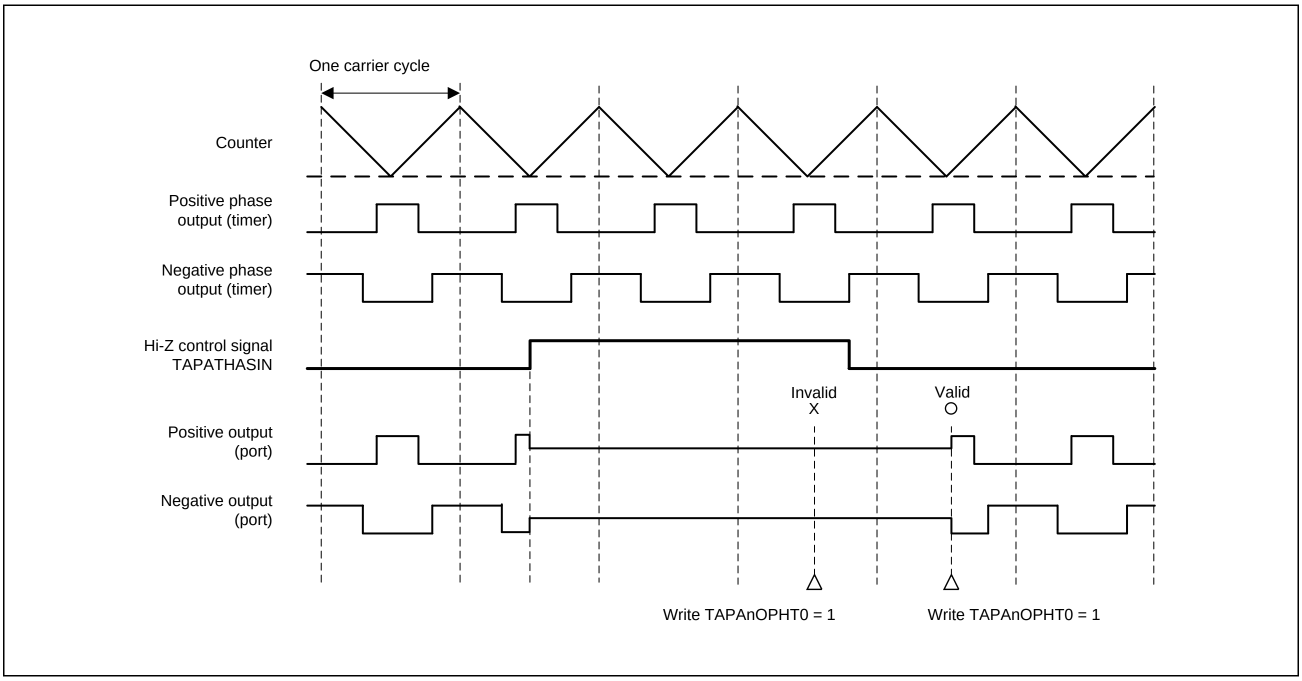
(1) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0, regardless of the level of TAPATHASIN.

(2) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

Writing 1 to the stop trigger bit (TAPAnOPHT.TAPAnOPHT0) of the Hi-Z control signal is ignored while TAPATHASIN is active (high level because TAPAnCTL0.TAPAnDCP is 1).

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0 after TAPATHASIN becomes inactive (low level because TAPAnCTL0.TAPAnDCP is 1).

36.4.4 Asynchronous Hi-Z Control Using Software Trigger

Hi-Z control for motor control output is possible by using the Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0 and Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0.

(1) Function of Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0

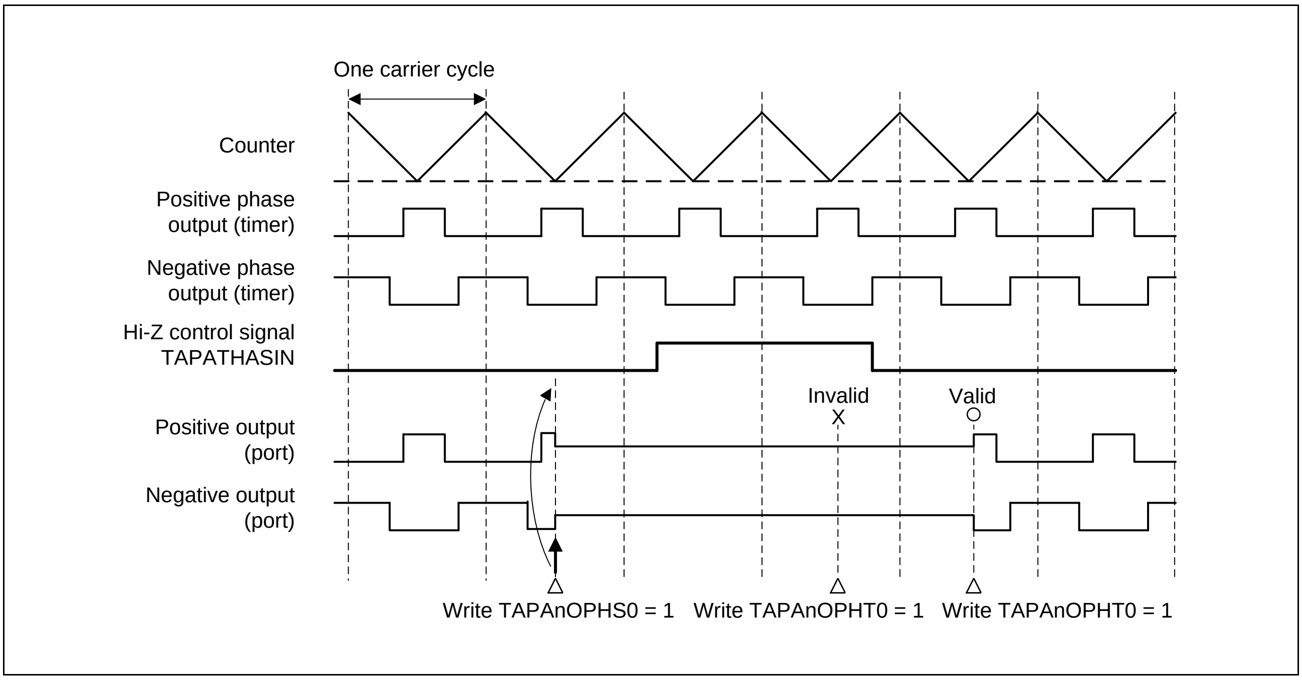
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the motor control output (Hi-Z output).

(2) Function of Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0

Whether the Hi-Z control stop trigger is valid or invalid depends on the conditions below:

TAPAnDCM	Function
0	Writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output.
1	If TAPATHASIN is inactive, writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output. If TAPATHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

(3) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control output (Hi-Z output) is forcibly stopped when 1 is written to TAPAnOPHS0.

After that, the level of the motor control output remains Hi-Z even if the rising edge of TAPATHASIN is detected.

Writing to TAPAnOPHT0 is ignored while TAPATHASIN is active (high level because TAPAnDCN is 0 and TAPAnDCP is 1).

After detection of the falling edge of TAPATHASIN, the motor control output restarts when 1 is written to TAPAnOPHT0 while TAPATHASIN is inactive (low level because TAPAnDCN is 0 and TAPAnDCP is 1).

36.4.5 Operating Procedure

The operating procedure for the asynchronous input Hi-Z control function is shown below:

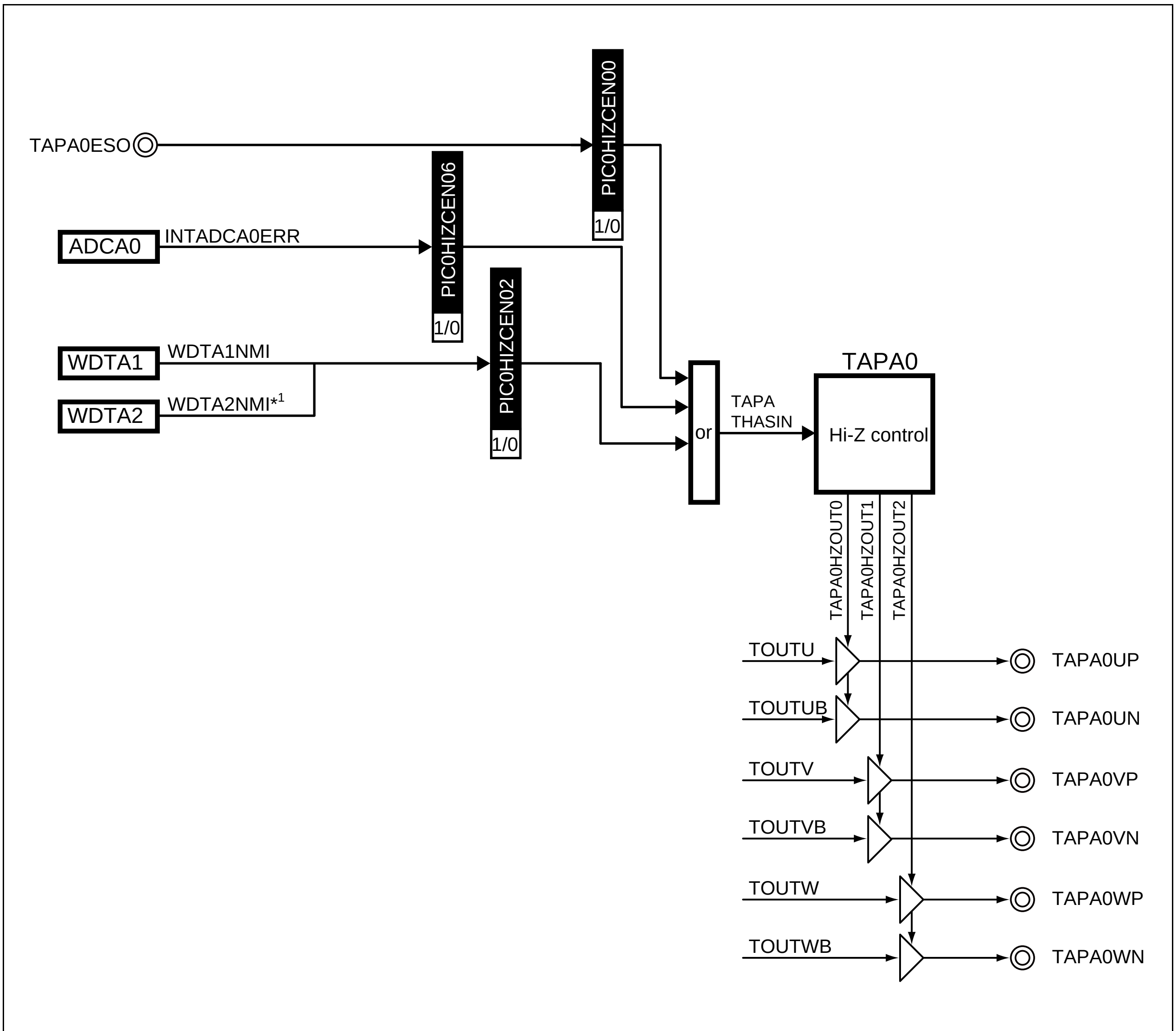
	Operation	Status of TAPA
Initial setup	Set up the TAPAnCTL0 register. Specify TAPAnDCP and TAPAnDCN to select the input edge. Specify TAPAnDCM to select the clear mode.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)
Start operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1.	Writing to TAPAnACTS is enabled.
	Set up the TAPAnACTS register. Set TAPAnACTS to 1.	Asynchronous Hi-Z control enabled (TAPAnFLG.TAPAnACE = 1)
During operation	Hi-Z control for the timer function outputs can be started by controlling the following: ● TAPAnOPHS register ● Asynchronous Hi-Z control signal (TAPATHASIN) Hi-Z control for the timer function outputs can be stopped by controlling the following: ● TAPAnOPHT register (If TAPAnDCM is 1, control by the TAPAnOPHT register is enabled only while TAPATHASIN is inactive.) The TAPA operating status can always be read using the TAPAnFLG register.	Hi-Z control for the motor control output pins is started by detecting the valid edge of the asynchronous Hi-Z control signal (TAPATHASIN) or by setting the Hi-Z control start trigger bit TAPAnOPHS0 to 1. Hi-Z control for the motor control output pins is stopped by setting the Hi-Z control stop trigger bit TAPAnOPHT0 to 1 according to the operation mode specified by the TAPAnDCM bit.
Stop operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1.	Writing to TAPAnACTT is enabled.
	Set up the TAPAnACTT register. Set TAPAnACTT to 1.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

Restart

36.4.6 TAPA0 Hi-Z Control Input Selection

In order to stop the motor control output in case of errors, error events are selected in PIC and the level of the motor control output is set to Hi-Z in TAPA0, as shown in the diagram below.

The TAPA function can be stopped by setting TAPA0ACTT = 01_H after setting PIC0HIZCEN0 = 00_H or TAPA0ACWE = 01_H.



Note 1. This function is supported in RH850/F1KH-D8.

Figure 36.5 Hi-Z Control Block Diagram

Switching into a Hi-Z state can be performed by the following:

- TAPA0ESO pin input
- A/D converter ADCA0 error signal ADCA0ERR

- RH850/F1KH-D8
  Window Watchdog Timer WDTA1 non maskable interrupt WDTA1NMI and Window Watchdog Timer WDTA2 non maskable interrupt WDTA2NMI

  RH850/F1KM-S4, RH850/F1KM-S1
  Window Watchdog Timer WDTA1 non maskable interrupt WDTA1NMI

For details about these signals, see the respective descriptions.

36.4.7 Registers

36.4.7.1 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control signal of TAPAn.

Access:	This register can be read or written in 8-bit units.
Address:	<PIC0_base> + 80_H
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 36.22 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Selects whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	RH850/F1KH-D8 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI and the WDTA2NMI interrupt signal. 0: Disable 1: Enable RH850/F1KM-S4, RH850/F1KM-S1 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Selects whether to enable or disable Hi-Z output control by the TAPA0ESO pin input. 0: Disable 1: Enable

36.5 INT Signal Output Selection Function

36.5.1 Configuration of the INT Signal Output Selection Function

This function generates the peak interrupt INTTAPA0IPEK0 and valley interrupt INTTAPA0IVLY0 by using the TAPATSIM0 signal, which is connected to the INT signal on the TAUD's triangular carrier cycle generation channel (master) and TAPATUDCM0 signal, which is connected to the counter up/down signal.

For the connection destination of TAPATSIM0, see **Section 36.1.7, Internal Output Signal**.

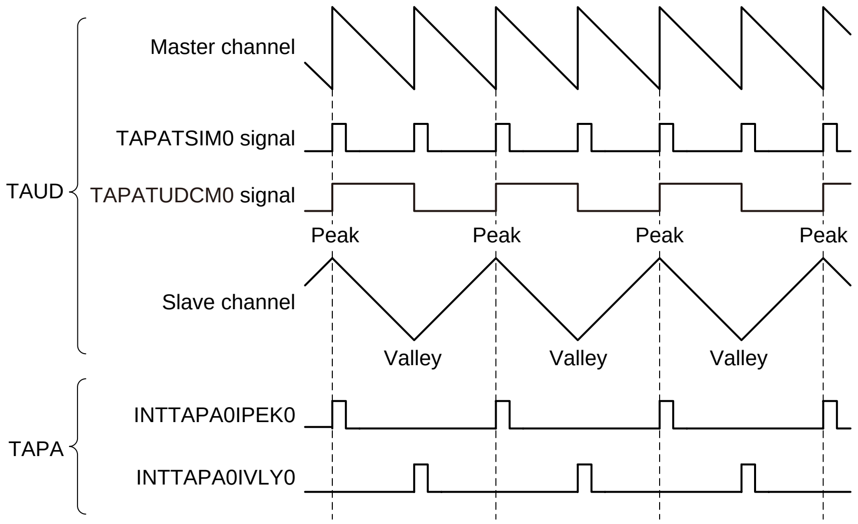


Figure 36.6 Basic Timing of Signals for the INT Signal Output Selection Function

Triangular carrier cycles are generated on the master channel.

The INT signal generated on the master channel in each half triangular carrier cycle is input to TAPAn as the TAPATSIM0 signal. TAPAn generates the INTTAPA0IPEK0 signal (peak interrupt) during high level of the TAPATUDCM0 signal and the INTTAPA0IVLY0 signal (valley interrupt) during low level of the TAPATUDCM0 signal by using the TAPATSIM0 and TAPATUDCM0 input signals.

CAUTION

The peak interrupt INTTAPA0IPEK0 and valley interrupt INTTAPA0IVLY0 are generated regardless of the function of the master channel of TAUD.

When not using these peak and valley interrupts, mask them by using the ICTAPAnIPEK0 and ICTAPAnIVLY0 registers, respectively.

36.5.2 Block Diagram

TAUDn and TAPAn are connected in the registers shown below by the INT signal output selection function.

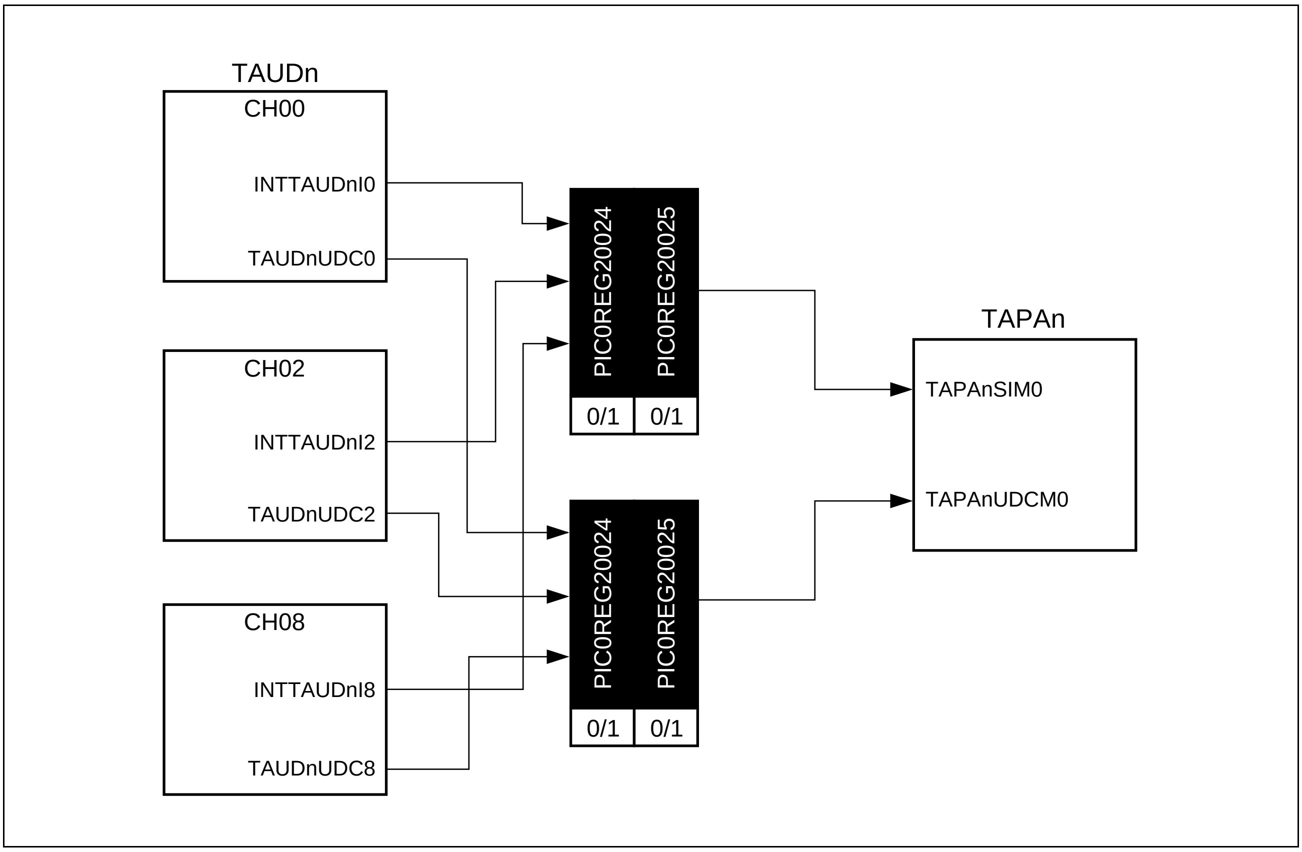


Figure 36.7 Connection of the INT Signals

36.5.3 Registers

36.5.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects the TAPA0 input.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG200: FFDD 00C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0REG2n025	PIC0REG2n024	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.23 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*[1]
25, 24	PIC0REG2n025 PIC0REG2n024	Select the TAUDn channel used by TAPATSIM0 and TAPATUDCM0. 00: Not selected 01: TAUD0 channel 0 selected 10: TAUD0 channel 2 selected 11: TAUD0 channel 8 selected
23 to 0	Reserved	*[1]

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

36.6 A/D Converter Conversion Trigger Selection Function

This function outputs the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1 from the TAPATCDENS0 and TAPATCDENS1 signals, which are connected to a compare match interrupt based on the triangular carrier cycle of TAUD, or a valley interrupt signal (INTTAPA0IVLY0).

36.6.1 Configuration of A/D Converter Conversion Trigger Selection Function

Table 36.24 Signals Used for TAPATADOUT Generation

Output Signal	Slave Match Detection Signal	Valley Interrupt Signal
TAPATADOUT0	TAPATCDENS0	INTTAPA0IVLY0
TAPATADOUT1	TAPATCDENS1	INTTAPA0IVLY0

Table 36.25 Operation of TAPATADOUT1 According to the Setting of TAPAnCTL1.TAPAnATS[3:2]

TAPAnATS3	TAPAnATS2	Description
0	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 from TAPATADOUT1.

Table 36.26 Operation of TAPATADOUT0 According to the Setting of TAPAnCTL1.TAPAnATS[1:0]

TAPAnATS1	TAPAnATS0	Description
0	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 from TAPATADOUT0.

36.6.2 Block Diagram

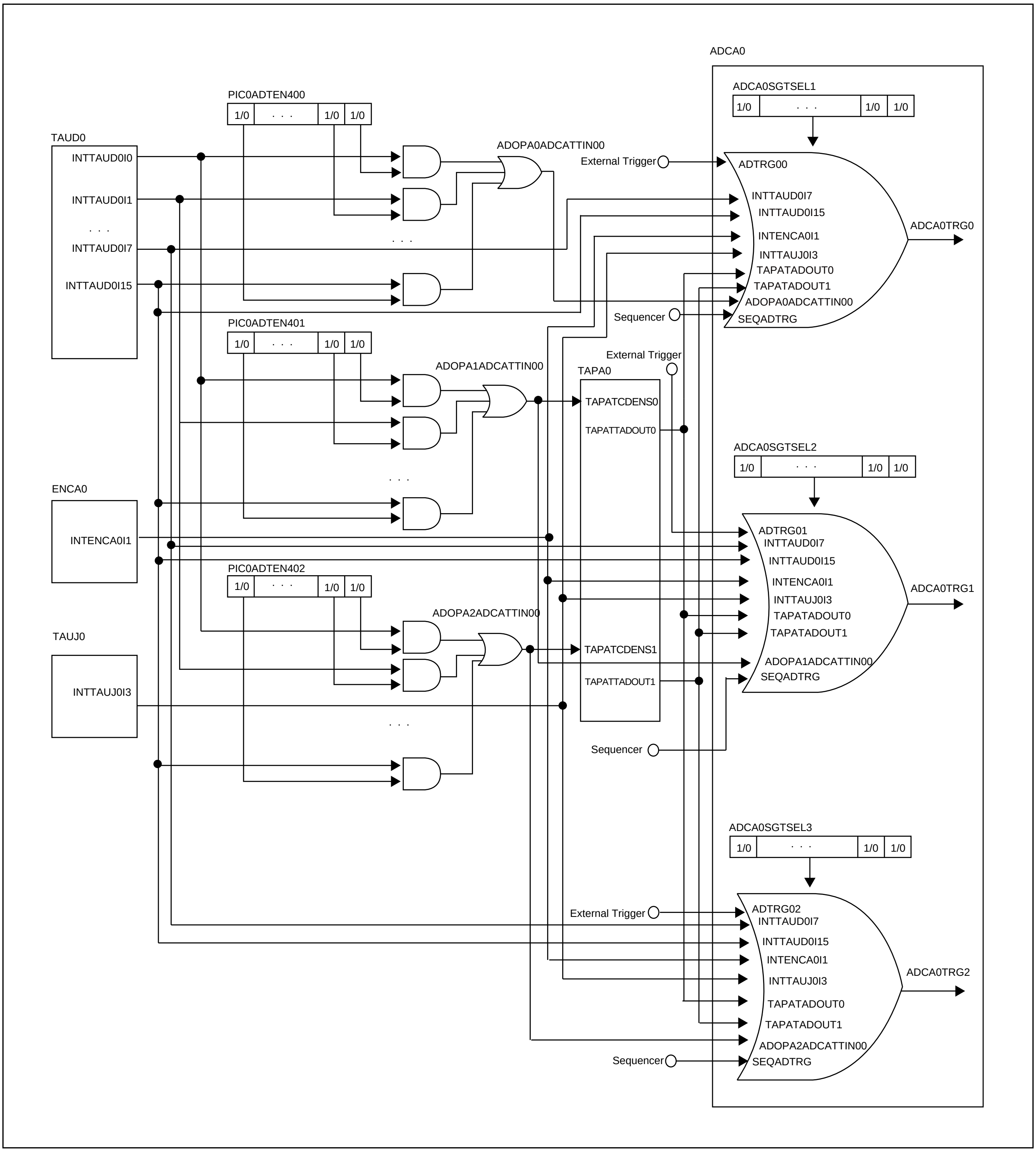


Figure 36.8 Block Diagram of A/D Conversion Trigger Selection Function

NOTE

See **Section 38.3.4.1, ADCAnSGTSELx — Scan Group x Start Trigger Control Register** for details on the settings of the ADCA0SGTSEL register.

36.6.3 Waveforms of A/D Converter Conversion Trigger Output Control Operation in Triangle PWM Mode

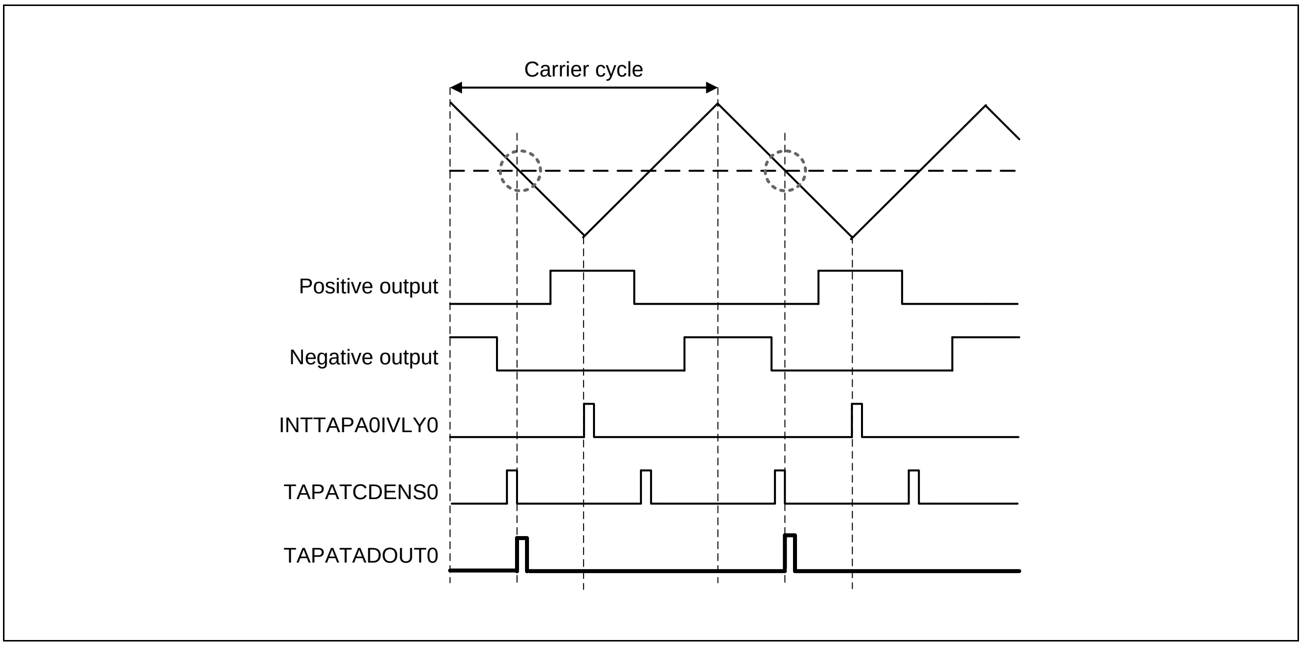


Figure 36.9 TAPAnATS[1:0] bits = 00_B: Output INT Signal while the Triangle Wave is Falling (Counting Down)

While the triangle wave is falling (counting down), the signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

In this case, no A/D converter conversion trigger signal is output while the triangle wave is rising (counting up).

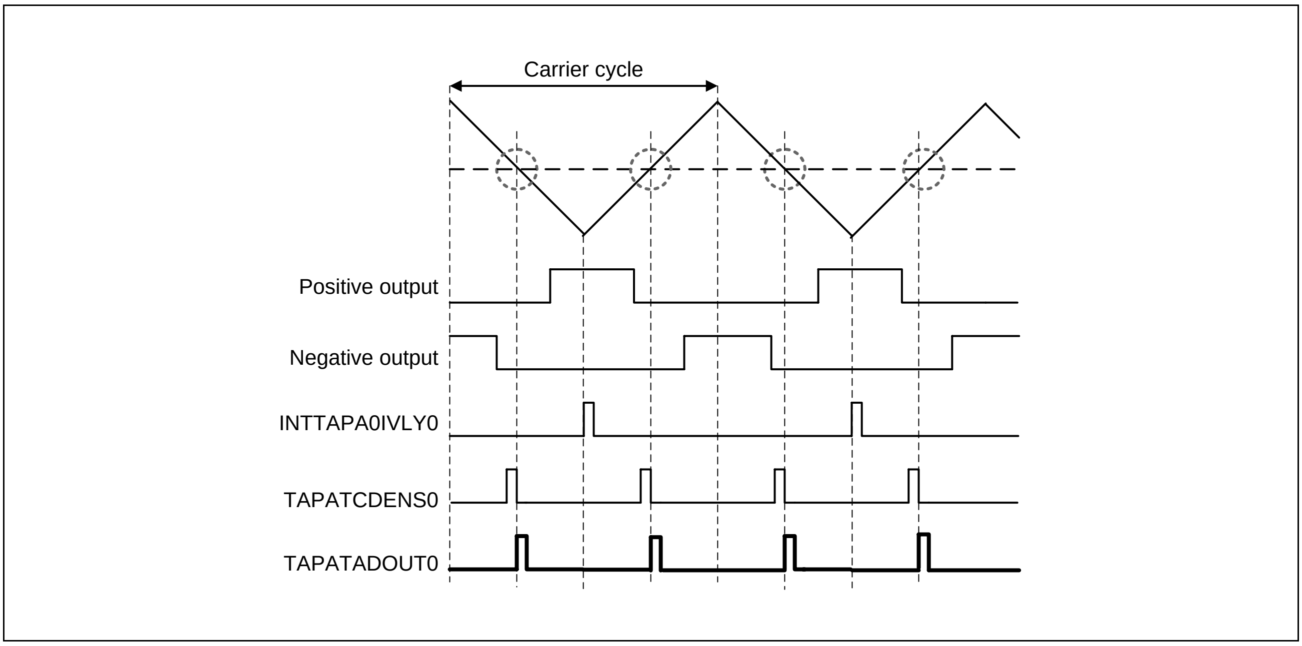


Figure 36.10 TAPAnATS[1:0] bits = 10_B: Output INT Signal while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

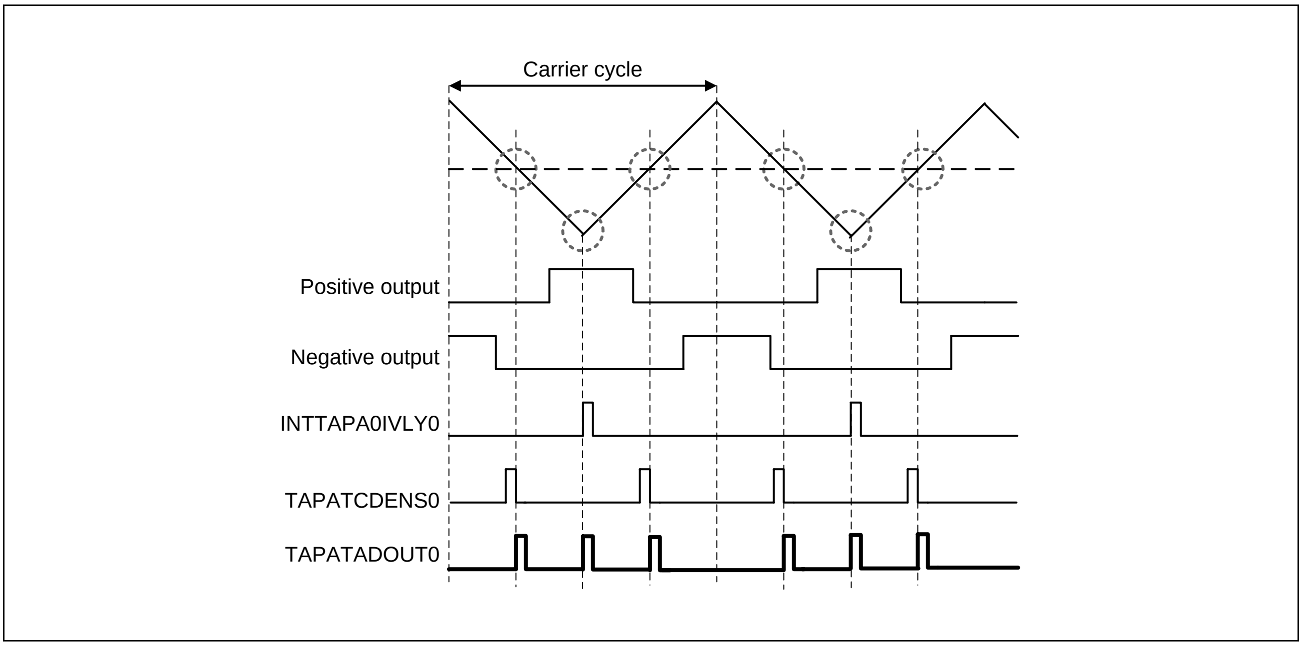


Figure 36.11 TAPAnATS[1:0] bits = 11_B: Output of INT Signal and Valley Interrupt while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 and valley interrupt INTTAPA0IVLY0 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

36.6.4 Operating Procedure for A/D Converter Conversion Trigger Selection Function

The operating procedure for the A/D converter conversion trigger selection function is shown below.

	Operation	Status of TAUD and TAPA
Initial setup	Initialize TAUD. Specify the timer operation mode. Set up the TAPAnCTL1 register. Specify TAPAnATS[1:0] (TAPATADOUT0 setting). Specify TAPAnATS[3:2] (TAPATADOUT1 setting). Set up the PIC0ADTEN4nj and PIC0REG2n0 registers according to the signal to be used. Specify PIC0ADTEN4nj (TAPATCDENS0 or TAPATCDENS1 setting). Specify PIC0REG2n0 (INTTAPA0IVLY0 setting).	TAUD and TAPA stop the operation.
Start operation	Start the TAUD operation.	TAUD starts the count operation.
During operation	TAUD operates according to the setting of each function.	The A/D conversion trigger selection function outputs either TAPATADOUT0 according to the setting of TAPAnATS[1:0] or TAPATADOUT1 according to the setting of TAPAnATS[3:2], based on the interrupt TAPATCDENS1 or TAPATCDENS0, which is input from TAUD, and the valley interrupt INTTAPA0IVLY0, which is generated by TAPA.
Stop operation	Stop the TAUD operation.	TAUD stops the count operation.

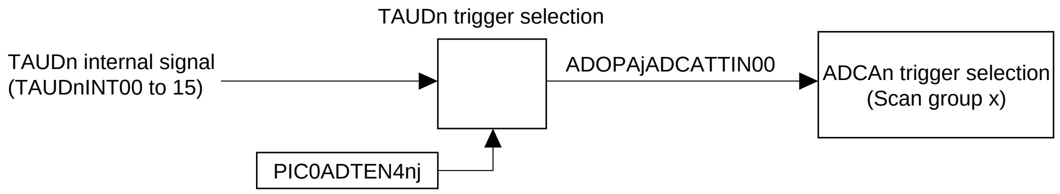
Restart

36.7 ADCA Trigger Selection Function

36.7.1 Functional Overview

This function generates ADCA hardware trigger signals by using TAUDn channel output.

36.7.2 Configuration



Note: For details, see **Figure 36.8, Block Diagram of A/D Conversion Trigger Selection Function**.

Figure 36.12 Block Diagram of ADCA Trigger Selection Function

36.7.3 Registers

36.7.3.1 PIC0ADTEN4nj — A/D Conversion Trigger Output Control Register 4nj (n = 0, j = 0 to 2)

This register selects an ADCA0 start trigger source from TAUDn channel m. (m = 0 to 15)

Access: This register can be read or written in 16-bit units.

Address: <PIC0_base> + 90_H + 4 × j

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0 ADTEN 4nj15	PIC0 ADTEN 4nj14	PIC0 ADTEN 4nj13	PIC0 ADTEN 4nj12	PIC0 ADTEN 4nj11	PIC0 ADTEN 4nj10	PIC0 ADTEN 4nj09	PIC0 ADTEN 4nj08	PIC0 ADTEN 4nj07	PIC0 ADTEN 4nj06	PIC0 ADTEN 4nj05	PIC0 ADTEN 4nj04	PIC0 ADTEN 4nj03	PIC0 ADTEN 4nj02	PIC0 ADTEN 4nj01	PIC0 ADTEN 4nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.27 PIC0ADTEN4nj Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0ADTEN4nj15 to PIC0ADTEN4nj00	Sets a trigger source of CHm (m = 0 to 15) in the TAUDn timer. 0: A/D trigger source of CHm in the TAUDn timer is disabled. 1: A/D trigger source of CHm in the TAUDn timer is enabled.

36.7.4 Example of Operation

(1) Initial setting: Set the function of each channel of the TAUD0 to be used.

(2) Setting of the A/D conversion trigger output control register 4nj (PIC0ADTEN4nj):
Setting the bits of A/D conversion trigger output control register 4nj ((PIC0ADTEN4nj) to 1 enables selection of an interrupt request signal from each channel of the TAUD0 as the trigger of the A/D conversion scan group.

– Register setting should be performed when A/D conversion is stopped.

(3) Setting of the A/D conversion trigger selection control register (ADCA0SGTSELx):
Setting the bits corresponding to each trigger to 1 enables to use the signal generated by executing the logical OR of each trigger as the start trigger of the A/D conversion scan group.

– Register setting should be performed when the A/D conversion is stopped.

(4) Enabling of TAUD0 timer operation
Each channel of the TAUD0 timer set in (1) starts.

36.7.5 Setup Flow

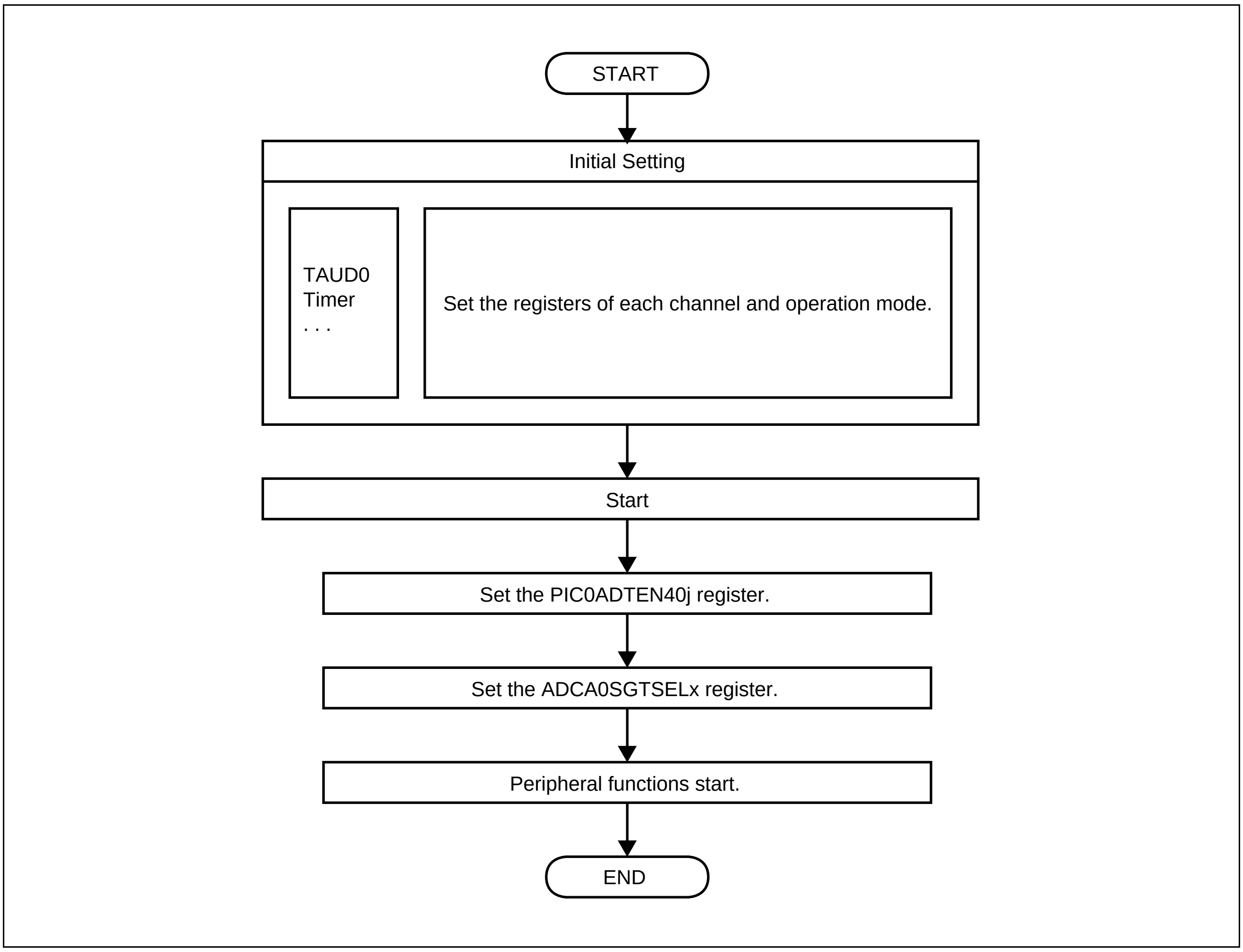


Figure 36.13 Setup Flow (j = 0 to 2)

36.8 Simultaneous Start Trigger Function

36.8.1 Functional Overview

The timers (TAUD0, TAUJ1, ENCA0) can be simultaneously started in any combination.

36.8.2 Configuration

(1) Configuration

Table 36.28 Configuration of Simultaneous Start Trigger Function

Configuration/Timer Function	Timer
Configuration of Timer	TAUD0, TAUJ1, ENCA0

(2) Block Diagram

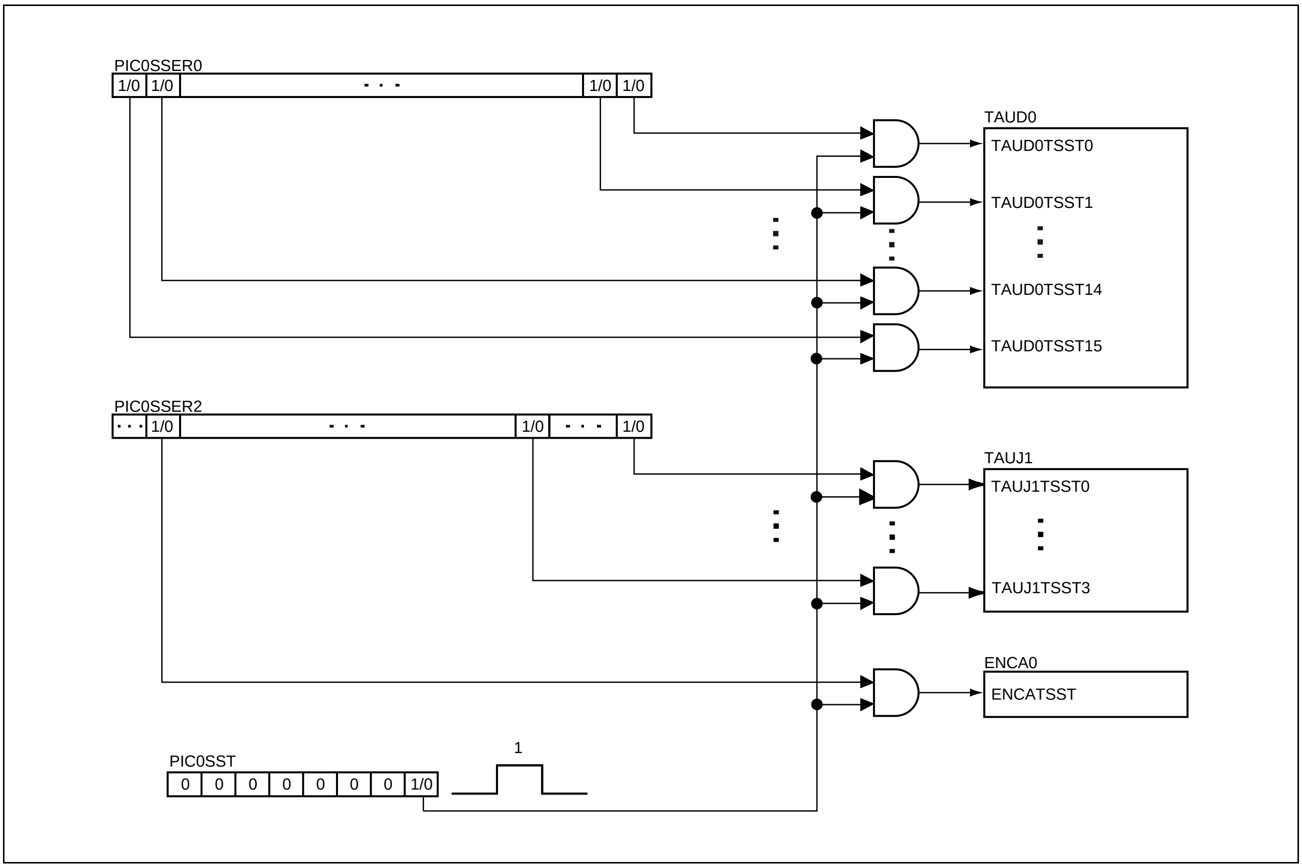


Figure 36.14 Block Diagram of Simultaneous Start Trigger

36.8.3 Registers

36.8.3.1 PIC0SSER0 — Simultaneous Start Control Register 0

The PIC0SSER0 register enables a start trigger for each channel of the TAUD0.

Access: This register can be read or written in 16-bit units

Address: <PIC0_base> + 10_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0SSER015	PIC0SSER014	PIC0SSER013	PIC0SSER012	PIC0SSER011	PIC0SSER010	PIC0SSER009	PIC0SSER008	PIC0SSER007	PIC0SSER006	PIC0SSER005	PIC0SSER004	PIC0SSER003	PIC0SSER002	PIC0SSER001	PIC0SSER000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.29 PIC0SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0SSER015 to PIC0SSER000	Enable a simultaneous start trigger for the CHm in the TAUD0. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

36.8.3.2 PIC0SSER2 — Simultaneous Start Control Register 2

The PIC0SSER2 register enables a start trigger for ENCA0 and TAUJ1.

Access: This register can be read or written in 16-bit units

Address: <PIC0_base> + 18_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0SSER214(ENCA0)	—	—	—	—	—	—	—	—	—	—	PIC0SSER203	PIC0SSER202	PIC0SSER201	PIC0SSER200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 36.30 PIC0SSER2 Register Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	PIC0SSER214	Enables a simultaneous start trigger for the ENCA0 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.
13 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	PIC0SSER203 to PIC0SSER200	Set a simultaneous start trigger for the CHm in the TAUJ1 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

36.8.3.3 PIC0SST — Simultaneous Start Trigger Control Register

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PIC0_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SYNCTRG
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.31 PIC0SST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SYNCTRG	Generates a start trigger for the timer whose simultaneous start is enabled. When read, this bit is always read as 0. 0: Disabled. 1: Simultaneous start trigger is generated (the pulse in the width of 1PCLK is output).

36.8.4 Example of Operation

(1) Operation example of timer configuration:
The timers that operates in operation mode to be selected can be simultaneously started in any combinations.

(2) Simultaneous start enable:
Setting the relevant bits in the PIC0SSER0 and PIC0SSER2 registers of the target timers to be simultaneously started to 1 enables these timers to simultaneously start.

(3) Start trigger output:
Writing 1 to the SYNCTRG bit in the PIC0SST register enables the target timers set in (2) to simultaneously start.

(4) Repeating (2) and (3) for the channels that have not started yet enables the different target timers to simultaneously start in multiple batches.

36.8.5 Setup Flow

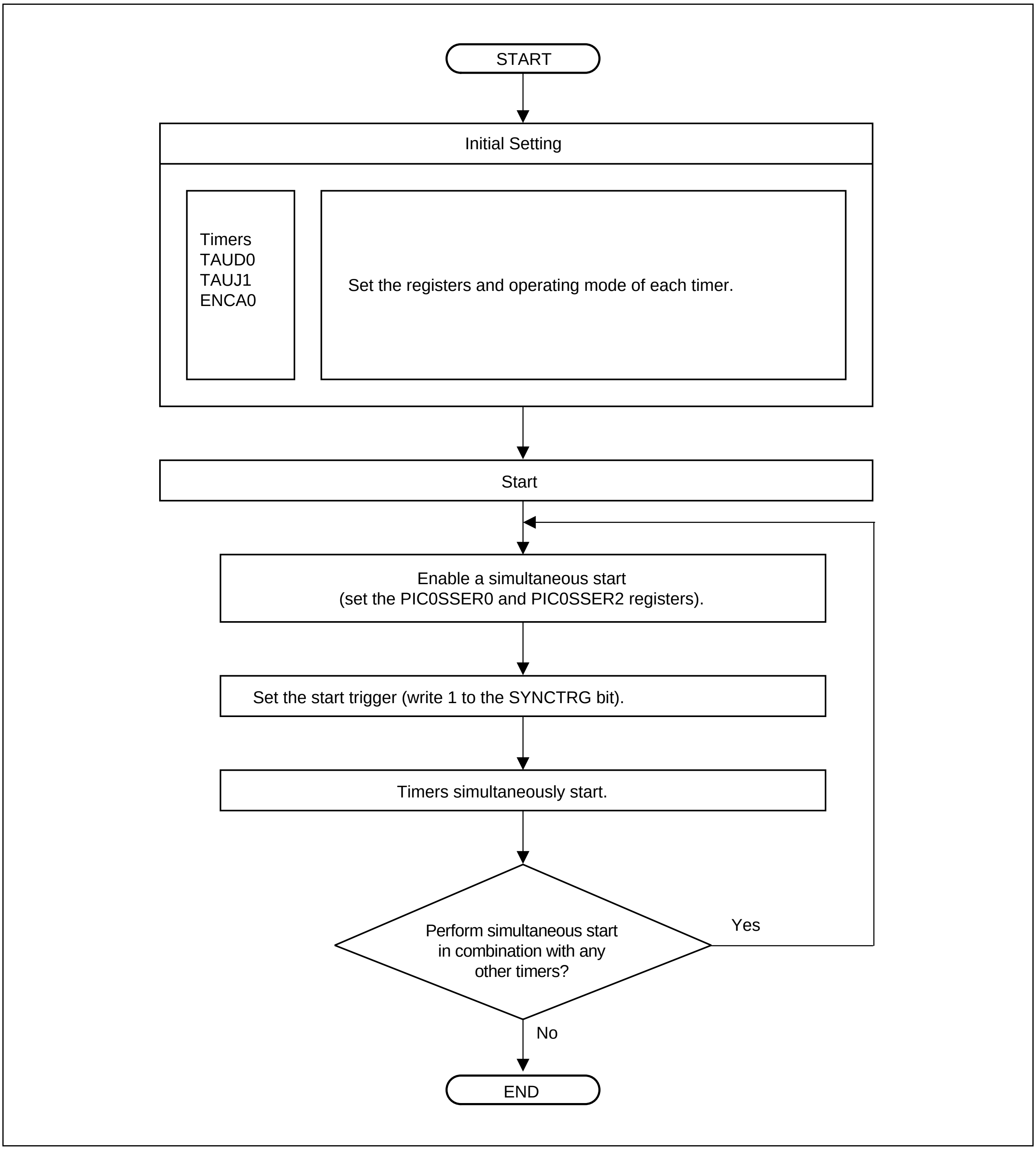


Figure 36.15 Setup Flow

36.9 Trigger & Pulse Width Measuring Function

36.9.1 Functional Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCA0 to TAUJ1 and TAUD0.

36.9.2 Configuration

(1) Configuration

Table 36.32 Configuration of Trigger & Pulse Width Measuring Function

Configuration/Timer Function	Timer
Configuration of Timer	ENCA0, TAUD0, TAUJ1

Table 36.33 Setting Functions of TAUJ1/TAUD0 Channels

TAU	Channels	Functions Name	M/S*[1]	Target Trigger of Pulse Width Measurement
TAUJ1	00	TINm input pulse interval measurement function	S	ENCAT0IEC*[2]
	01	TINm input pulse interval measurement function	S	ENCAT0IEC*[2]
TAUD0	00	TINm input pulse interval measurement function	S	ENCAT0EQ0, ENCAT0EQ1
	01	TINm input pulse interval measurement function	S	ENCAT0EQ1
	02	TINm input pulse interval measurement function	S	ENCAT0EQ0

Note 1. M: Master channel, S: Slave channel

Note 2. Read ENCAT0IEC as ENCATIEC (encoder clear interrupt) in **Table 35.7, Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)**.

(2) Block Diagram

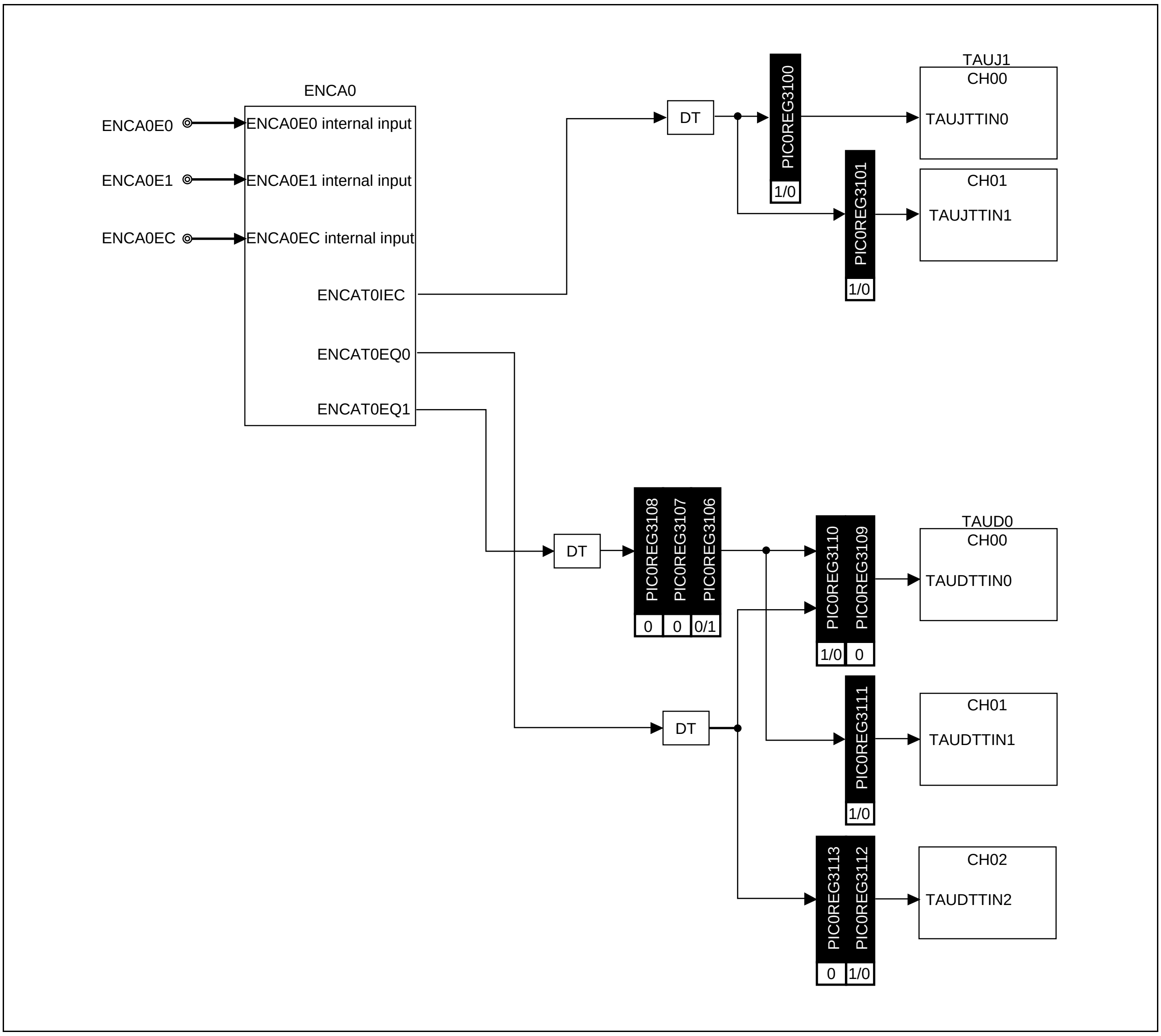


Figure 36.16 Block Diagram of Trigger & Pulse Width Measuring Function

36.9.3 Registers

36.9.3.1 PIC0REG31 — Timer I/O Control Register 31

Access: This register can be read or written in 32-bit units.

Address: <PIC0_base> + EC$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PIC0REG3113	PIC0REG3112	PIC0REG3111	PIC0REG3110	PIC0REG3109	PIC0REG3108	PIC0REG3107	PIC0REG3106	—	—	—	—	PIC0REG3101	PIC0REG3100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 36.34 PIC0REG31 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 14</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>13, 12</td><td>PIC0REG3113 to PIC0REG3112</td><td>Select a TIN input signal to CH2 of TAUD0.
<table>
<tr><th>PIC0REG 3113</th><th>PIC0REG 3112</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>CH2 of TAUD0 is not used for trigger width measurement.</td></tr>
<tr><td>0</td><td>1</td><td>DT output signal of ENCAT0EQ0</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>11</td><td>PIC0REG3111</td><td>Select a TIN input signal to CH1 of TAUD0.
0: CH1 of TAUD0 is not used for trigger width measurement.
1: Signal selected in PIC0REG3106 to PIC0REG3108 (when measuring the ENCATEQ1 signal)</td></tr>
<tr><td>10, 9</td><td>PIC0REG3110 to PIC0REG3109</td><td>Select a TIN input signal to CH0 of TAUD0.
<table>
<tr><th>PIC0REG 3110</th><th>PIC0REG 3109</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>Signal selected in PIC0REG3106 to PIC0REG3108</td></tr>
<tr><td>0</td><td>1</td><td>DT output signal of ENCAT0EQ0</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>8 to 6</td><td>PIC0REG3108 to PIC0REG3106</td><td>Select a TIN input signal to CH0 and CH1 of TAUD0.
<table>
<tr><th>PIC0REG 3108</th><th>PIC0REG 3107</th><th>PIC0REG 3106</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>CH0 of TAUD0 is not used for trigger width measurement.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>DT output signal of ENCAT0EQ1</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>5 to 2</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
</table>

Table 36.34 PIC0REG31 Register Contents

Bit Position	Bit Name	Function
1	PIC0REG3101	Selects a TIN input signal to CH1 of TAUJ1. 0: CH1 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC
0	PIC0REG3100	Selects a TIN input signal to CH0 of TAUJ1. 0: CH0 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC

36.9.4 Example of Operation

The trigger and pulse width measurement function is achieved by combining the ENCA0 trigger signals (ENCAT0IEC, ENCAT0EQ0, ENCAT0EQ1) and the following functions of TAUD0 and TAUJ1.

- TAUDTTINm input pulse interval measurement function (TAUD0)
- TAUJTTINm input pulse interval measurement function (TAUJ1)

Also, the following function of PIC is used to convert the trigger signal input to TINm into a level-sensitive toggle signal.

- DT circuit

The trigger and pulse width measurement function implements measurement of the ENCA0 output trigger signal interval using the TAUDTTINm input pulse interval measurement function of TAUD0 and the TAUJTTINm input pulse interval measurement function of TAUJ1.

(1) TAUDTTINm Input Pulse Interval Measurement Function, TAUJTTINm Input Pulse Interval Measurement Function

When the valid TINm edge of TAUD0 or TAUJ1 is detected, the CNTm value is captured into CDRm and CNTm is cleared.

CAUTION

Set both edges (rising and falling edges) of TINm to be detected as valid (TAUD0CMURm.TAUD0TIS[1:0] = 10_B, TAUJ1CMURm.TAUJ1TIS[1:0] = 10_B) for this function.

For details of the TAUD and TAUJ functions, see the corresponding sections.

(2) DT Circuit

The DT circuit is used to convert the trigger signal output from ENCA0 into a level-sensitive toggle signal.

As shown in **Figure 36.17, Operation of DT Circuit**, the output signal is toggled upon each input trigger signal generation.

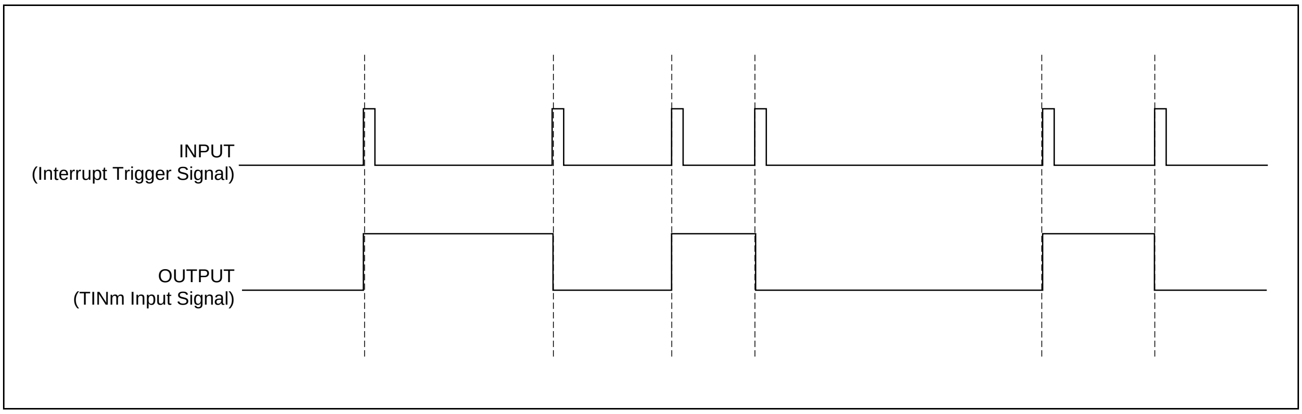


Figure 36.17 Operation of DT Circuit

PIC provides input signal conversion and signal connection to TAUD0 and TAUJ1 to measure the generation interval of trigger signals from ENCA0.

The timing chart of the trigger and pulse width measurement function is shown below.

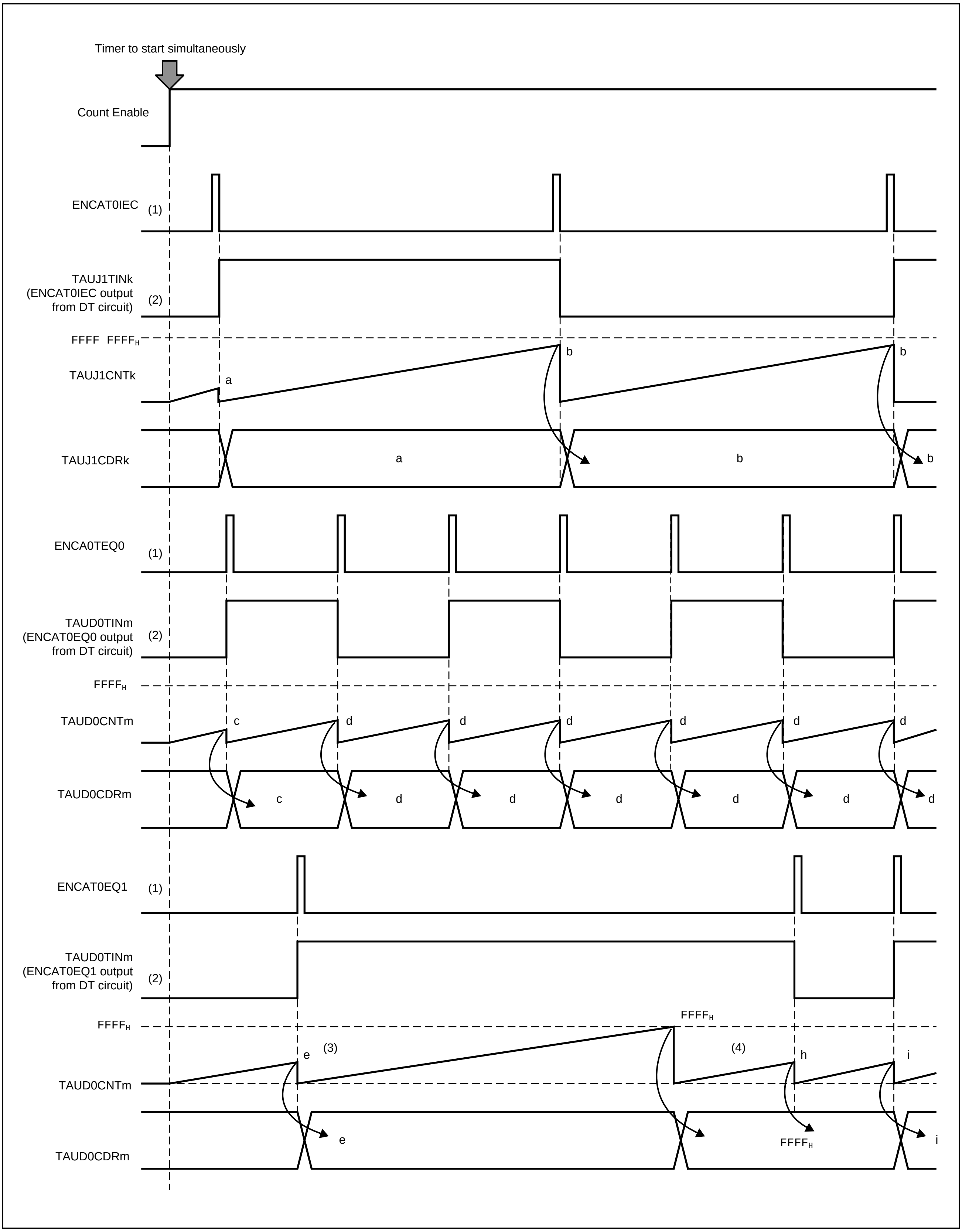


Figure 36.18 Operation Example of the Function of Trigger and Pulse Width Measurement (m = 0 to 2, k = 0, 1)

(1) The following signals are output from ENCA0 as triggers:

- ENCAT0IEC (interrupt trigger signal output when timer counter value is cleared by ENCA0EC input)
- ENCAT0EQ0 (trigger signal output according to timing of a match of timer counter value and value of compare register 0)
- ENCAT0EQ1 (trigger signal output according to timing of a match of timer counter value and value of compare register 1)

(2) The trigger signal output from ENCA0 is converted to a level-sensitive toggle signal by the DT circuit and is output to TINm of TAUD0 and TAUJ1.

(3) By setting both TINm edges (rising and falling edges) of TAUD0 and TAUJ1 as valid, the CNTm value is captured into CDRm at the TINm toggle timing and cleared to 0000_H. This operation is repeated.
The first captured value (shown as “a” in the figure) from the start of operation indicates the interval from the start of TAUJ operation until trigger input.

(4) When an overflow occurs, the count value $FFFF_H$ (FFFF FFFF_H for TAUJ) is captured but the count value is not captured at the first trigger after the overflow.

With the above operation, the trigger generation interval can be measured.

The following table shows the combinations of the trigger signals and measurement timers, and the bit settings of the pertinent PIC registers for setting the signal paths and the I/O selection registers. Appropriately set these bits according to the trigger signal to be measured and the measurement timer to be used.

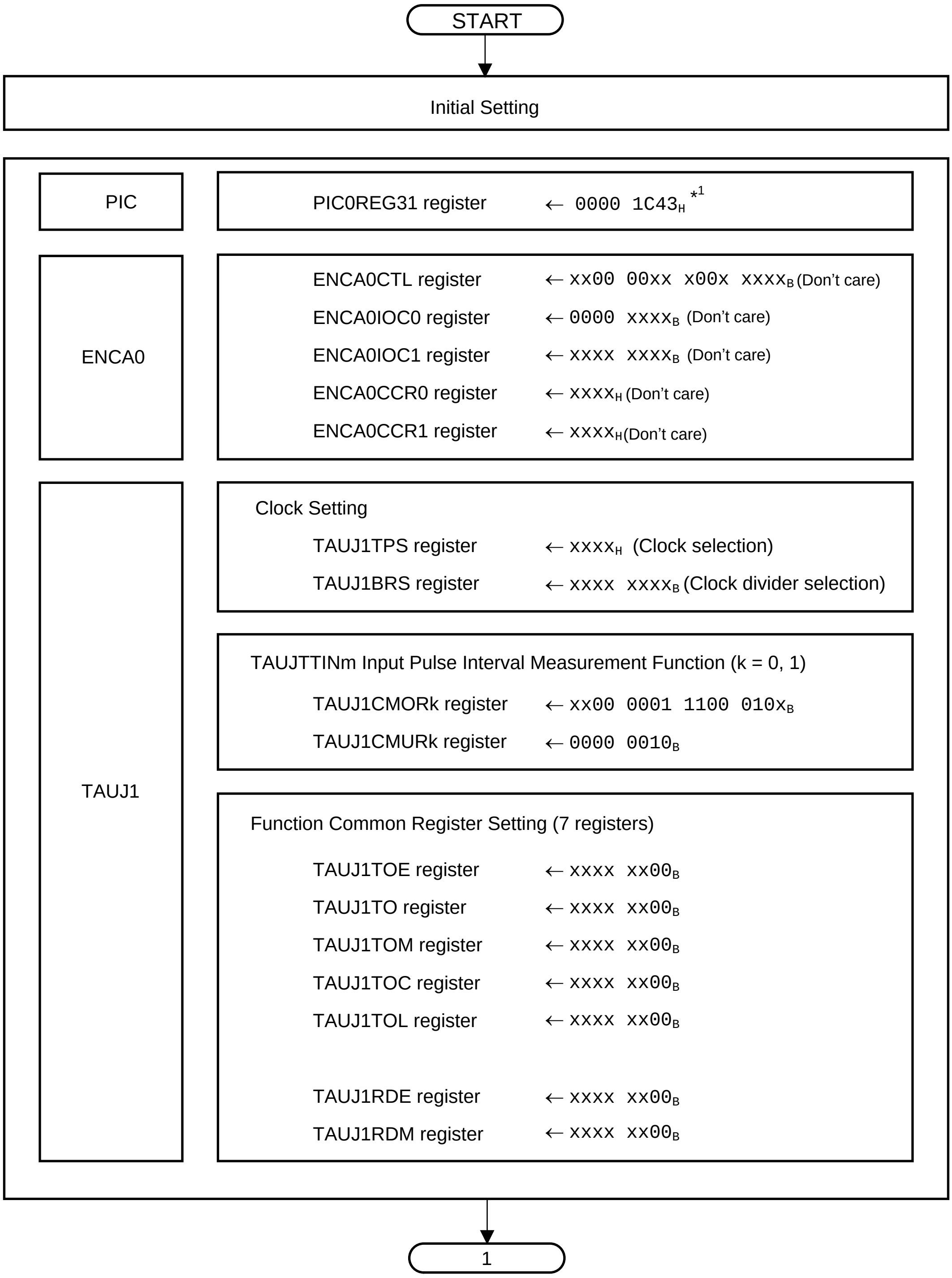
Table 36.35 Combinations of Trigger Signals and Measurement Timers

Interrupt Trigger Signal	Measurement Timer	PIC Register Bit Setting	
ENCAT0IEC	TAUJ1 CH0	PIC0REG3100 = 1	
	TAUJ1 CH1	PIC0REG3101 = 1	
ENCAT0EQ0	TAUD0 CH0	PIC0REG3109 = 0 PIC0REG3110 = 1	
	TAUD0 CH2	PIC0REG3112 = 1 PIC0REG3113 = 0	
ENCAT0EQ1	TAUD0 CH0	PIC0REG3106 = 1 PIC0REG3107 = 0 PIC0REG3108 = 0	PIC0REG3109 = 0 PIC0REG3110 = 0
	TAUD0 CH1		PIC0REG3111 = 1

36.9.5 Setup Flow

The setup flow in this section shows the general setup flow to measure the pulse interval, which applies to all the following combinations. For the combinations of the trigger signals and measurement timers, see **Table 36.35, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1



PIC0REG31 register ← 0000 1C43$_{\text{H}}$ *1

ENCA0CTL register ← xx00 00xx x00x xxxx$_{\text{B}}$ (Don't care)
ENCA0IOC0 register ← 0000 xxxx$_{\text{B}}$ (Don't care)
ENCA0IOC1 register ← xxxx xxxx$_{\text{B}}$ (Don't care)
ENCA0CCR0 register ← xxxx$_{\text{H}}$ (Don't care)
ENCA0CCR1 register ← xxxx$_{\text{H}}$(Don't care)

Clock Setting
TAUJ1TPS register ← xxxx$_{\text{H}}$ (Clock selection)
TAUJ1BRS register ← xxxx xxxx$_{\text{B}}$ (Clock divider selection)

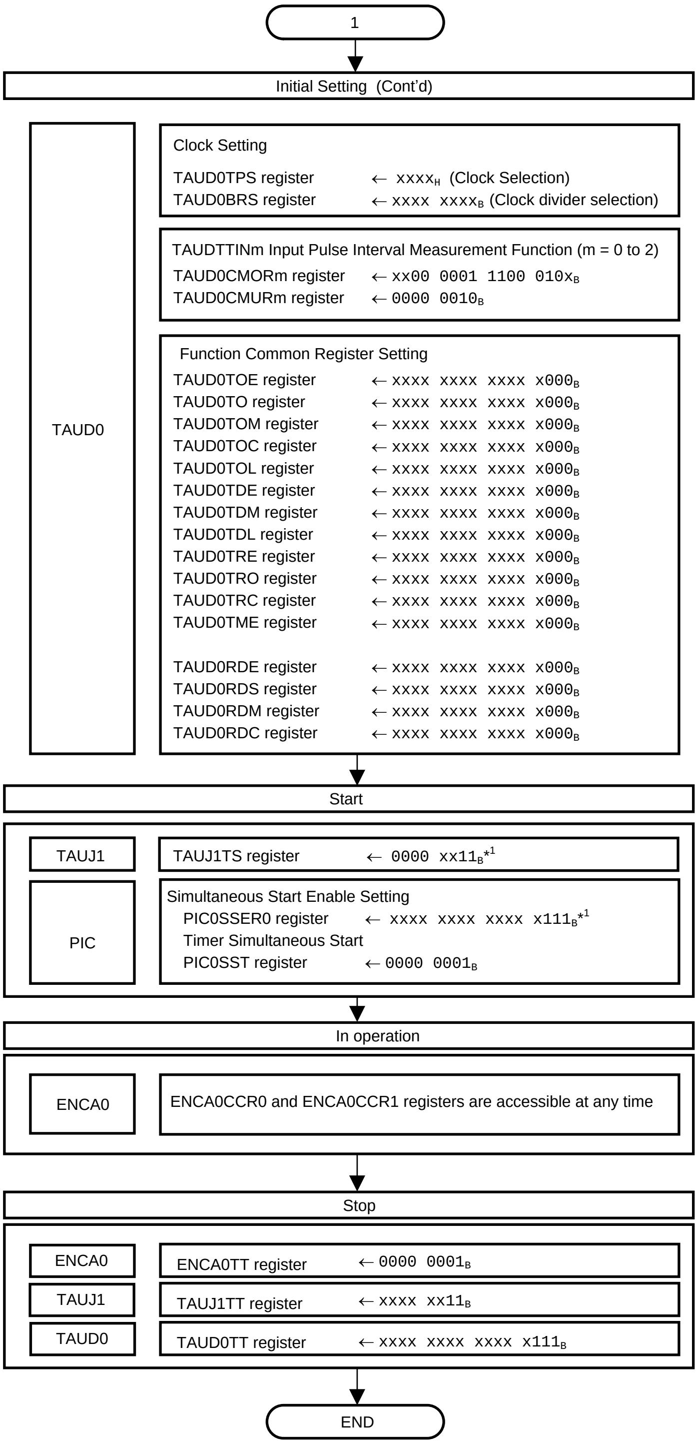
TAUJTTINm Input Pulse Interval Measurement Function (k = 0, 1)
TAUJ1CMORk register ← xx00 0001 1100 010x$_{\text{B}}$
TAUJ1CMURk register ← 0000 0010$_{\text{B}}$

Function Common Register Setting (7 registers)
TAUJ1TOE register ← xxxx xx00$_{\text{B}}$
TAUJ1TO register ← xxxx xx00$_{\text{B}}$
TAUJ1TOM register ← xxxx xx00$_{\text{B}}$
TAUJ1TOC register ← xxxx xx00$_{\text{B}}$
TAUJ1TOL register ← xxxx xx00$_{\text{B}}$

TAUJ1RDE register ← xxxx xx00$_{\text{B}}$
TAUJ1RDM register ← xxxx xx00$_{\text{B}}$

Note 1. Change the settings depending on the combination of the trigger signal to be measured and measurement timer to be used.

Figure 36.19 Setup Flow



Note 1. Change the settings depending on the combination of the trigger signal to be measured and measurement timer to be used.

Figure 36.20 Setup Flow (Cont'd)

36.9.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up measurement of the pulse interval for all the combinations below. For the combinations of the trigger signals and measurement timers, see **Table 36.35, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1

Table 36.36 ENCA0 Setting

Register	Bit Position	Bit Name	Setting Value	Note
ENCA0CTL	15	ENCA0CME	Don't care	Enables or disables compare match interrupt detection mask
	14	ENCA0MCS	Don't care	Selects a cancelation trigger for compare match interrupt detection mask
	13 to 10		0	Fixed to 0
	9	ENCA0CRM1	Don't care	Selects the ENCA0CCR1 register function
	8	ENCA0CRM0	Don't care	Selects the ENCA0CCR0 register function
	7	ENCA0CTS	Don't care	Selects trigger for capture operation of ENCA0CCR1.
	6, 5		0	Fixed to 0
	4	ENCA0LDE	Don't care	Enables or disables reload operation when underflow is generated
	3	ENCA0ECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR1
	2	ENCA0ECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR0
	1, 0	ENCA0UDS[1:0]	Don't care	Selects the counter up/down control by ENCA0E0 and ENCA0E1
ENCA0IOC0	7 to 4		0	Fixed to 0
	3, 2	ENCA0TIS[3:2]	Don't care	Selects the valid edge for capture trigger 1 (ENCA0I1)
	1, 0	ENCA0TIS[1:0]	Don't care	Selects the valid edge for capture trigger 0 (ENCA0I0)
ENCA0IOC1	7	ENCA0SCE	Don't care	Enables encoder special-clear
	6	ENCA0ZCL	Don't care	Selects the clear level of Z phase for a encoder special-clear
	5	ENCA0BCL	Don't care	Selects the clear level of B phase for a encoder special-clear
	4	ENCA0ACL	Don't care	Selects the clear level of A phase for a encoder special-clear
	3, 2	ENCA0ECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCA0EIS[1:0]	Don't care	Selects encoder input (A or B phase) edge

Table 36.37 TAUJ1 Setting (k = 0, 1)
TAUJ1 (TAUJTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUJ1CMORk	15,14	TAUJ1CKS[1:0]	Don’t care	Operation Clock Setting
	13,12	TAUJ1CCS[1:0]	00	
	11	TAUJ1MAS	0	
	10, 9, 8	TAUJ1STS[2:0]	001	
	7, 6	TAUJ1COS[1:0]	11	
	5		0	Fixed to 0
	4, 3, 2, 1	TAUJ1MD[4:1]	0010	
	0	TAUJ1MD0	Don’t care	
TAUJ1CMURk	1, 0	TAUJ1TIS[1:0]	10	

NOTE

When TAUJ1CMORk is used for the TAUDTTINm input pulse interval measurement function, the TAUJ1CKS[1:0] (operating clock selection) and TAUJ1MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUJ1COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 33, Timer Array Unit J (TAUJ)**.

For TAUJ common registers (TAUJ1TOE, TAUJ1TO, TAUJ1TOM, TAUJ1TOC, TAUJ1TOL, TAUJ1RDE, and TAUJ1RDM), only set the bits corresponding to the used channels to 0.

Table 36.38 TAUD0 Setting (m = 0 to 2)
TAUD0 (TAUDTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUD0CMORm	15, 14	TAUD0CKS[1:0]	Don’t care	Operation Clock Setting
	13, 12	TAUD0CCS[1:0]	00	
	11	TAUD0MAS	0	
	10 to 8	TAUD0STS[2:0]	001	
	7, 6	TAUD0COS[1:0]	11	
	5		0	Fixed to 0
	4 to 1	TAUD0MD[4:1]	0010	
	0	TAUD0MD0	Don’t care	
TAUD0CMURm	1, 0	TAUD0TIS[1:0]	10	

NOTE

When TAUD0CMORm is used for the TAUDTTINm input pulse interval measurement function, the TAUD0CKS[1:0] (operating clock selection) and TAUD0MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUD0COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 32, Timer Array Unit D (TAUD)**.

For TAUD common registers (TAUD0TOE, TAUD0TO, TAUD0TOM, TAUD0TOC, TAUD0TOL, TAUD0TDE, TAUD0TDM, TAUD0TDL, TAUD0TRE, TAUD0TRO, TAUD0TRC, TAUD0TME, TAUD0RDE, TAUD0RDS, TAUD0RDM, and TAUD0RDC), only set the bits corresponding to the used channels to 0.

Table 36.39 PIC Setting

Register	Bit Position	Bit Name	Setting Value	Note
PIC0REG31	13, 12	PIC0REG3113 PIC0REG3112	0 1	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN2 input signal
	11	PIC0REG3111	1	Selects the signal selected with PIC0REG3106 to PIC0REG3108 (DT output signal from ENCAT0EQ1) as TAUD0TTIN1 input signal
	10, 9	PIC0REG3110 PIC0REG3109	1 0	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN0 input signal
	8 to 6	PIC0REG3108 PIC0REG3107 PIC0REG3106	0 0 1	Selects the DT output signal from ENCAT0EQ1 as TAUD0TTIN1 or TAUD0TTIN0 input signal
	1	PIC0REG3101	1	Selects the DT output signal from ENCAT0IEC as TAUJ1TTIN1 input signal
	0	PIC0REG3100	1	Selects the DT output signal from ENCAT0IEC as TAUJ1TTIN0 input signal

36.10 A/D Trigger Encoder Capture Function

36.10.1 Functional Overview

The value of the encoder counter synchronized with A/D conversion can be obtained by using an A/D conversion trigger signal as a capture signal of ENCA0.

36.10.2 Configuration

(1) Configuration

Table 36.40 Configuration of A/D Trigger Encoder Capture Function

A/D Converter	Encoder Timer
ADCA0	ENCA0

(2) Block Diagram

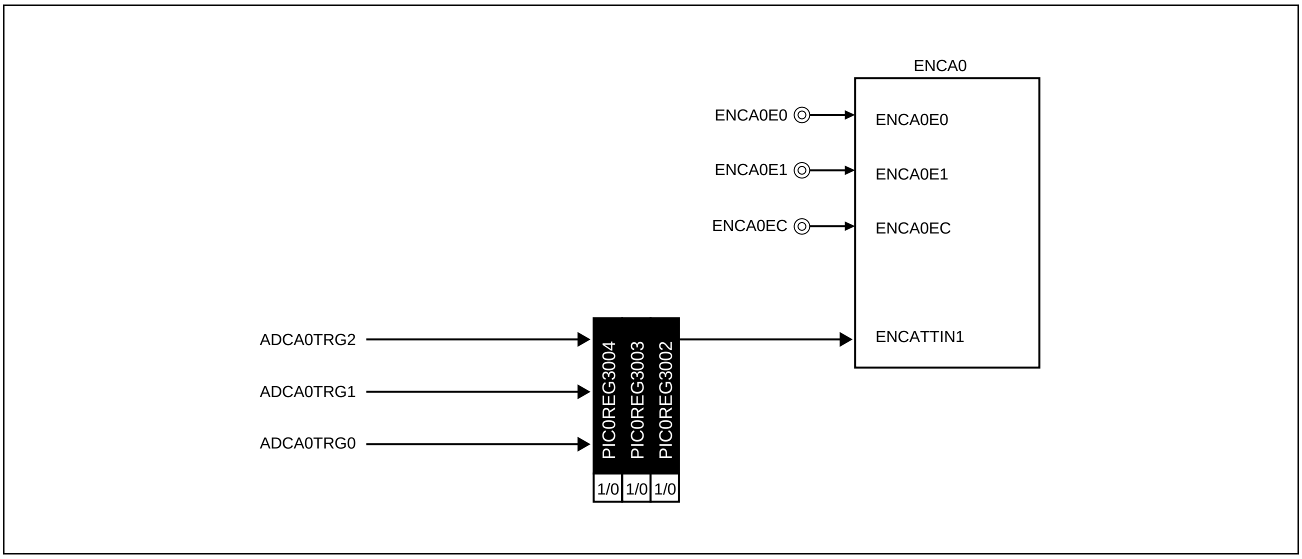


Figure 36.21 Block Diagram of A/D Trigger Encoder Capture Function

CAUTION

1. It takes ENCA0 one additional clock cycle to capture the input signal of the ADCA0TRG0, ADCA0TRG1, and ADCA0TRG2 pins compared with the number of clock cycles it takes to capture the input signal of the ENCA0TIN1 pin when using CKSCLK_IPERI1, and three additional clock cycles when using CKSCLK_AADCA. Be sure to take this into account when configuring your system.
2. Configure the edge detection function by using the edge detection function registers of the digital noise filter, which are FCLA0CTL0_ADC0,
FCLA0CTL1_ADC0, and FCLA0CTL2_ADC0 (see **Section 2A.12.1.4, Input Pins that Incorporate Digital Filter Type D, Section 2B.12.1.4, Input Pins that Incorporate Digital Filter Type D** and **Section 2C.12.1.4, Input Pins that Incorporate Digital Filter Type D** for details), and specify “rising edge” for edge detection of the ENCA0TIN1 capture trigger input of ENCA0 (ENCA0IOC0.ENCA0TIS [3:2] = 01_B). Do not set ENCA0IOC0.ENCA0TIS [3:2] to 10_B (falling edge) or 11_B (both edges).
3. ADCA0TRG0-2 are selected by ADCA0SGTSEL1-3 signal shown in **Figure 36.8, Block Diagram of A/D Conversion Trigger Selection Function**.

36.10.3 Registers

36.10.3.1 PIC0REG30 — Timer I/O Control Register 30

Access: This register can be read or written in 32-bit units.

Address: <PIC0_base> + E8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG3004	PIC0REG3003	PIC0REG3002	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 36.41 PIC0REG30 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 5</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>4 to 2</td><td>PIC0REG3004 to PIC0REG3002</td><td>Selects an input signal to ENCATTIN1.
<table>
<tr><th>PIC0REG 3004</th><th>PIC0REG 3003</th><th>PIC0REG 3002</th><th>Input Signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Capture is not performed by an A/D trigger signal in ENCA0.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>ADCA0TRG2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ADCA0TRG1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ADCA0TRG0</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
</table>

36.10.4 Example of Operation

The A/D trigger encoder capture function is implemented by connecting A/D conversion trigger signal ADCAnTRGi (n = 0, i = 0 to 2) to ENCA0.

CAUTION

When using this function, the ENCA0 interrupt signal ENCATINT1 should not be selected as the A/D converter trigger. If selected, the correct operation cannot be performed because the following loop occurs: ADCAnTRG1 generation → ENCA0 capture operation → ENCATINT1 generation by capture operation → ADCAnTRG1 generation.

The following shows a timing chart of the A/D trigger encoder capture function using the ADCA0TRG1 as a trigger.

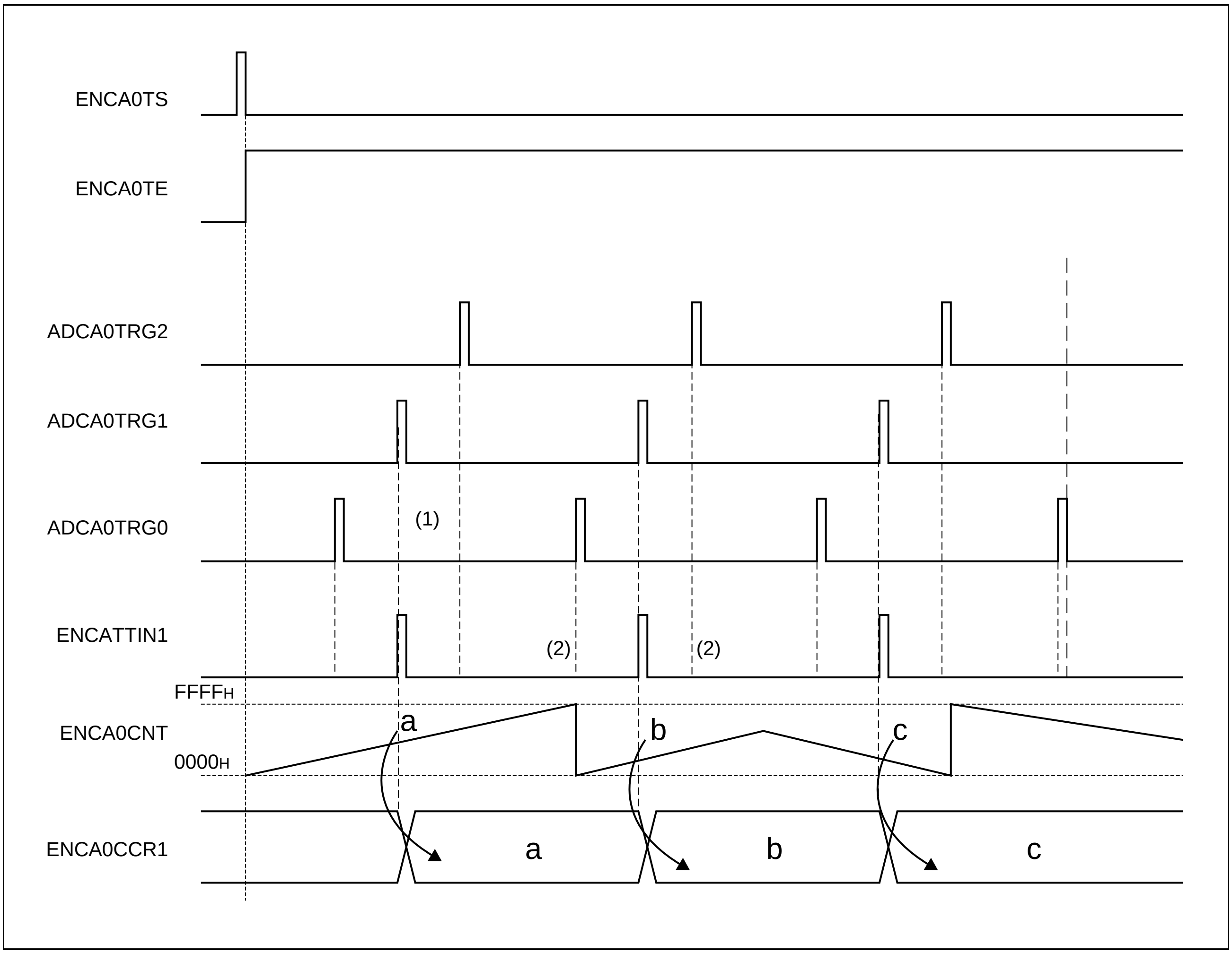


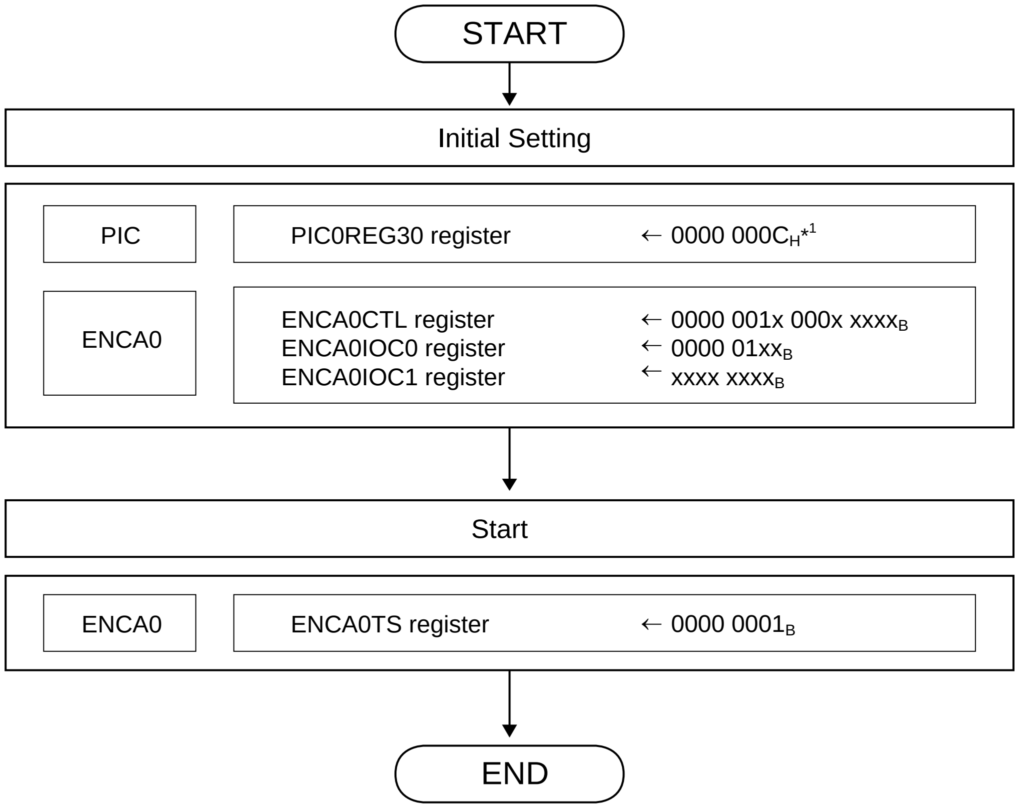
Figure 36.22 Operation Example of Trigger Encoder Capture Function

(1) When ADCA0TRG1 is selected as ENCA0 capture trigger 1 signal ENCATTIN1, the valid ADCA0TRG1 is input to ENCA0 as the ENCATTIN1 signal and ENCA0 is captured.

(2) When a hardware trigger signal (ADCA0TRG0, ADCA0TRG2) other than ADCA0TRG1 is generated, the ENCATTIN1 signal is not generated and ENCA0 is not captured.

36.10.5 Setup Flow

The setup flow in this section shows the general setup flow to perform the capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal.



Note 1. Change the settings depending on the hardware trigger to perform the capture operation.

Figure 36.23 Setup Flow

36.10.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal. Change the settings depending on the hardware trigger to perform the capture operation.

Table 36.42 ENCAn Setting

Register	Bit Position	Bit Name	Setting Value	Remark
ENCAnCTL	15	ENCAnCME	0	Disables compare match interrupt detection masking
	14	ENCAnMCS	0	Selects release trigger for compare match interrupt detection masking
	13 to 10		0	Fixed to 0
	9	ENCAnCRM1	1	Sets the ENCAnCCR1 register for capture operation
	8	ENCAnCRM0	Don’t care	Selects the function of ENCAnCCR0 register
	7	ENCAnCTS	0	Selects ENCATTIN1 as trigger for capture operation
	6, 5		0	Fixed to 0
	4	ENCAnLDE	Don’t care	Enables or disables reload operation when ENCAnCCR0 register underflow occurs
	3	ENCAnECM1	Don’t care	Enables or disables clearing of the counter on compare match of ENCAnCCR1 register
	2	ENCAnECM0	Don’t care	Enables or disables clearing of the counter on compare match of ENCAnCCR0 register
	1,0	ENCAnUDS[1:0]	Don’t care	Select the counter up/down control by ENCAnE0 and ENCAnE1
ENCAnIOC0	7 to 4		0	Fixed to 0
	3, 2	ENCAnTIS[3:2]	0[*1] 1[*1]	Select the valid edge of capture trigger 1 (ENCATTIN1) for the rising edge detection
	1, 0	ENCAnTIS[1:0]	Don’t care	Select the valid edge of capture trigger 0 (ENCATTIN0)
ENCAnIOC1	7	ENCAnSCE	Don’t care	Enables encoder special-clear
	6	ENCAnZCL	Don’t care	Selects the clear level (input level) of Z phase for encoder special-clear
	5	ENCAnBCL	Don’t care	Selects the clear level (input level) of B phase for encoder special-clear
	4	ENCAnACL	Don’t care	Selects the clear level (input level) of A phase for encoder special-clear
	3, 2	ENCAnECS[1:0]	Don’t care	Select encoder clear input (Z phase) edge
	1, 0	ENCAnEIS[1:0]	Don’t care	Select encoder input (A or B phase) edge
PIC0REG30	4	PIC0REG3004	Don’t care	Selects ADCA0 trigger signal of ENCATTIN1
	3	PIC0REG3003	Don’t care	
	2	PIC0REG3002	Don’t care	

Note 1. Change the setting depending on the hardware trigger to perform the capture operation.

NOTE

Bits ENCA0CRM1 and ENCA0CTS in ENCA0CTL are fixed: ENCA0CRM1 = 1 (ENCA0CCR1 register function) and ENCA0CTS = 0 (trigger source of capture to the ENCA0CCR1 register). All the other bits can be set arbitrarily.

36.11 Three-Phase PWM Output with Dead Time

36.11.1 Functional Overview

This feature generates each of the set signals (active level timing signals) and clear signals (inactive level timing signals) once or less per cycle and then uses the results to output a three-phase PWM waveform with dead time.

For the PWM output feature of TAUD, only the clear timing used during each cycle is specified by specifying the duty value, but for the feature described here, the set timing can also be specified, which makes more flexible PWM output with dead time possible.

36.11.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

Table 36.43 Configuration of Three-Phase PWM Output with Dead Time

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

(1) TAUDn Configuration

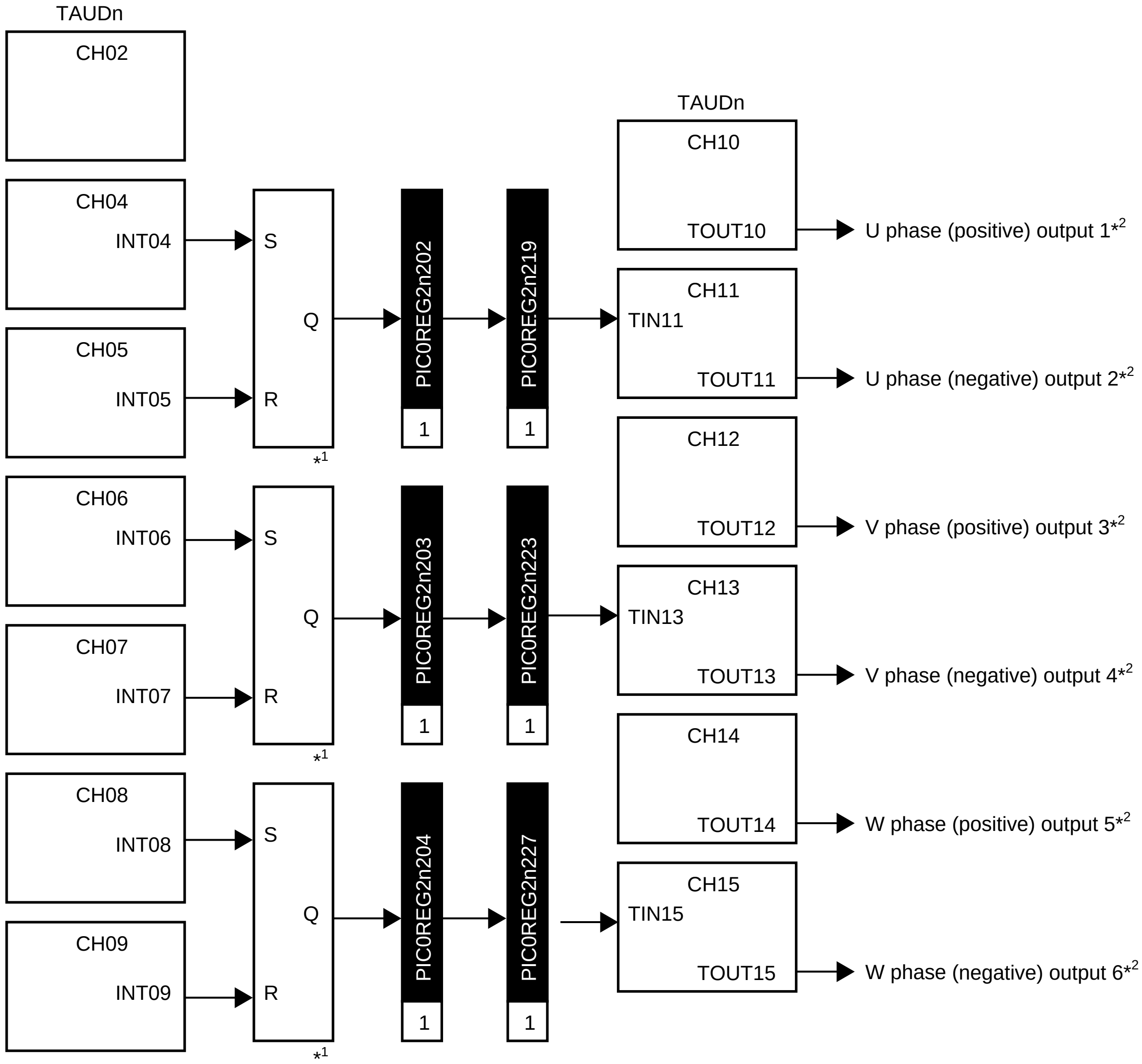
Because CH10, CH12, and CH14 are only used for TOUTm, these channels can be used for features that do not use TOUTm (m = 10, 12, 14).

Table 36.44 TAUDn Configuration

CH	Function Name	M/S	CDR Setting	Description
2	PWM output (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase signal setting)	
5		S	Duty (U phase signal clearing)	
6		S	Duty (V phase signal setting)	
7		S	Duty (V phase signal clearing)	
8		S	Duty (W phase signal setting)	
9		S	Duty (W phase signal clearing)	
10	Any feature that does not use TOUT10	S		TOUT10: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT11: UB phase output
12	Any feature that does not use TOUT12	S		TOUT12: V phase output
13	One-phase PWM output	S	Dead time (V phase)	TOUT13: VB phase output
14	Any feature that does not use TOUT14	S		TOUT14: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT15: WB phase output

Note: M: Master channel, S: Slave channel

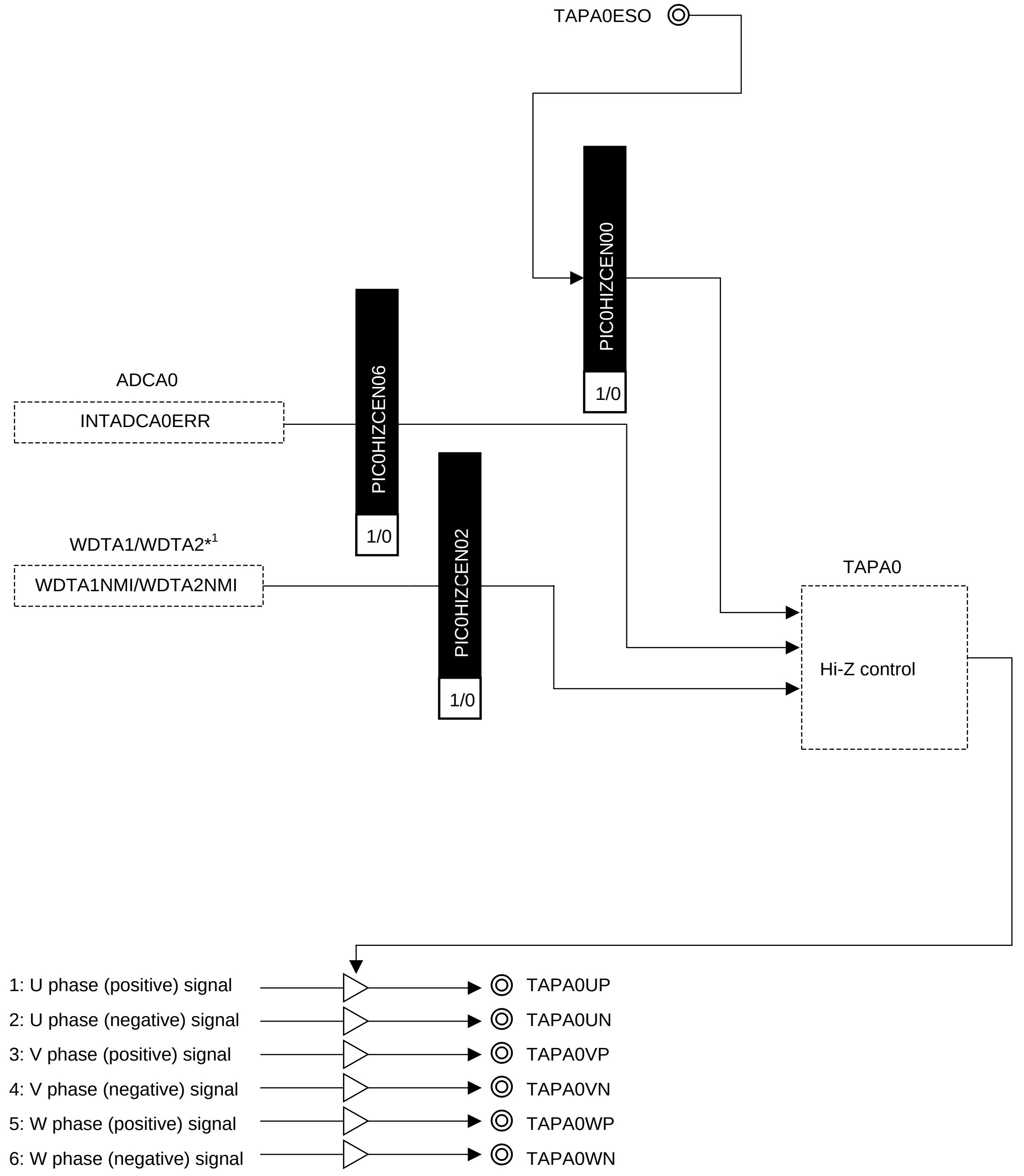
(2) Block Diagram



Note 1. SR flip-flop circuit

Note 2. For the connection destination, see **Figure 36.25, Block Diagram: Motor Output Buffer Control**.

Figure 36.24 Block Diagram: Three-Phase PWM Output with Dead Time



Note 1. WDTA2 is supported in RH850/F1KH-D8.

Note: For details about the blocks in dotted lines, see the corresponding section about each function.

- ADCA0: **Section 38, A/D Converter (ADCA)**
- TAPA0: **Section 36.4, Asynchronous Hi-Z Control Function**
- WDTA1/WDTA2: **Section 29, Window Watchdog Timer (WDTA)**

Figure 36.25 Block Diagram: Motor Output Buffer Control

36.11.3 Registers

36.11.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n227	—	—	—	PIC0REG2n223	—	—	—	PIC0REG2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 36.45 PIC0REG2n2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>Reserved</td><td>*1</td></tr>
<tr><td>27</td><td>PIC0REG2n227</td><td>Selects the signal input to TAUDTTIN15.
<table>
<tr><th>PIC0REG2n227</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n204 bit.</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>26 to 24</td><td>Reserved</td><td>*1</td></tr>
<tr><td>23</td><td>PIC0REG2n223</td><td>Selects the signal input to TAUDTTIN13.
<table>
<tr><th>PIC0REG2n223</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n203 bit.</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>22 to 20</td><td>Reserved</td><td>*1</td></tr>
<tr><td>19</td><td>PIC0REG2n219</td><td>Selects the signal input to TAUDTTIN11.
<table>
<tr><th>PIC0REG2n219</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n202 bit.</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>18 to 5</td><td>Reserved</td><td>*1</td></tr>
<tr><td>4</td><td>PIC0REG2n204</td><td>Selects the TIN input signal to TAUDTTIN15.
0: Setting prohibited
1: Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.</td></tr>
<tr><td>3</td><td>PIC0REG2n203</td><td>Selects the TIN input signal to TAUDTTIN13.
0: Setting prohibited
1: Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.</td></tr>
<tr><td>2</td><td>PIC0REG2n202</td><td>Selects the TIN input signal to TAUDTTIN11.
0: Setting prohibited
1: Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.</td></tr>
<tr><td>1, 0</td><td>Reserved</td><td>*1</td></tr>
</table>

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

36.11.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: <PIC0_base> + 80_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 36.46 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Selects whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	RH850/F1KH-D8 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI and the WDTA2NMI interrupt signal. 0: Disable 1: Enable RH850/F1KM-S4, RH850/F1KM-S1 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Selects whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

36.11.4 Operation Example

This example shows how to generate each of the set signals and clear signals once or less per cycle and then use the results to output a three-phase PWM waveform with dead time.

This is achieved by combining the following TAUD features:

- PWM output
- One-phase PWM output

In addition, the following peripheral interconnections are used to create the PWM waveform supplied from the set and clear signals generated during PWM output to the input TINm signal (m = 11, 13, or 15) of one-phase PWM output:

- SR flip-flop circuit

Three-phase PWM output is achieved by assigning the one-phase PWM output with dead time achieved using the above features to the U, V, and W phases. Therefore, the set and clear signals of PWM output can be freely specified for each PWM phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

36.11.4.1 Pwm Output

PWM output uses a combination of CH2, CH4, and CH5.

By specifying the cycle for CDR02, the U phase set value for CDR04, and the U phase clear value for CDR05, a set/clear signal is generated for the SR flip-flop circuit that generates the input TIN11 signal of one-phase PWM output from INT04 and INT05.

Instead of CH4 and CH5, which are used for the above described U phase set/clear signal generation, the V phase uses CH6 and CH7, and the W phase uses CH8 and CH9.

36.11.4.2 One-Phase PWM Output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM signal with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM signal with dead time, while the W phase uses CH14 and CH15.

CAUTION

Specify the same clock for each TAUDn channel that uses the PWM output and one-phase PWM output features.

For details about the TAUD functions, see **Section 32, Timer Array Unit D (TAUD)**.

36.11.4.3 SR Flip-Flop Circuit

The PWM waveform supplied to input TIN11 of one-phase PWM output is generated by using the U phase set signal generated by CH4 of TAUD and the U phase clear signal generated by CH5.

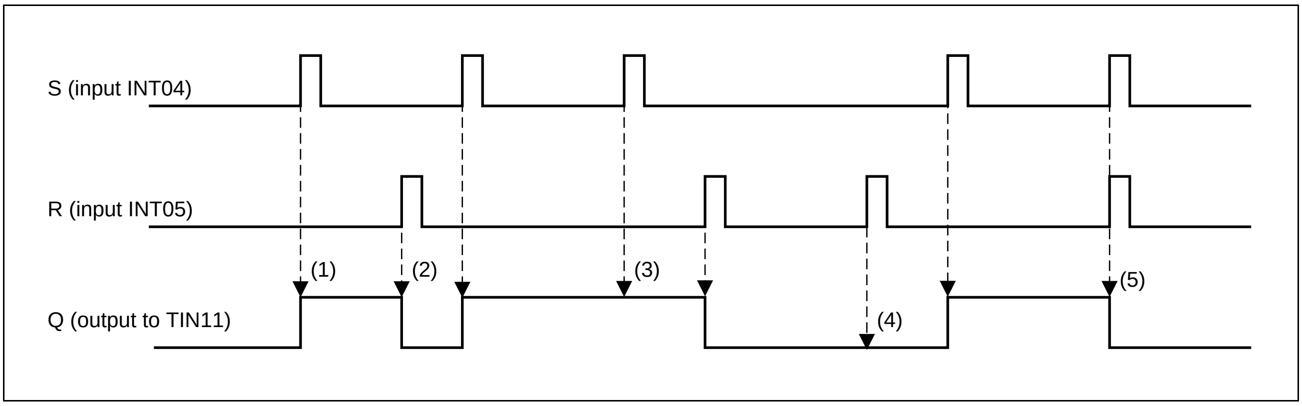


Figure 36.26 SR Flip-Flop Circuit Operation Timing Chart (U phase example)

(1) When a signal is input to input S, output Q goes to the high level at the rising edge of S.

(2) When a signal is input to input R, output Q goes to the low level at the rising edge of R.

(3) If a signal is input to input S while output Q is at the high level, output Q is not affected.

(4) If a signal is input to input R while output Q is at the low level, output Q is not affected.

(5) If a signal is input to input S and input R at the same time, input R is prioritized and output Q goes to the low level at the rising edge of R.

The V phase uses INT06 and INT07 as input to supply a PWM waveform to TIN13, and the W phase uses INT08 and INT09 as input to supply a PWM waveform to TIN15.

The output change timing of the PWM waveform generated during one-phase PWM output is based on PWM output.

The active level output timing set signal and inactive level output timing clear signal of PWM are generated during PWM output. By inputting these signals to the SR flip-flop circuit, a PWM signal that can be changed at any time is generated.

A one-phase PWM signal is output by generating a positive or negative PWM waveform and then adding dead time to it according to changes in the generated PWM signal.

PIC is used to set/clear signal generated during PWM output as the TIN input for one-phase PWM output through the SR flip-flop circuit.

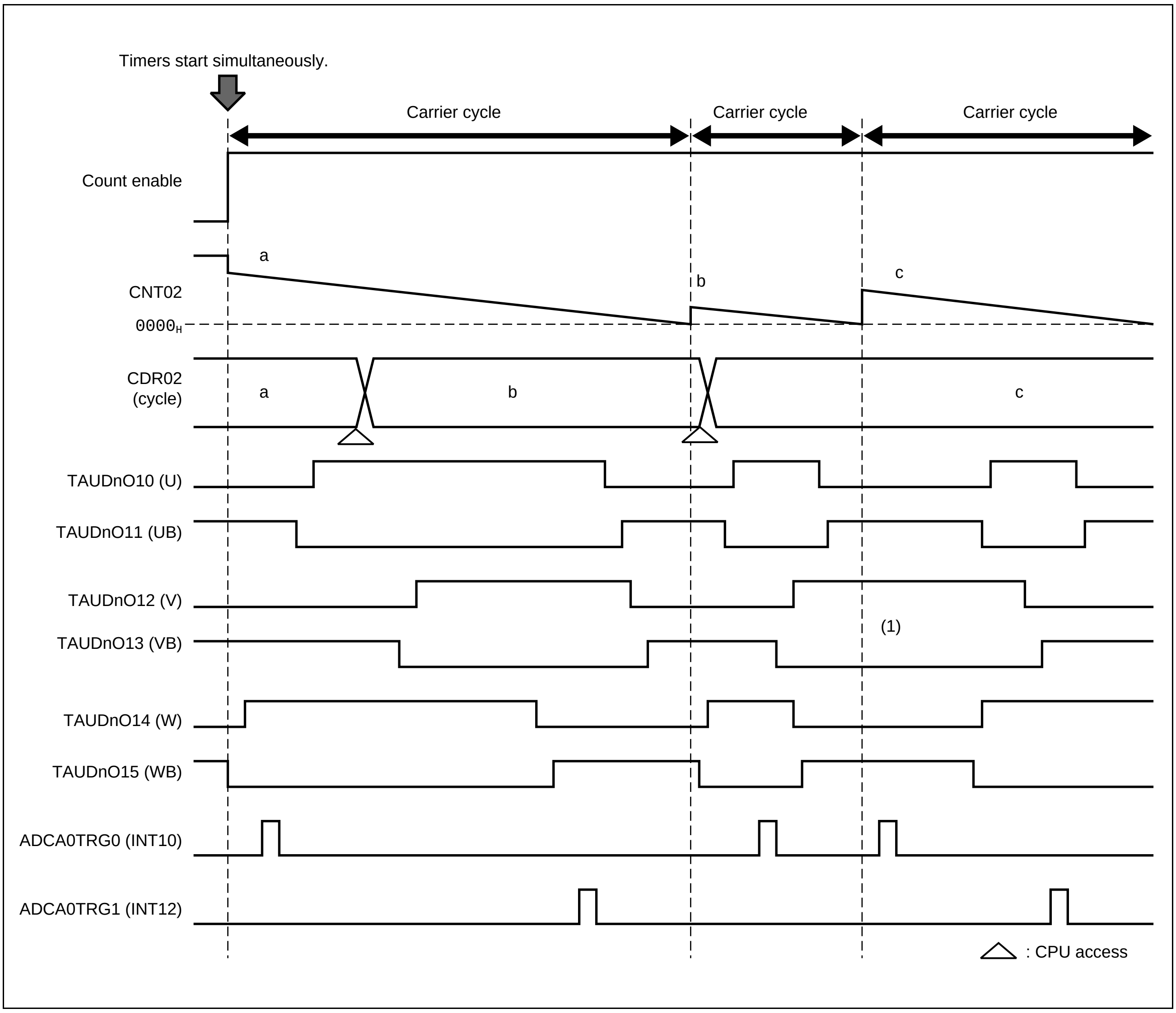


Figure 36.27 Example of Three-Phase PWM (U/UB, V/VB, W/WB) Output with Dead Time

Figure 36.27, Example of Three-Phase PWM (U/UB, V/VB, W/WB) Output with Dead Time shows a typical example of three-phase PWM output with dead time.

By appropriately setting up the set/clear signal output timing, PWM output that extends across carrier cycles (point (1)) and other types of output are also possible.

In this example, ADCA0TRG0 and ADCA0TRG1 (which are at the bottom) use the CNT and INT signals of CH10 and CH12, which are not used for one-phase PWM output, and the A/D trigger signal is output by performing type-1 A/D trigger output.

In this way, because only the TOUTm signal that performs signal output for the channel performing positive phase output is used during one-phase PWM output, any feature that uses CNTm, CDRm, or INTm can be specified. For details, see **Section 32, Timer Array Unit D (TAUD)** (m = 10, 12, or 14).

The following figures show timing charts for outputting a three-phase PWM signal with dead time.

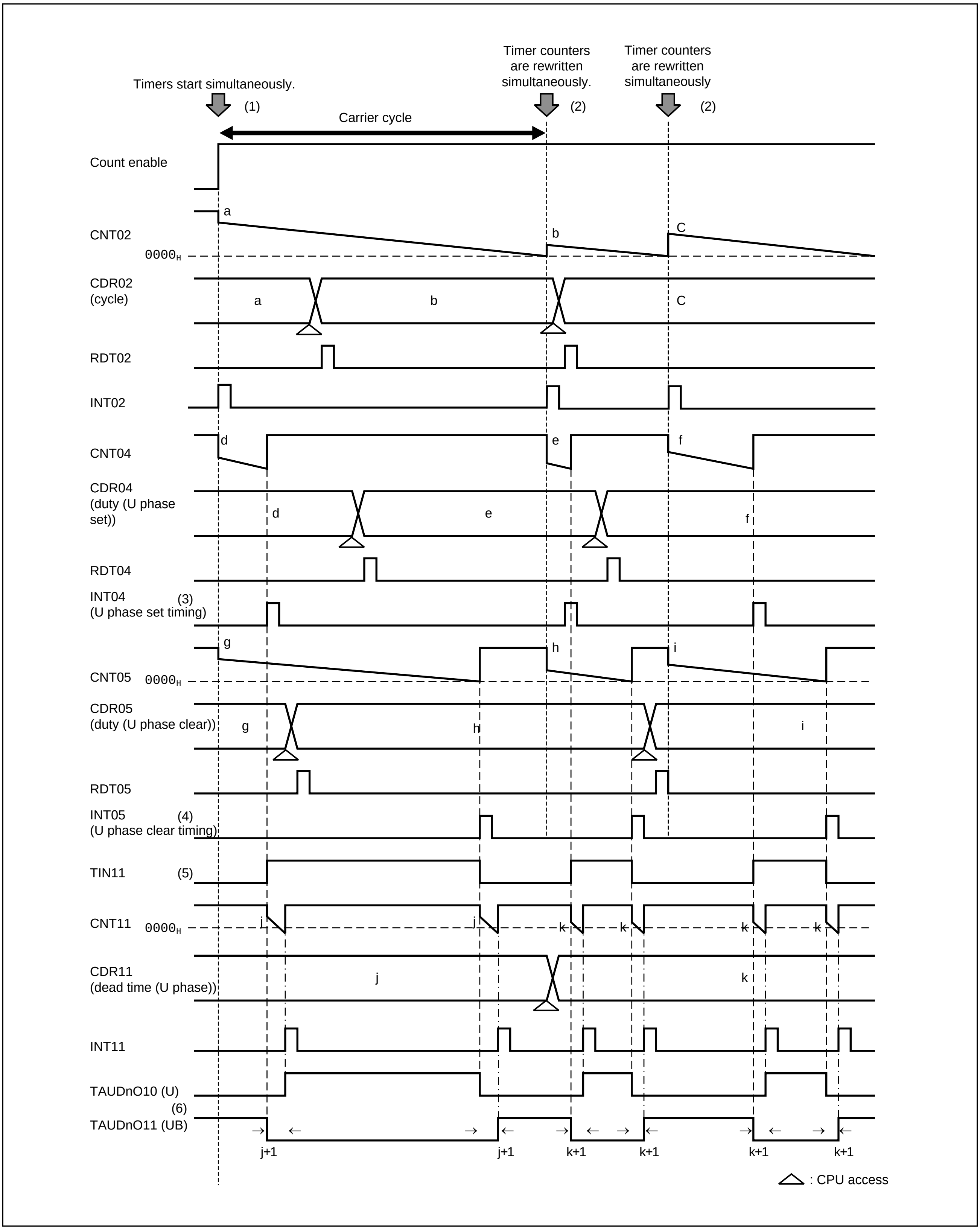


Figure 36.28 Example of One-Phase PWM (U phase, UB phase) Output with Dead Time

An operation example of the timer configuration for performing the U phase PWM output in **Figure 36.28, Example of One-Phase PWM (U phase, UB phase) Output with Dead Time** is provided below.

(1) By simultaneously starting timers, CH2 (the carrier cycle timer), CH4 (the U phase set signal output timing timer), and CH5 (the U phase clear signal output timing timer) are started simultaneously.
The CH11 timer is also enabled, but until a TIN11 edge is detected, which is the count start timing, counting is not performed.

(2) For CH4 and CH5, when there is a CH2 underflow, the settings from CDR04 and CDR05 are reloaded to CNT04 and CNT05.

(3) When there is a CH4 underflow, the U phase set timing signal (INT04) is generated.

(4) When there is a CH5 underflow, the U phase clear timing signal (INT05) is generated.

(5) The peripheral interconnections supply the output of the SR flip-flop circuit that uses INT04 (the set timing signal) and INT05 (the clear timing signal) as input to the input TIN11 signal of one-phase PWM output.

(6) During one-phase PWM output, a PWM waveform with dead time is output by detecting a TIN11 edge.

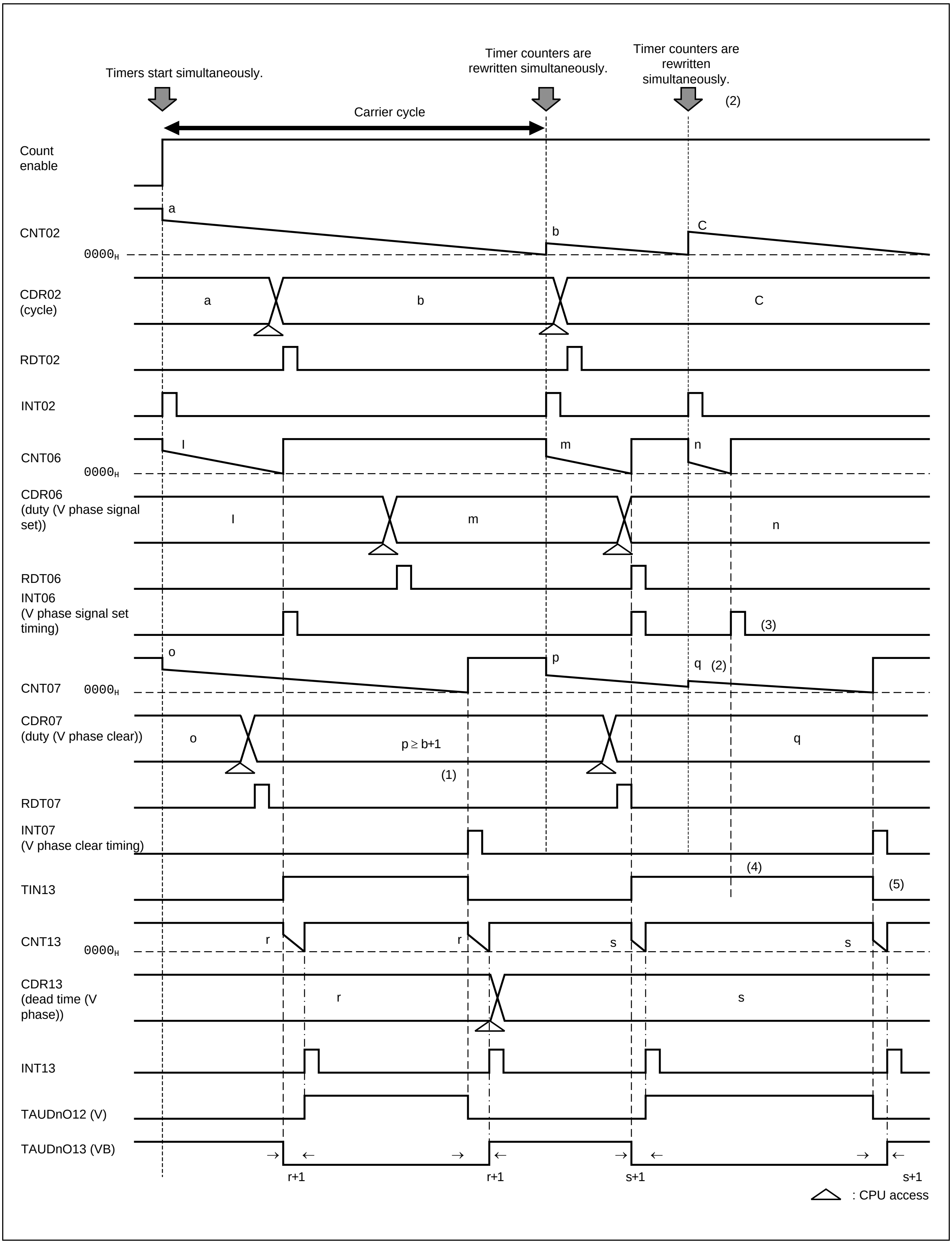


Figure 36.29 Example of One-Phase PWM (V phase, VB phase) Output with Dead Time

An operation example of the timer configuration for performing the V phase PWM output in **Figure 36.29, Example of One-Phase PWM (V phase, VB phase) Output with Dead Time** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

(1) If the setting of CH7 (the V phase clear signal output timing timer), which generates the V phase clear timing signal (INT07), is greater than the CH2 (the carrier cycle timer) setting.

(2) Before a V phase clear timing signal (INT07) is generated by a CH7 underflow, a CH2 (carrier cycle timer) underflow occurs, and the CH7 setting is reloaded.

(3) It causes the V phase clear timing signal (INT07) not to be generated, resulting in consecutive generation of the V phase set timing signal (INT06).

(4) In this case, because the V phase set timing signal (INT06) is ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform. Therefore, a PWM waveform that extends across carrier cycles is output.

(5) The PWM output is changed at the timing of the next V phase clear timing signal (INT07).

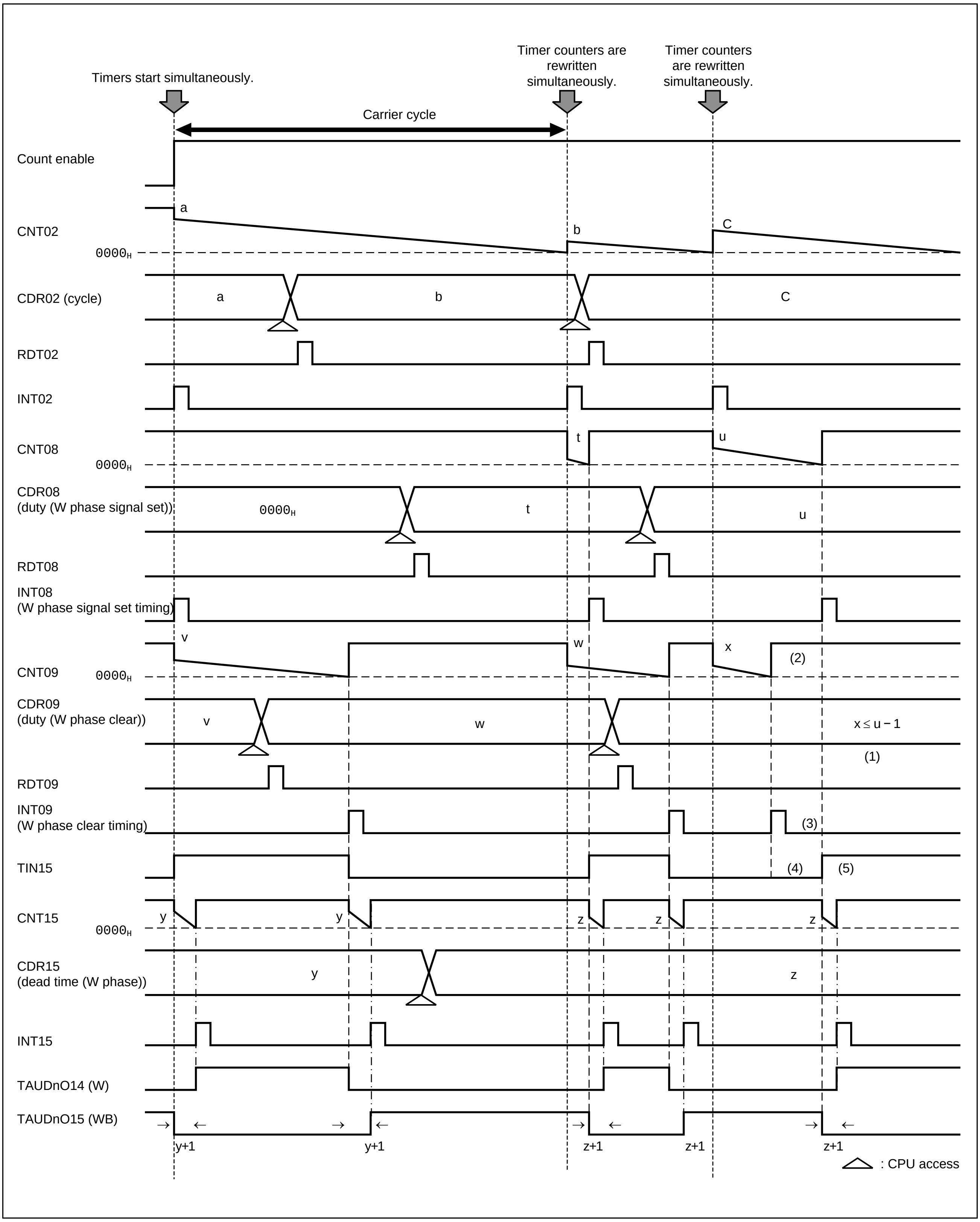


Figure 36.30 Example of One-Phase PWM (W phase, WB phase) Output with Dead Time

An operation example of the timer configuration for performing the W phase PWM output in **Figure 36.30, Example of One-Phase PWM (W phase, WB phase) Output with Dead Time** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

(1) If the setting of CH9 (the W phase clear signal output timing timer), which generates the W phase clear timing signal (INT09), is less than the CH8 (the W phase set signal output timing timer) setting.

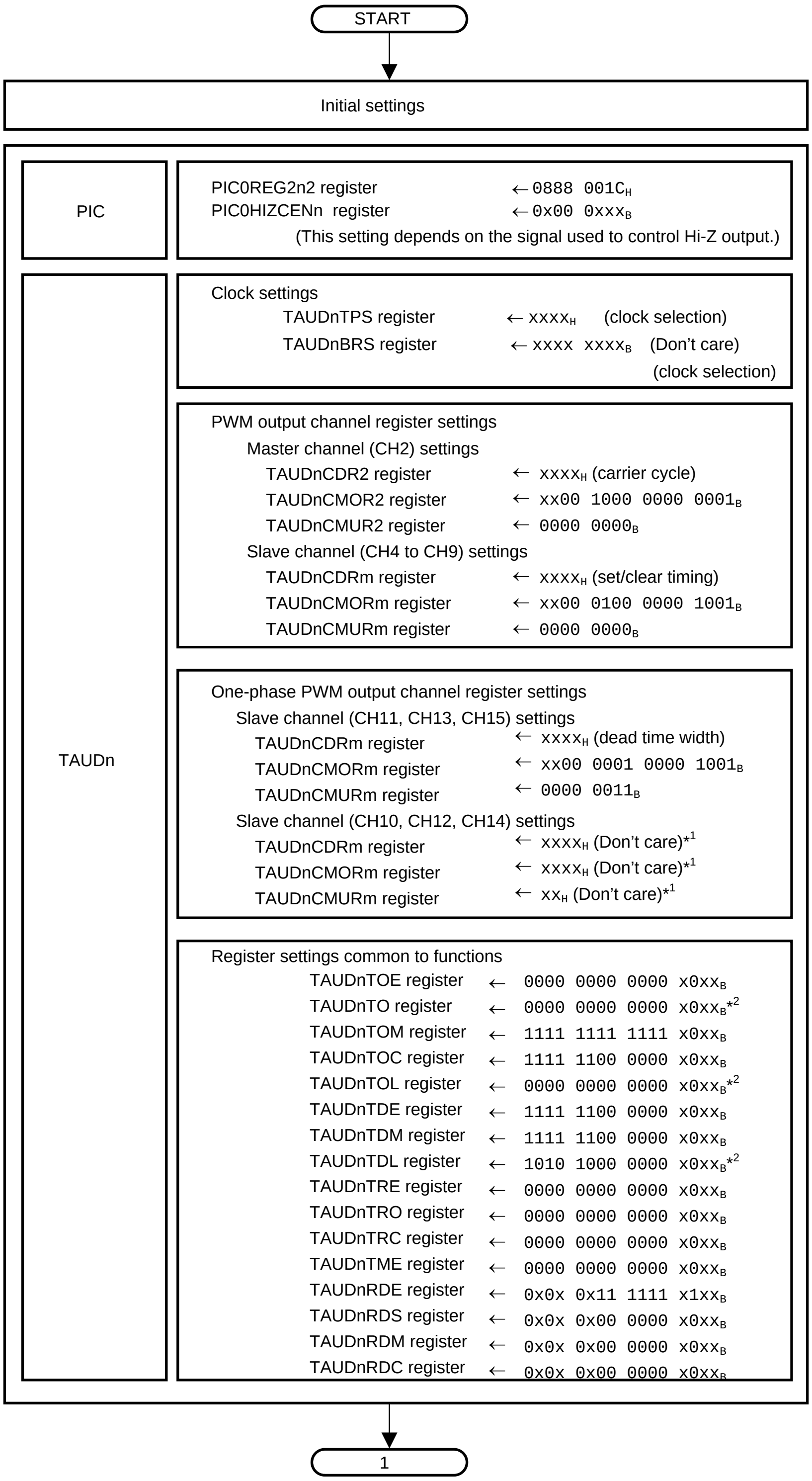
(2) Before a W phase set timing signal (INT08) is generated by a CH8 underflow, a CH9 (W phase clear signal output timing timer) underflow occurs, and the W phase clear timing signal (INT09) is generated.

(3) This results in consecutive W phase clear timing signals (INT09) being generated.

(4) In this case, because the consecutively generated W phase clear timing signals (INT09) are ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform.

(5) The PWM output is changed at the timing of the next W phase set timing signal (INT08).

36.11.5 Setup Flow



PIC0REG2n2 register ← 0888 001C$_H$
PIC0HIZCENn register ← 0x00 0xxx$_B$
(This setting depends on the signal used to control Hi-Z output.)

Clock settings
TAUDnTPS register ← xxxx$_H$ (clock selection)
TAUDnBRS register ← xxxx xxxx$_B$ (Don't care) (clock selection)

PWM output channel register settings
Master channel (CH2) settings
TAUDnCDR2 register ← xxxx$_H$ (carrier cycle)
TAUDnCMOR2 register ← xx00 1000 0000 0001$_B$
TAUDnCMUR2 register ← 0000 0000$_B$
Slave channel (CH4 to CH9) settings
TAUDnCDRm register ← xxxx$_H$ (set/clear timing)
TAUDnCMORm register ← xx00 0100 0000 1001$_B$
TAUDnCMURm register ← 0000 0000$_B$

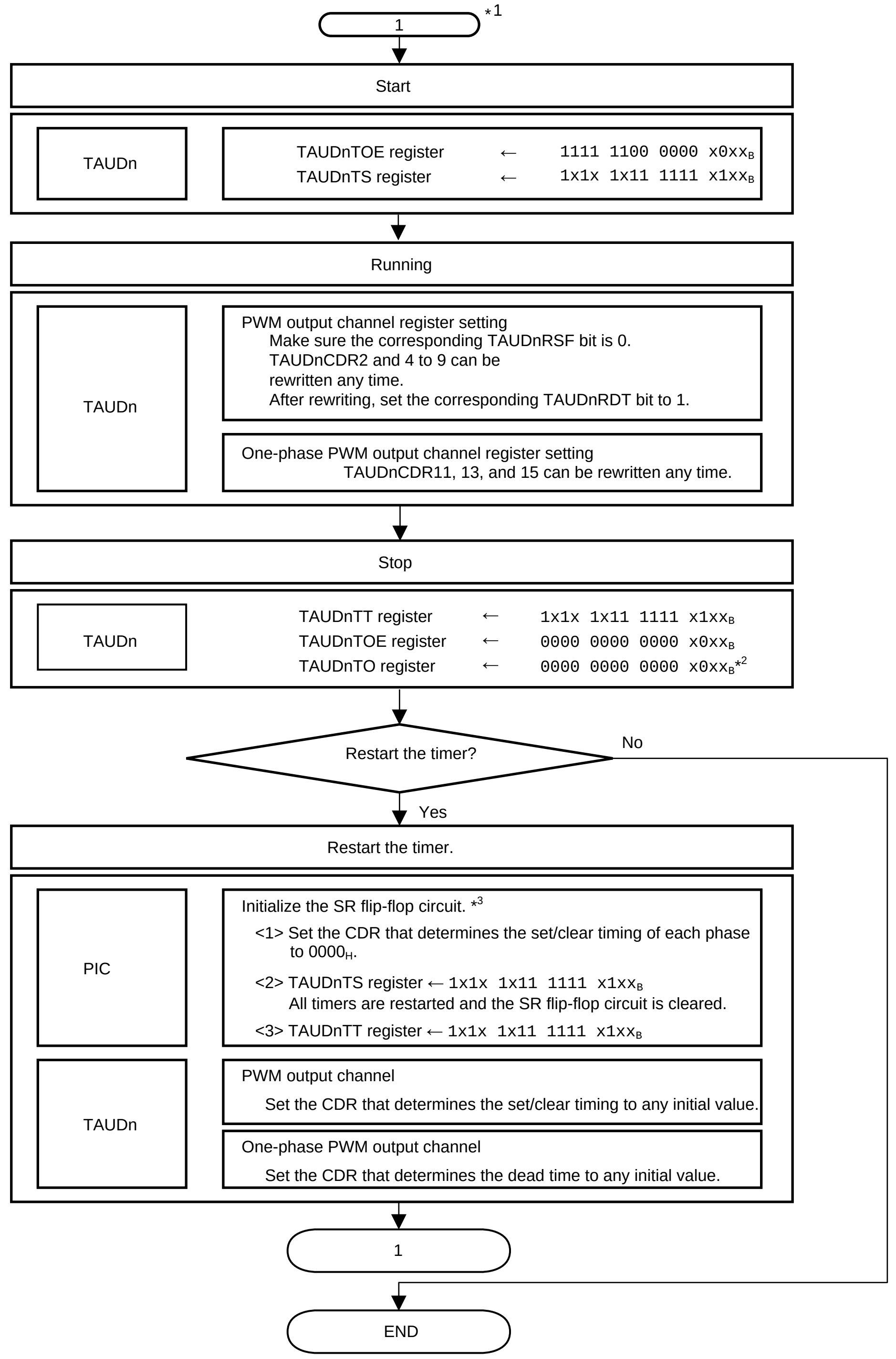
One-phase PWM output channel register settings
Slave channel (CH11, CH13, CH15) settings
TAUDnCDRm register ← xxxx$_H$ (dead time width)
TAUDnCMORm register ← xx00 0001 0000 1001$_B$
TAUDnCMURm register ← 0000 0011$_B$
Slave channel (CH10, CH12, CH14) settings
TAUDnCDRm register ← xxxx$_H$ (Don't care)*1
TAUDnCMORm register ← xxxx$_H$ (Don't care)*1
TAUDnCMURm register ← xx$_H$ (Don't care)*1

Register settings common to functions
TAUDnTOE register ← 0000 0000 0000 x0xx$_B$
TAUDnTO register ← 0000 0000 0000 x0xx$_B$*2
TAUDnTOM register ← 1111 1111 1111 x0xx$_B$
TAUDnTOC register ← 1111 1100 0000 x0xx$_B$
TAUDnTOL register ← 0000 0000 0000 x0xx$_B$*2
TAUDnTDE register ← 1111 1100 0000 x0xx$_B$
TAUDnTDM register ← 1111 1100 0000 x0xx$_B$
TAUDnTDL register ← 1010 1000 0000 x0xx$_B$*2
TAUDnTRE register ← 0000 0000 0000 x0xx$_B$
TAUDnTRO register ← 0000 0000 0000 x0xx$_B$
TAUDnTRC register ← 0000 0000 0000 x0xx$_B$
TAUDnTME register ← 0000 0000 0000 x0xx$_B$
TAUDnRDE register ← 0x0x 0x11 1111 x1xx$_B$
TAUDnRDS register ← 0x0x 0x00 0000 x0xx$_B$
TAUDnRDM register ← 0x0x 0x00 0000 x0xx$_B$
TAUDnRDC register ← 0x0x 0x00 0000 x0xx$_B$

Note 1. Specify a feature that does not use TOUTm.

Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 36.31 Setup Flow (Active High Example)



Note 1. Specify the selection register and output port to use after specifying the initial settings for the peripheral interconnections and timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Note 3. If initialization processing is not performed, the SR flip-flop circuit enters the timer-stopped state, and the timer restart output pulse might be output at an unintended level.

Figure 36.32 Setup Flow (Active High Example) (continued)

36.11.6 Setting Examples for Operation Functions

This section provides example settings for each register.

36.11.6.1 TAUDn Settings (Active High Example)

Table 36.47 TAUDn: CH2-related (PWM Output Master Channel*[1])

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 36.48 TAUDn: CH4 to CH9-related (PWM Output Slave Channel*[1]) (m = 4 to 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Any*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register used during PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Table 36.49 TAUDn: CH11, CH13, and CH15-related (One-phase PWM Output) (m = 11, 13, or 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*[1]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output). For details, see **Section 32, Timer Array Unit D (TAUD)**.

Table 36.50 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0	These are fixed to 0 because TOUT09 to TOUT04 are not used.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*[1]	Output a low-level signal to TOUT15 to TOUT10.
	9 to 4	TAUDnTO09 to TAUDnTO04	0	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	
TAUDnTOM	15 to 4	TAUDnTOM15 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't care	

Table 36.50 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	0	Synchronous operation mode 1
	3	TAUDnTOC03	Don’t care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don’t care	
TAUDnTOL	15 to 4	TAUDnTOL15 to TAUDnTOL04	0*1	Positive logic output (active high)
	3	TAUDnTOL03	Don’t care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don’t care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control.*2
	9 to 4	TAUDnTDE09 to TAUDnTDE04	0	Disable dead time control.
	3	TAUDnTDE03	Don’t care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don’t care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 4	TAUDnTDM09 to TAUDnTDM04	0	Invalid because dead time control is disabled.
	3	TAUDnTDM03	Don’t care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don’t care	
TAUDnTDL	15	TAUDnTDL15	1*1	Dead time is in the negative segment of the W phase output
	14	TAUDnTDL14	0*1	Dead time is in the positive segment of the W phase output
	13	TAUDnTDL13	1*1	Dead time is in the negative segment of the V phase output
	12	TAUDnTDL12	0*1	Dead time is in the positive segment of the V phase output
	11	TAUDnTDL11	1*1	Dead time is in the negative segment of the U phase output
	10	TAUDnTDL10	0*1	Dead time is in the positive segment of the U phase output
	9 to 4	TAUDnTDL09 to TAUDnTDL04	0	Invalid because dead time control is disabled.
	3	TAUDnTDL03	Don’t care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don’t care	
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Stop real-time output.
	3	TAUDnTRE03	Don’t care	
	2	TAUDnTRE02	0	Stop real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don’t care	

Table 36.50 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't care	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't care	
	9 to 4	TAUDnRDE09 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 4	TAUDnRDS09 to TAUDnRDS04	0	Enable simultaneous rewriting by using a master channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't care	

Table 36.50 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't care	
	9 to 4	TAUDnRDM09 to TAUDnRDM04	0	Load the signal when the master channel starts counting.
	3	TAUDnRDM03	Don't care	
	2	TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't care	
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't care	
	9 to 4	TAUDnRDC09 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't care	
	2	TAUDnRDC02	1	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 32, Timer Array Unit D (TAUD)**.

36.11.6.2 PIC Settings

Table 36.51 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	1	Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
	3	PIC0REG2n203	1	Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
	2	PIC0REG2n202	1	Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.

36.12 High-accuracy Triangle PWM Output with Dead Time

36.12.1 Functional Overview

Compared to the triangle PWM output with dead time of TAUD, this makes it possible to control the variable dead time areas near duties of 100% and 0%. This makes more accurate triangle PWM output possible.

For the triangle PWM output with dead time feature of TAUD, it is not possible to output a UB phase dead time pulse, such as when transitioning to U phase 0% triangular wave output. (See **Figure 36.33, Timing of Dead Time Output by the TAUD Feature for Outputting a Triangle PWM Signal with Dead Time**)

For this feature, a pulse is generated in combination with the TAUD timer output, and a pseudo dead time pulse is added.

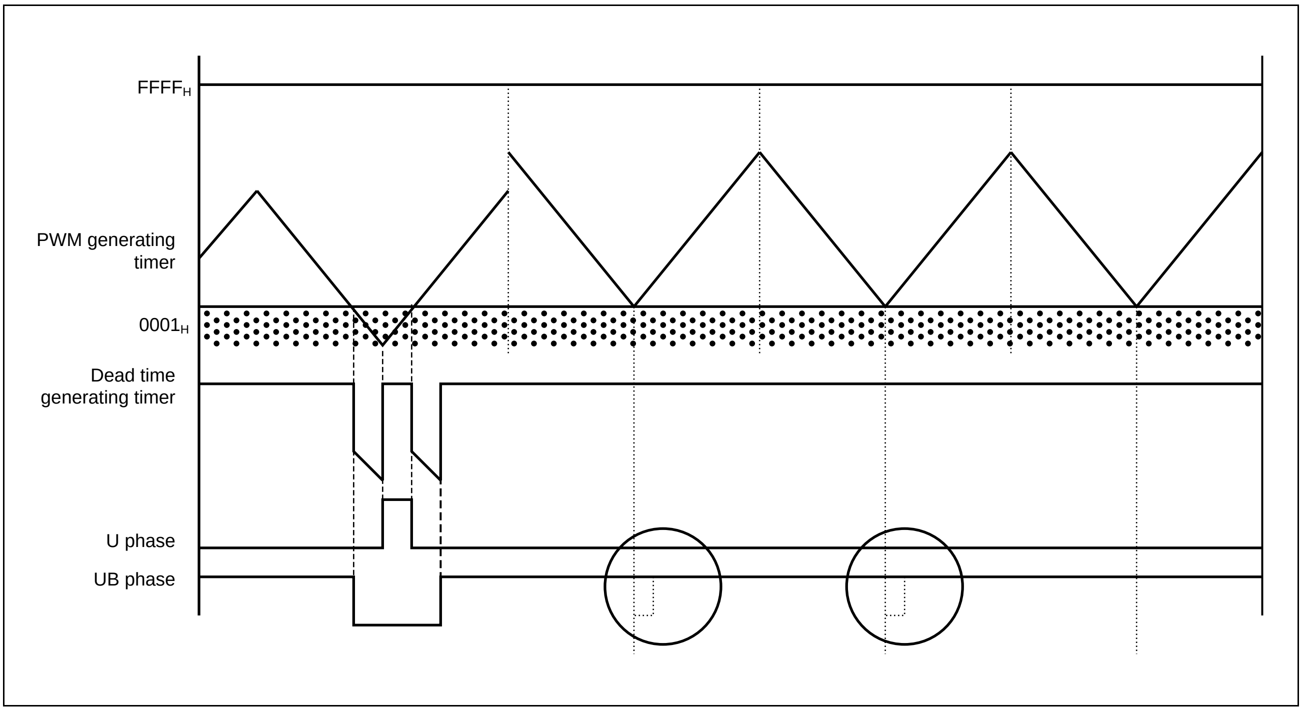


Figure 36.33 Timing of Dead Time Output by the TAUD Feature for Outputting a Triangle PWM Signal with Dead Time

36.12.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

Table 36.52 Configuration of Delay Pulse Output with Dead Time

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

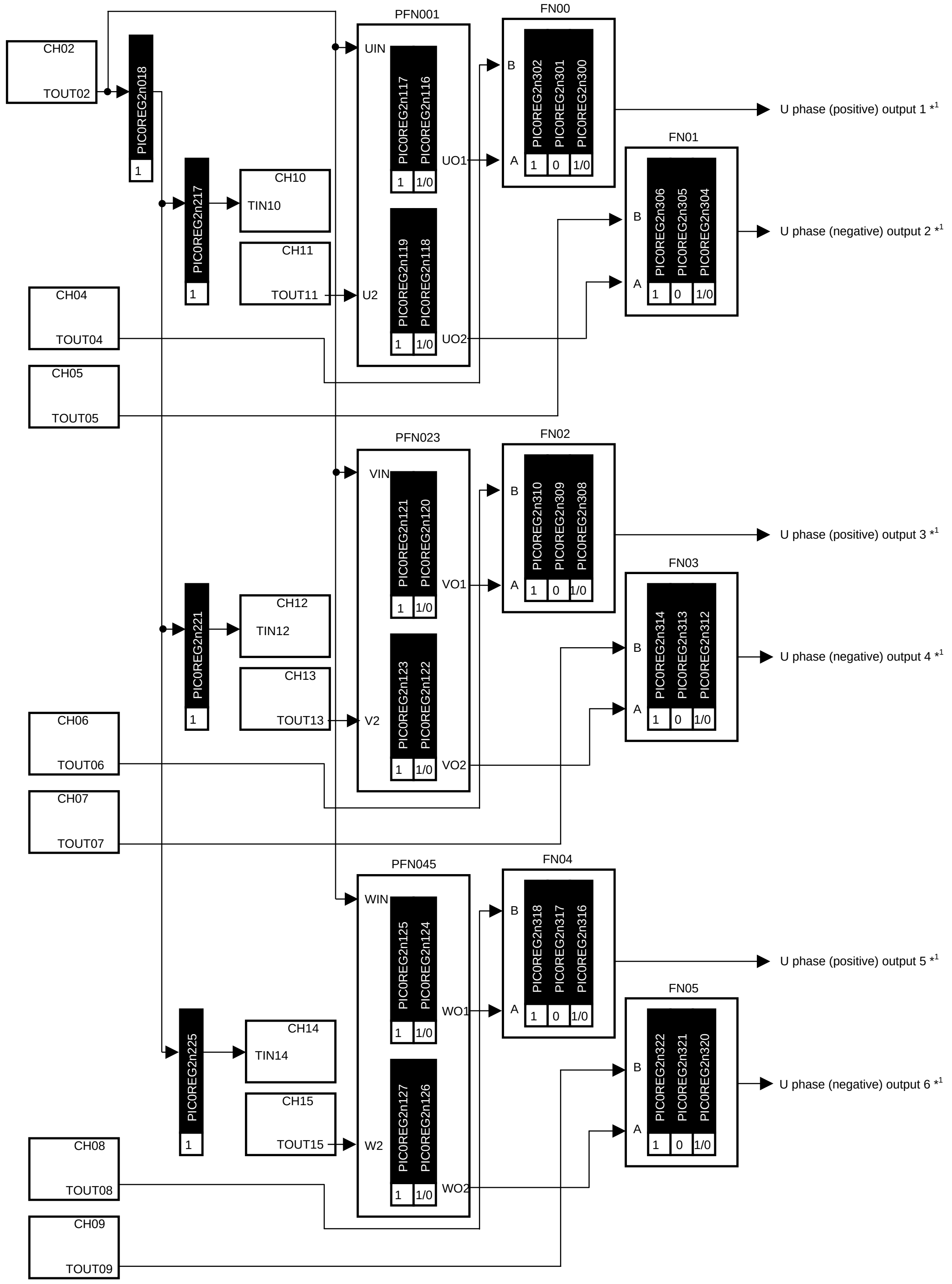
(1) TAUDn Configuration

Table 36.53 TAUD Configuration

CH	Function Name	M/S*1	CDR Setting	Description
2	Triangle PWM output with dead time (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase)	
5		S	Dead time (U phase)	
6		S	Duty (V phase)	
7		S	Dead time (V phase)	
8		S	Duty (W phase)	
9		S	Dead time (W phase)	
10	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for U phase PWM.
11		S	Pulse width	
12	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for V phase PWM.
13		S	Pulse width	
14	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for W phase PWM.
15		S	Pulse width	

Note 1. M: Master channel, S: Slave channel

(2) Block Diagram



Note 1. For the connection destination, see **Figure 36.25, Block Diagram: Motor Output Buffer Control**.

Figure 36.34 Block Diagram: High-Accuracy Triangle PWM Output with Dead Time

36.12.3 Registers

36.12.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects TAUDn input.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG200: FFDD 00C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n018	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.54 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	*[1]
18	PIC0REG2n018	Select the TIN input signal to TAUDTTIN10, TAUDTTIN12, and TAUDTTIN14. 0: Setting prohibited 1: Select TAUDTTOUT2.
17 to 0	Reserved	*[1]

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

36.12.3.2 PIC0REG2n1 — Timer I/O Control Register 2n1 (n = 0)

This register selects the logic of a combination circuit.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG201: FFDD 00C4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n127	PIC0REG2n126	PIC0REG2n125	PIC0REG2n124	PIC0REG2n123	PIC0REG2n122	PIC0REG2n121	PIC0REG2n120	PIC0REG2n119	PIC0REG2n118	PIC0REG2n117	PIC0REG2n116
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.55 PIC0REG2n1 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>27, 26</td><td>PIC0REG2n127
PIC0REG2n126</td><td>Select the FN05 A input signal according to the output logic specified for CH9 of TAUDn.
<table>
<tr><th>PIC0REG2n127</th><th>PIC0REG2n126</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL09 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL09 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>25, 24</td><td>PIC0REG2n125
PIC0REG2n124</td><td>Select the FN04 A input signal according to the output logic specified for CH8 of TAUDn.
<table>
<tr><th>PIC0REG2n125</th><th>PIC0REG2n124</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL08 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL08 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>23, 22</td><td>PIC0REG2n123
PIC0REG2n122</td><td>Select the FN03 A input signal according to the output logic specified for CH7 of TAUDn.
<table>
<tr><th>PIC0REG2n123</th><th>PIC0REG2n122</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL07 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL07 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Table 36.55 PIC0REG2n1 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>21, 20</td><td>PIC0REG2n121
PIC0REG2n120</td><td>Select the FN02 A input signal according to the output logic specified for CH6 of TAUDn.
<table>
<tr><th>PIC0REG2n121</th><th>PIC0REG2n120</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL06 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL06 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>19, 18</td><td>PIC0REG2n119
PIC0REG2n118</td><td>Select the FN01 A input signal according to the output logic specified for CH5 of TAUDn.
<table>
<tr><th>PIC0REG2n119</th><th>PIC0REG2n118</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL05 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL05 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>17, 16</td><td>PIC0REG2n117
PIC0REG2n116</td><td>Select the FN00 A input signal according to the output logic specified for CH4 of TAUDn.
<table>
<tr><th>PIC0REG2n117</th><th>PIC0REG2n116</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL04 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL04 = 1).)</td></tr>
<tr><td colspan="2">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>15 to 0</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
</table>

36.12.3.3 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0REG2n225	—	—	—	PIC0REG2n221	—	—	—	PIC0REG2n217	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.56 PIC0REG2n2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 26</td><td>Reserved</td><td>*1</td></tr>
<tr><td>25</td><td>PIC0REG2n225</td><td>Select the TIN input signal to TAUDTTIN14.
<table>
<tr><th>PIC0REG2n225</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>24 to 22</td><td>Reserved</td><td>*1</td></tr>
<tr><td>21</td><td>PIC0REG2n221</td><td>Select the TIN input signal to TAUDTTIN12.
<table>
<tr><th>PIC0REG2n221</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>20 to 18</td><td>Reserved</td><td>*1</td></tr>
<tr><td>17</td><td>PIC0REG2n217</td><td>Select the TIN input signal to TAUDTTIN10.
<table>
<tr><th>PIC0REG2n217</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>16 to 0</td><td>Reserved</td><td>*1</td></tr>
</table>

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

36.12.3.4 PIC0REG2n3 — Timer I/O Control Register 2n3 (n = 0)

This register selects the logic of a combination circuit.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG203: FFDD 00CC$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC0REG2n322	PIC0REG2n321	PIC0REG2n320	—	PIC0REG2n318	PIC0REG2n317	PIC0REG2n316
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0REG2n314	PIC0REG2n313	PIC0REG2n312	—	PIC0REG2n310	PIC0REG2n309	PIC0REG2n308	—	PIC0REG2n306	PIC0REG2n305	PIC0REG2n304	—	PIC0REG2n302	PIC0REG2n301	PIC0REG2n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 36.57 PIC0REG2n3 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 23</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>22
21
20</td><td>PIC0REG2n322
PIC0REG2n321
PIC0REG2n320</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH9 of TAUDn.
<table>
<tr><th>PIC0REG 2n322</th><th>PIC0REG 2n321</th><th>PIC0REG 2n320</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL09 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL09 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>19</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>18
17
16</td><td>PIC0REG2n318
PIC0REG2n317
PIC0REG2n316</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH8 of TAUDn.
<table>
<tr><th>PIC0REG 2n318</th><th>PIC0REG 2n317</th><th>PIC0REG 2n316</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL08 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL08 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>15</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
</table>

Table 36.57 PIC0REG2n3 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>14
13
12</td><td>PIC0REG2n314
PIC0REG2n313
PIC0REG2n312</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH7 of TAUDn.
<table>
<tr><th>PIC0REG 2n314</th><th>PIC0REG 2n313</th><th>PIC0REG 2n312</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL07 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL07 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>11</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>10
9
8</td><td>PIC0REG2n310
PIC0REG2n309
PIC0REG2n308</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH6 of TAUDn.
<table>
<tr><th>PIC0REG 2n310</th><th>PIC0REG 2n309</th><th>PIC0REG 2n308</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL06 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL06 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>6
5
4</td><td>PIC0REG2n306
PIC0REG2n305
PIC0REG2n304</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH5 of TAUDn.
<table>
<tr><th>PIC0REG 2n306</th><th>PIC0REG 2n305</th><th>PIC0REG 2n304</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL05 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL05 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3</td><td>Reserved</td><td>When read, the value after reset is returned.
When writing, write the value after reset.</td></tr>
<tr><td>2
1
0</td><td>PIC0REG2n302
PIC0REG2n301
PIC0REG2n300</td><td>Select the logical operation to perform on input signals A and B according to the output logic specified for CH4 of TAUDn.
<table>
<tr><th>PIC0REG 2n302</th><th>PIC0REG 2n301</th><th>PIC0REG 2n300</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>A and B
(Select this when the active high setting is specified (TAUDnTOL04 = 0).)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>A or B
(Select this when the active low setting is specified (TAUDnTOL04 = 1).)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

36.12.3.5 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: PIC0HIZCEN0: FFDD 0080$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 36.58 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	RH850/F1KH-D8 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI and the WDTA2NMI interrupt signal. 0: Disable 1: Enable RH850/F1KM-S4, RH850/F1KM-S1 Select whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

36.12.4 Operation Example

This is achieved by combining the following TAUD features:

- Triangle PWM output with dead time
- One-shot pulse output

In addition, the following PIC is also used because the pulse to be inserted into the variable dead time area is generated for the positive or negative phase:

- Combination circuit (PFN001, PFN023, and PFN045)

In addition, the following peripheral interconnections are also used because the pulse to be inserted into the variable dead time area is combined with the triangle PWM output waveform:

- Logical operation circuit (FN0i) (i = 0 to 5)

A high-accuracy triangle PWM signal with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

36.12.4.1 Triangle PWM Output with Dead Time

A triangle PWM signal with dead time is output from TOUT04 and TOUT05 by using CH2, CH4, and CH5 in combination.

36.12.4.2 One-shot Pulse Output

A CDR11 pulse for which the width is delayed by the delay time (CDR10) from the valid edge of the TIN10 (TOUT02) signal of CH10 is output as TOUT11 by using CH10 and CH11 in combination.

This pulse is used as the variable dead time area pulse used near duties of 100% and 0%.

CAUTION

Specify each CDR setting for one-shot pulse output such that the following condition is satisfied: CDR05 ≥ (CDR10 + CDR11)

If a value that does not satisfy the above condition is specified, the output waveform might be affected. To minimize this effect, in addition to satisfying the above setting condition, leave CDR11 set to 0000_H until the variable dead time area pulse is required.
Detect both rising and falling edges as the valid TIN10 (TOUT02) edge, and set TAUDnTOL11 to 1 (active low).
Specify the same operation clock for each TAUDn channel used for outputting a triangle PWM signal with dead time or a one-shot pulse.

For details about the TAUD functions, see **Section 32, Timer Array Unit D (TAUD)**.

36.12.4.3 U phase Combination Circuit (PFN001)

This circuit generates a variable dead time area pulse (FN00 A, FN01 A) for adding a generated one-shot pulse to a generated triangle PWM signal with dead time.

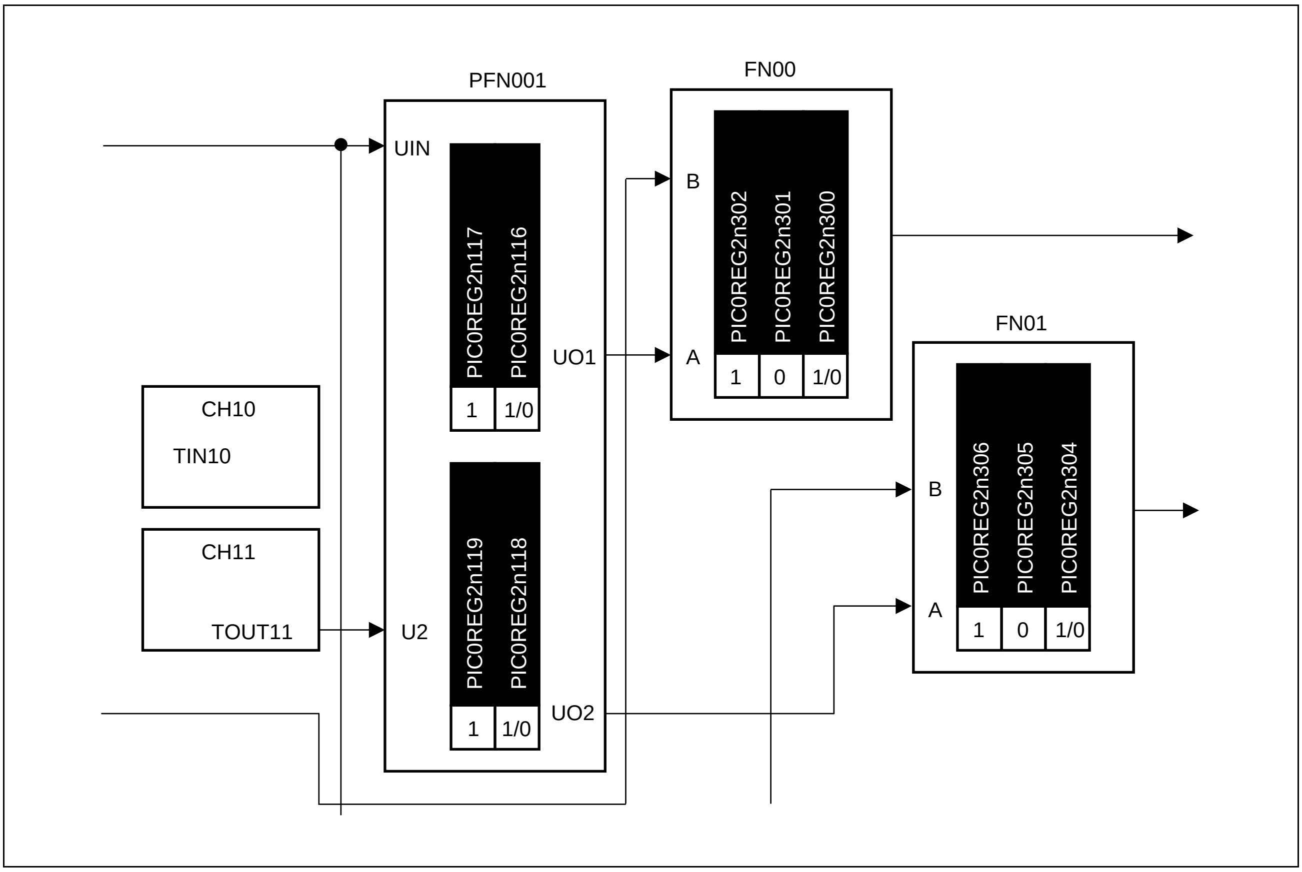


Figure 36.35 Block Diagram Excerpt (PFN001, FN00, and FN01)

The table below shows the relationships between combination circuit input (UIN, U2) and output (UO1, UO2).

Table 36.59 U and UB Phase Combination Circuit (PFN001) I/O Table

- UO1 (U phase variable dead time area pulse) output

UIN (TOUT02)	U2 (TOUT11)	UO1	
		PIC0REG2n117, 16 = 10_B U phase output active high (TAUDnTOL04 = 0)	PIC0REG2n117, 16 = 11_B U phase output active low (TAUDnTOL04 = 1)
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	0

- UO2 (UB phase variable dead time area pulse) output

UIN (TOUT02)	U2 (TOUT11)	UO2	
		PIC0REG2n119, 18 = 10_B UB phase output active high (TAUDnTOL05 = 0)	PIC0REG2n119, 18 = 11_B UB phase output active low (TAUDnTOL05 = 1)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NOTE

The PIC0REG2n116, PIC0REG2n117, PIC0REG2n118, and PIC0REG2n119 settings change depending on the active U phase and UB phase levels of the generated triangle PWM signal with dead time.

36.12.4.4 Logical Operation Circuit (FN0i) (i = 0 or 1)

This circuit combines an output triangle PWM signal with dead time (TOUT04, TOUT05) with combination circuit output (UO1 and UO2 of PFN001) and generates a PWM signal to which a variable dead time area pulse has been added.

The combination logic for the logical operation circuit is switched according to the PIC0REG2n3 register setting. (Bits 0 to 2 are specified for U phase output, and bits 4 to 6 are specified for UB phase output.)

Set up the logical operation circuit as shown in the table below. The combined signal is output from the TAPAnUP and TAPAnUM pins according to the specified combination logic.

Table 36.60 Logical Operation Circuit (FN0i) (i = 0 or 1) Settings and TAPAnUP and TAPAnUM Pin Output

- U phase output (TOUT04)

Active Level	PIC0REG2n302 to 00	TAPAnUP Pin Output Waveform
Active high (TAUDnTOL04 = 0)	100_B	AND of FN00 B (TOUT04) and FN00 A (UO1)
Active low (TAUDnTOL04 = 1)	101_B	OR of FN00 B (TOUT04) and FN00 A (UO1)

- UB phase output (TOUT05)

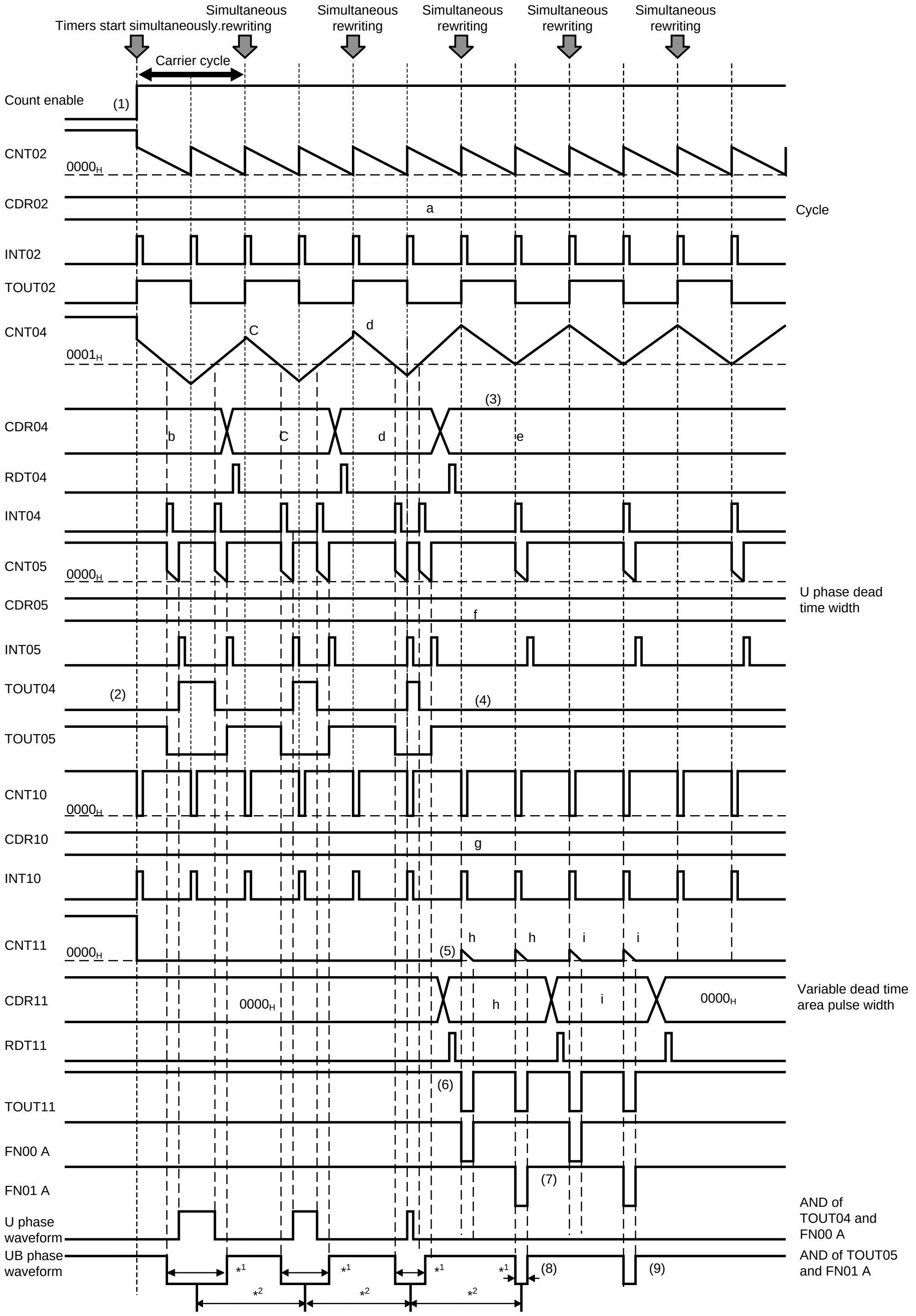
Active Level	PIC0REG2n306 to 04	TAPAnUM Pin Output
Active high (TAUDnTOL05 = 0)	100_B	AND of FN01 B (TOUT05) and FN01 A (UO2)
Active low (TAUDnTOL05 = 1)	101_B	OR of FN01 B (TOUT05) and FN01 A (UO2)

Because the above makes variable dead time control possible to ensure output accuracy near duties of 0% and 100% even for TAUD, a more accurate triangle PWM signal can be output than that output using the TAUD feature for outputting a triangle PWM signal with dead time.

For the V/VB phase and W/WB phase, the used channels and register bits differ, but the settings are the same, as shown in **Figure 36.34, Block Diagram: High-Accuracy Triangle PWM Output with Dead Time.**

The peripheral interconnections provide a connection for adding the pulse generated during one-shot pulse output to the PWM signal generated during output of a triangle PWM signal with dead time by using the combination circuit and logical operation circuit of the peripheral interconnections.

The following figures show timing charts for outputting a high-accuracy triangle PWM signal with dead time.



Note 1. The variable dead time area pulse uses a sawtooth wave and is therefore expanded on one side, unlike a pulse that uses a triangle wave, which is expanded on both sides.

Note 2. Because the variable dead time area pulse is expanded on one side, the length of the one-phase PWM output cycle for the variable dead time area increases by 1/2 the added variable dead time area pulse width.

Figure 36.36 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))

An operation example in which the system transitions to a U phase of 0% and UB phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 36.36, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))** is provided below. Output of a triangle PWM signal with dead time is active high.

(1) When timer operation is started, output of a triangle PWM with dead time is started by the CH2, CH4, and CH5 channels of TAUDn.

(2) A triangle PWM waveform with dead time is generated from TOUT04 and TOUT05.

(3) A U phase duty output value of 0% is specified for CDR04.

(4) Due to the setting in (3), TOUT04 output is the inactive level, and TOUT05 output is the active level. However, no variable dead time area pulse is output during this operation.

(5) To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 when specifying the 0% U phase duty in (3).
For this example, the CDR11 setting is fixed to 0000_H until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.

(6) The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.

(7) The pulse output in (6) is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).

(8) The pulse generated in (7) is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output).

(9) By later changing the CDR11 setting, which specifies the width of the variable dead time area pulse, the desired variable dead time area pulse can be added.

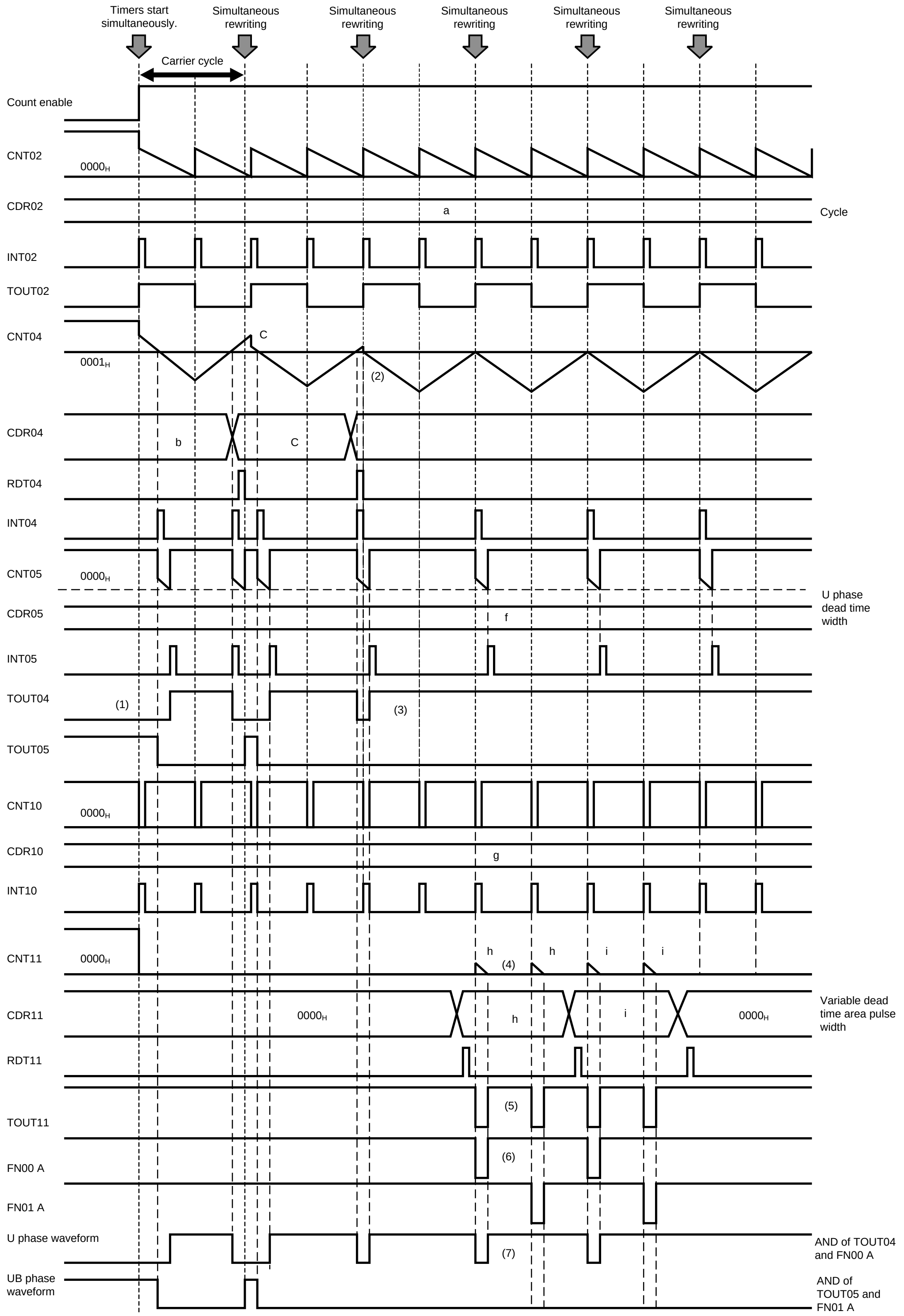


Figure 36.37 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))

An operation example in which the system transitions to a U phase of 100% and UB phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 36.37, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))** is provided below. Output of a triangle PWM signal with dead time is active high.

(1) The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same.

(2) A U phase duty output value of 100% (CDR04 = 0000_H) is specified for CDR04.

(3) Due to the setting in (2), TOUT04 output is the active level, and TOUT05 output is the inactive level. However, no variable dead time area pulse is output during this operation.

(4) To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 one cycle after specifying the 100% U phase duty setting in (2).
For this example, the CDR11 setting is fixed to 0000_H until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.

(5) The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.

(6) The pulse output in (5) is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).

(7) The pulse generated in (6) is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output).

CAUTION

If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the variable dead time area pulse is affected by the amount shown by (2) for the last PWM signal output from TOUT04 and shown by feature specification (1), as shown in **Figure 36.37, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**.
To cancel this effect, the CDR11 setting is delayed one cycle.

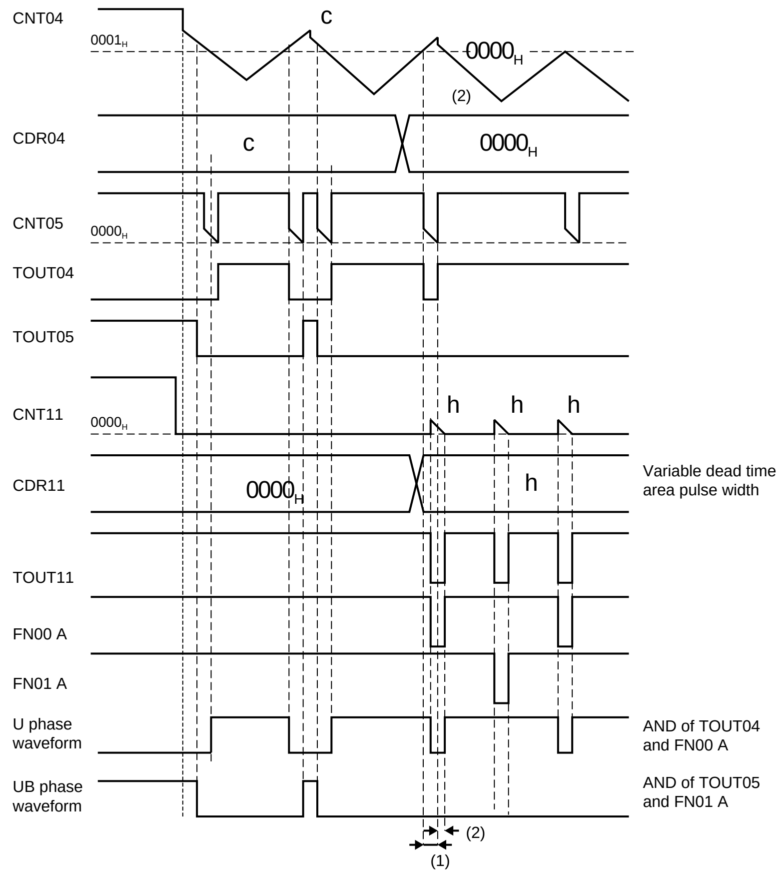


Figure 36.38 Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse

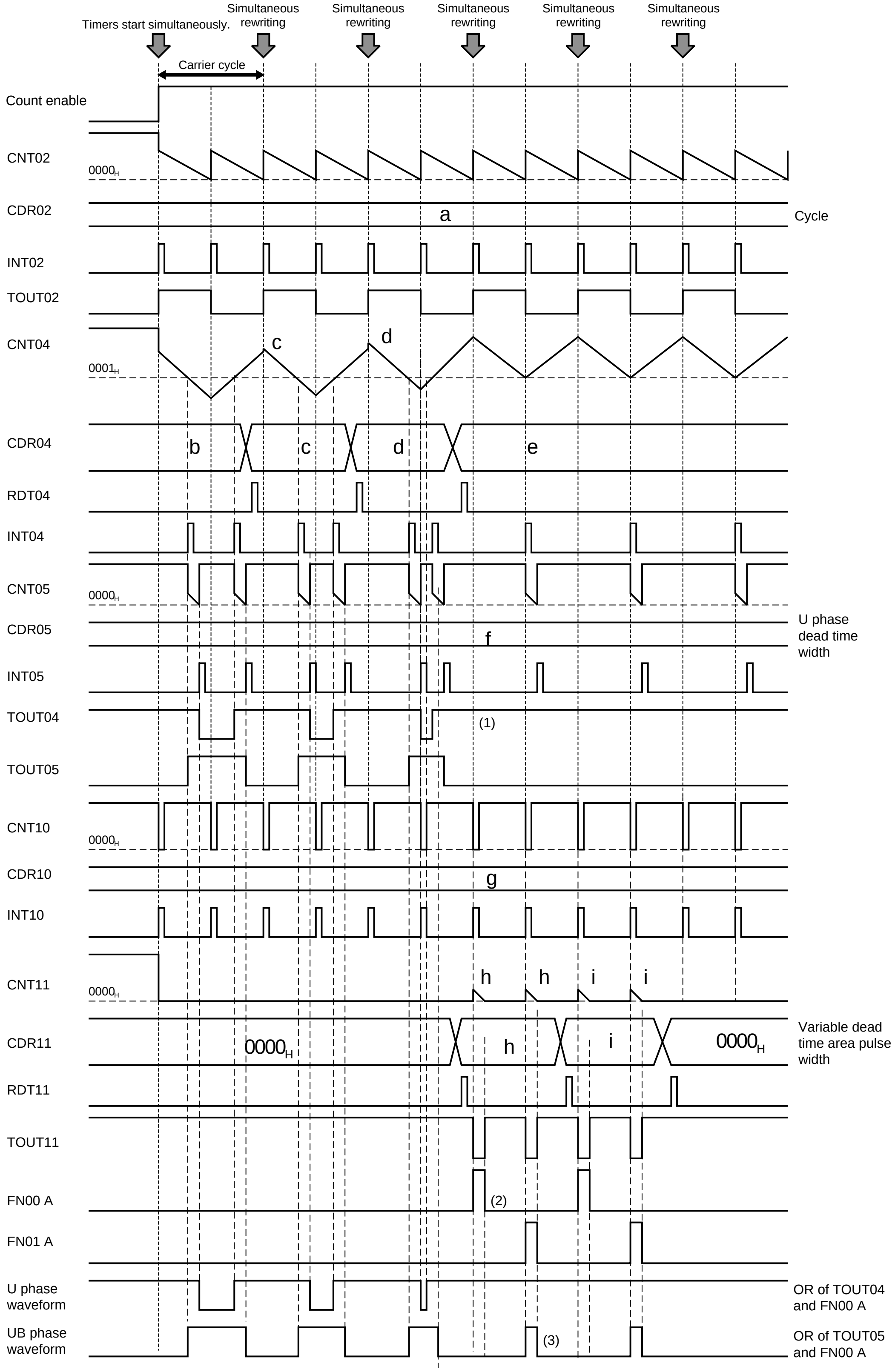


Figure 36.39 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (TAUDnTOL04 = 1 (Active Low), TAUDnTOL05 = 1 (Active Low))

An operation example in which the system transitions to a U phase of 100% and UB phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 36.39, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (TAUDnTOL04 = 1 (Active Low), TAUDnTOL05 = 1 (Active Low))** is provided below. Output of a triangle PWM signal with dead time is active low.

(1) The timer operation from the start of timer operation until the output of a triangle PWM with dead time is the same as in **Figure 36.36, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output from TOUT04 and TOUT05.

(2) Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).

(3) In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in (2) is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output) as an active low PWM signal.

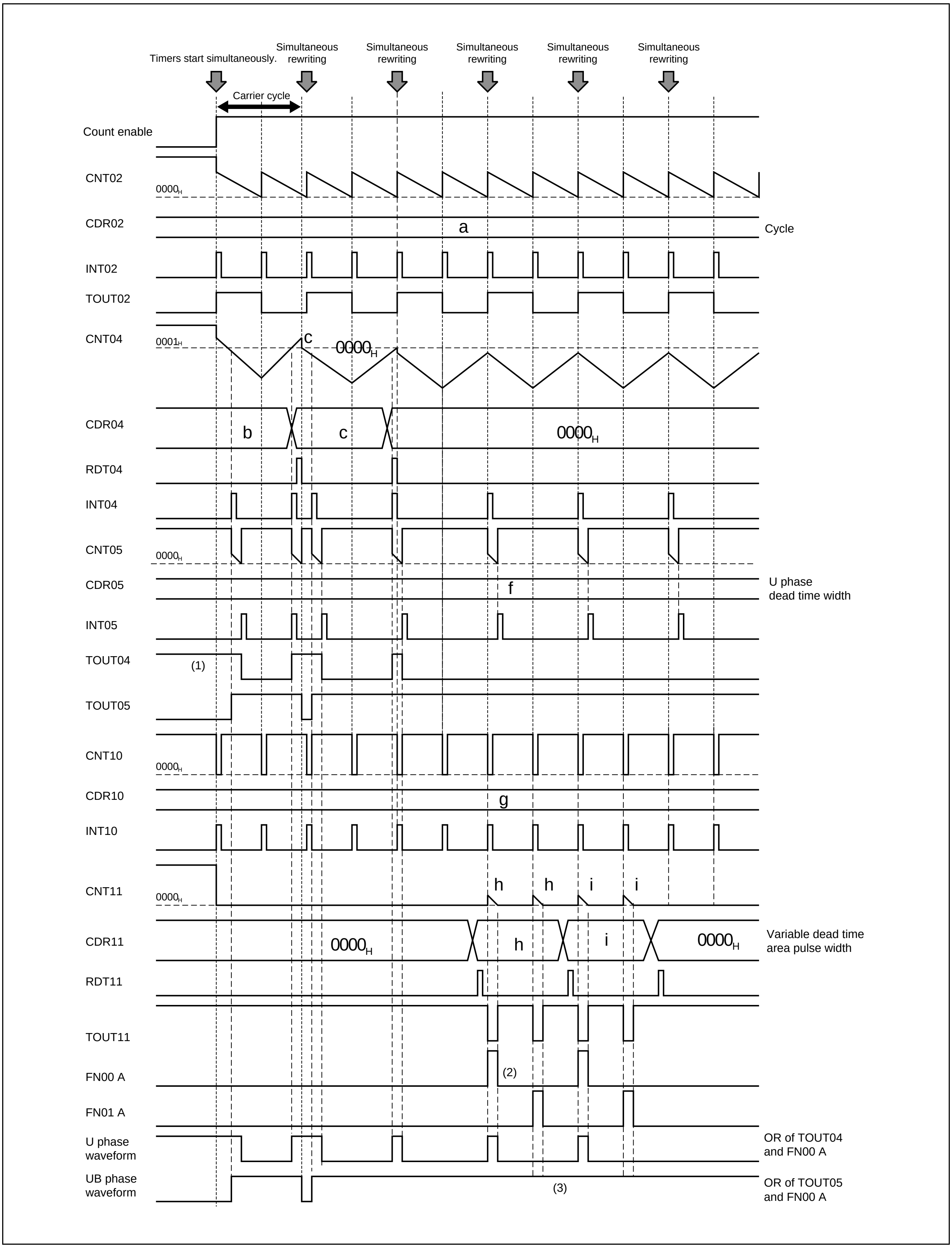


Figure 36.40 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active Low) and TAUDnTOL05 = 0 (Active Low))

An operation example in which the system transitions to a U phase of 0% and UB phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 36.40, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active Low) and TAUDnTOL05 = 0 (Active Low))** is provided below. Output of a triangle PWM signal with dead time is active low.

(1) The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same as in **Figure 36.37, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output.

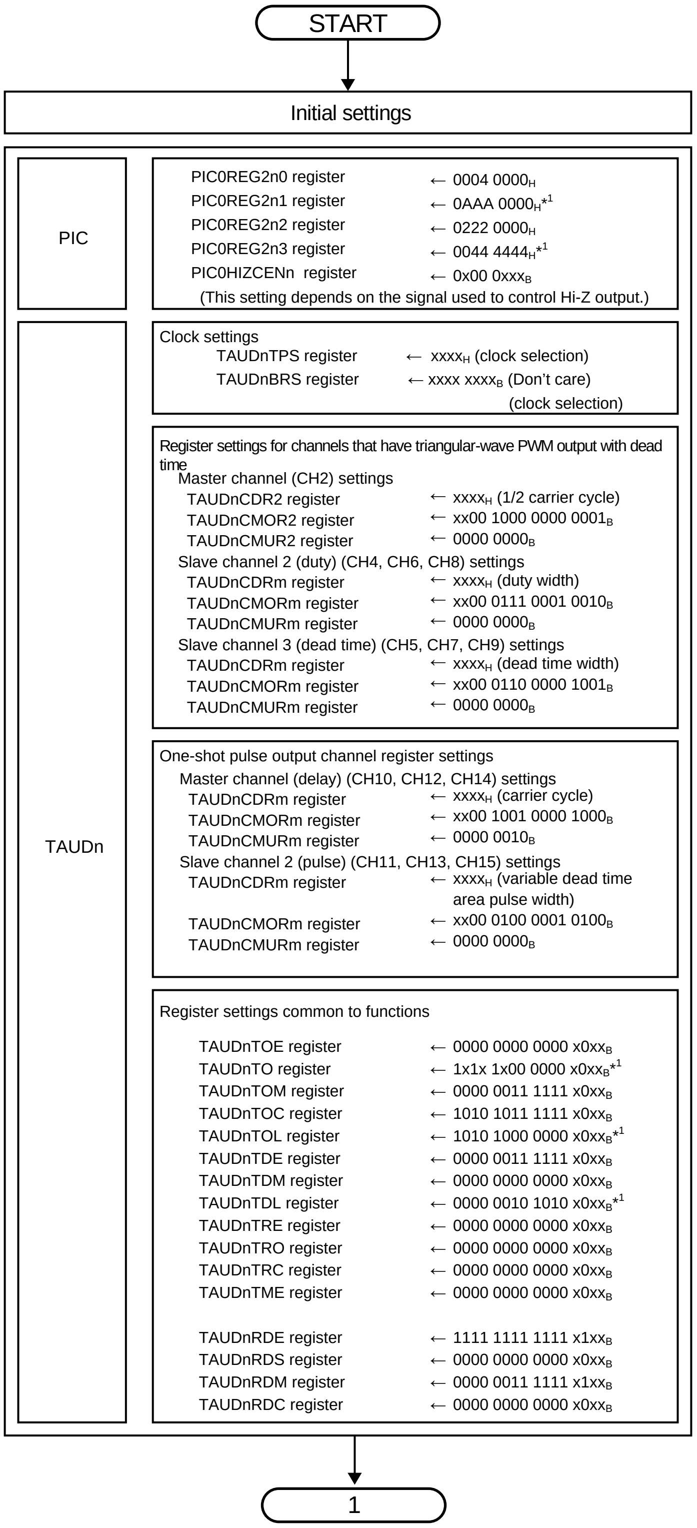
(2) Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).

(3) In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in (2) is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output) as an active low PWM signal.

CAUTION

If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the last PWM signal output from TOUT04 is adversely affected due to the feature specifications.
To cancel this effect, the CDR11 setting is delayed one cycle.
For details, see **Figure 36.38, Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse**.

36.12.5 Setup Flow



PIC0REG2n0 register ← 0004 0000$_H$
PIC0REG2n1 register ← 0AAA 0000$_H$*1
PIC0REG2n2 register ← 0222 0000$_H$
PIC0REG2n3 register ← 0044 4444$_H$*1
PIC0HIZCENn register ← 0x00 0xxx$_B$
(This setting depends on the signal used to control Hi-Z output.)

Clock settings
TAUDnTPS register ← xxxx$_H$ (clock selection)
TAUDnBRS register ← xxxx xxxx$_B$ (Don't care)
(clock selection)

Register settings for channels that have triangular-wave PWM output with dead time
Master channel (CH2) settings
TAUDnCDR2 register ← xxxx$_H$ (1/2 carrier cycle)
TAUDnCMOR2 register ← xx00 1000 0000 0001$_B$
TAUDnCMUR2 register ← 0000 0000$_B$
Slave channel 2 (duty) (CH4, CH6, CH8) settings
TAUDnCDRm register ← xxxx$_H$ (duty width)
TAUDnCMORm register ← xx00 0111 0001 0010$_B$
TAUDnCMURm register ← 0000 0000$_B$
Slave channel 3 (dead time) (CH5, CH7, CH9) settings
TAUDnCDRm register ← xxxx$_H$ (dead time width)
TAUDnCMORm register ← xx00 0110 0000 1001$_B$
TAUDnCMURm register ← 0000 0000$_B$

One-shot pulse output channel register settings
Master channel (delay) (CH10, CH12, CH14) settings
TAUDnCDRm register ← xxxx$_H$ (carrier cycle)
TAUDnCMORm register ← xx00 1001 0000 1000$_B$
TAUDnCMURm register ← 0000 0010$_B$
Slave channel 2 (pulse) (CH11, CH13, CH15) settings
TAUDnCDRm register ← xxxx$_H$ (variable dead time area pulse width)
TAUDnCMORm register ← xx00 0100 0001 0100$_B$
TAUDnCMURm register ← 0000 0000$_B$

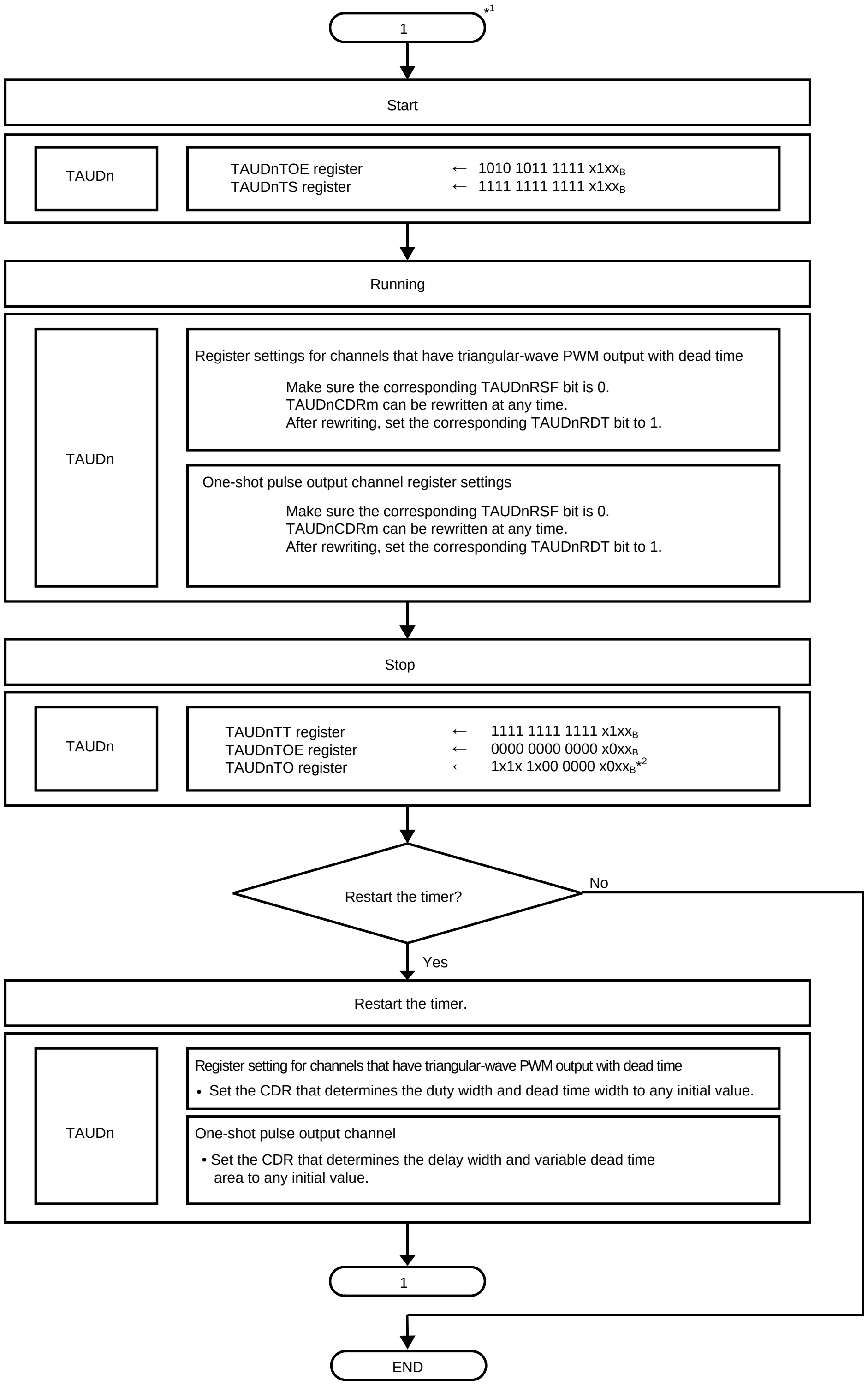
Register settings common to functions

TAUDnTOE register ← 0000 0000 0000 x0xx$_B$
TAUDnTO register ← 1x1x 1x00 0000 x0xx$_B$*1
TAUDnTOM register ← 0000 0011 1111 x0xx$_B$
TAUDnTOC register ← 1010 1011 1111 x0xx$_B$
TAUDnTOL register ← 1010 1000 0000 x0xx$_B$*1
TAUDnTDE register ← 0000 0011 1111 x0xx$_B$
TAUDnTDM register ← 0000 0000 0000 x0xx$_B$
TAUDnTDL register ← 0000 0010 1010 x0xx$_B$*1
TAUDnTRE register ← 0000 0000 0000 x0xx$_B$
TAUDnTRO register ← 0000 0000 0000 x0xx$_B$
TAUDnTRC register ← 0000 0000 0000 x0xx$_B$
TAUDnTME register ← 0000 0000 0000 x0xx$_B$

TAUDnRDE register ← 1111 1111 1111 x1xx$_B$
TAUDnRDS register ← 0000 0000 0000 x0xx$_B$
TAUDnRDM register ← 0000 0011 1111 x1xx$_B$
TAUDnRDC register ← 0000 0000 0000 x0xx$_B$

Note 1. Change settings according to the active level of the PWM signal to be output.

Figure 36.41 Setup Flow (Active High Example)



Note 1. Specify the selection register and output port to use after specifying the initial settings for the peripheral interconnections and timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 36.42 Setup Flow (Active High Example) (continued)

36.12.6 Setting Examples for Operation Functions

This section provides example settings for each register.

36.12.6.1 TAUDn Settings (active high example)

Table 36.61 TAUDn: CH2-related (Master Channel Used To Output A Triangle PWM Signal with Dead Time*[1])

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	At the start of operation, output INTm and toggle TOUTm.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	Fixed

Note 1. The master channel and slave channel names are defined for TAUD triangle PWM output with dead time. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register of the master channel used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.
For this feature, set TAUDnMD0 to 1.

Table 36.62 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Triangle PWM Signal with Dead Time*[1]) (m = 4, 6, or 8)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	111	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1001	
	0	TAUDnMD0	0	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 36.63 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Triangle PWM Signal with Dead Time*[1]) (m = 5, 7, or 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	110	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 36.64 TAUDn: CH10, CH12, and CH14-related (Master Channel used to Output a One-shot Pulse*[1]) (m = 10, 12, or 14)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	10	Detect both rising and falling edges as valid.

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 36.65 TAUDn: CH11, CH13, and CH15-related (Slave Channel used to Output a One-Shot Pulse*[1]) (m = 11, 13, or 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don’t care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel. Specify the same clock setting as for the master channel (CH2) used to output a triangle PWM signal with dead time.

NOTE

For the TAUDnCMORm register used during one-shot pulse output, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.
For this feature clear TAUDnMD0 to 0.

Table 36.66 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15	TAUDnTOE15	0 1	Disable the timer. Enable the timer.
	14	TAUDnTOE14	0	
	13	TAUDnTOE13	0 1	Disable the timer. Enable the timer.
	12	TAUDnTOE12	0	
	11	TAUDnTOE11	0 1	Disable the timer. Enable the timer.
	10	TAUDnTOE10	0	
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0 1	Disable the timer. Enable the timer.
	3	TAUDnTOE03	Don’t care	
	2	TAUDnTOE02	0 1	Disable the timer. Enable the timer.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don’t care	

Table 36.66 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTO	15	TAUDnTO15	1*1	Output a high-level signal to TOUT15.
	14	TAUDnTO14	Don’t care	
	13	TAUDnTO13	1*1	Output a high-level signal to TOUT13.
	12	TAUDnTO12	Don’t care	
	11	TAUDnTO11	1*1	Output a high-level signal to TOUT11.
	10	TAUDnTO10	Don’t care	
	9 to 4	TAUDnTO09 to TAUDnTO04	0*1	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don’t care	
	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don’t care	
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	0	Independent operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don’t care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don’t care	
TAUDnTOC	15	TAUDnTOC15	1	Operation mode 2
	14	TAUDnTOC14	0	Operation mode 1
	13	TAUDnTOC13	1	Operation mode 2
	12	TAUDnTOC12	0	Operation mode 1
	11	TAUDnTOC11	1	Operation mode 2
	10	TAUDnTOC10	0	Operation mode 1
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1	Operation mode 2
	3	TAUDnTOC03	Don’t care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don’t care	
TAUDnTOL	15	TAUDnTOL15	1*1	Inverted logic output (active low)
	14	TAUDnTOL14	Don’t care	
	13	TAUDnTOL13	1*1	Inverted logic output (active low)
	12	TAUDnTOL12	Don’t care	
	11	TAUDnTOL11	1*1	Inverted logic output (active low)
	10	TAUDnTOL10	Don’t care	
	9 to 4	TAUDnTOL09 to TAUDnTOL04	0*1	Positive logic output (active high)
	3	TAUDnTOL03	Don’t care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don’t care	

Table 36.66 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	0	Disable dead time control.
	9 to 4	TAUDnTDE09 to TAUDnTDE04	1	Enable dead time control.[*2]
	3	TAUDnTDE03	Don’t care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don’t care	
TAUDnTDM	15 to 9	TAUDnTDM15 to TAUDnTDM09	0	
	3	TAUDnTDM03	Don’t care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don’t care	
TAUDnTDL	15 to 10	TAUDnTDL15 to TAUDnTDL10	0	Invalid because dead time control is disabled.
	9	TAUDnTDL09	1[*1]	Dead time is in the negative segment of the W phase output
	8	TAUDnTDL08	0[*1]	Dead time is in the positive segment of the W phase output
	7	TAUDnTDL07	1[*1]	Dead time is in the negative segment of the V phase output
	6	TAUDnTDL06	0[*1]	Dead time is in the positive segment of the V phase output
	5	TAUDnTDL05	1[*1]	Dead time is in the negative segment of the U phase output
	4	TAUDnTDL04	0[*1]	Dead time is in the positive segment of the U phase output
	3	TAUDnTDL03	Don’t care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don’t care	
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Disable real-time output.
	3	TAUDnTRE03	Don’t care	
	2	TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don’t care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don’t care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don’t care	

Table 36.66 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't care	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't care	
TAUDnRDE	15 to 4	TAUDnRDE15 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't care	
TAUDnRDS	15 to 4	TAUDnRDS15 to TAUDnRDS04	0	Do not enable simultaneous rewriting by using another upper channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Do not enable simultaneous rewriting by using another upper channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't care	
TAUDnRDM	15 to 10	TAUDnRDM15 to TAUDnRDM10	0	Perform simultaneous rewriting when the master channel starts counting.
	9 to 4	TAUDnRDM09 to TAUDnRDM04	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	3	TAUDnRDM03	Don't care	
	2	TAUDnRDM02	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't care	
TAUDnRDC	15 to 4	TAUDnRDC15 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't care	
	2	TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 32, Timer Array Unit D (TAUD)**.

36.12.6.2 PIC Settings (Active High Example)

Table 36.67 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n0	18	PIC0REG2n018	1	Select the TOUT signal of CH2 of TAUDn.
PIC0REG2n1	27, 26	PIC0REG2n127 PIC0REG2n126	1 0	Negative W phase active high combination circuit output
	25, 24	PIC0REG2n125 PIC0REG2n124	1 0	Positive W phase active high combination circuit output
	23, 22	PIC0REG2n123 PIC0REG2n122	1 0	Negative V phase active high combination circuit output
	21, 20	PIC0REG2n121 PIC0REG2n120	1 0	Positive V phase active high combination circuit output
	19, 18	PIC0REG2n119 PIC0REG2n118	1 0	Negative U phase active high combination circuit output
	17, 16	PIC0REG2n117 PIC0REG2n116	1 0	Positive U phase active high combination circuit output
PIC0REG2n2	25	PIC0REG2n225	1	Select the input selected by the PIC0REG2n018 bit.
	21	PIC0REG2n221	1	Select the input selected by the PIC0REG2n018 bit.
	17	PIC0REG2n217	1	Select the input selected by the PIC0REG2n018 bit.
PIC0REG2n3	22, 21, 20	PIC0REG2n322 PIC0REG2n321 PIC0REG2n320	1 0 0	Negative W phase active high logical operation circuit output
	18, 17, 16	PIC0REG2n318 PIC0REG2n317 PIC0REG2n316	1 0 0	Positive W phase active high logical operation circuit output
	14, 13, 12	PIC0REG2n314 PIC0REG2n313 PIC0REG2n312	1 0 0	Negative V phase active high logical operation circuit output
	10, 9, 8	PIC0REG2n310 PIC0REG2n309 PIC0REG2n308	1 0 0	Positive V phase active high logical operation circuit output
	6, 5, 4	PIC0REG2n306 PIC0REG2n305 PIC0REG2n304	1 0 0	Negative U phase active high logical operation circuit output
	2, 1, 0	PIC0REG2n302 PIC0REG2n301 PIC0REG2n300	1 0 0	Positive U phase active high logical operation circuit output

36.13 Delay Pulse Output with Dead Time

36.13.1 Functional Overview

This feature outputs a three-phase PWM with dead time that is later than the cycle timing by an amount equal to the delay amount.

Unlike the function of three-phase PWM output with dead time, a PWM signal that has a reset in the next cycle can be output.

36.13.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0, m = 0 to 15)

Table 36.68 Configuration of Delay Pulse Output with Dead Time

Timer	Timer Motor Control Function
TAUD0 CH2 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

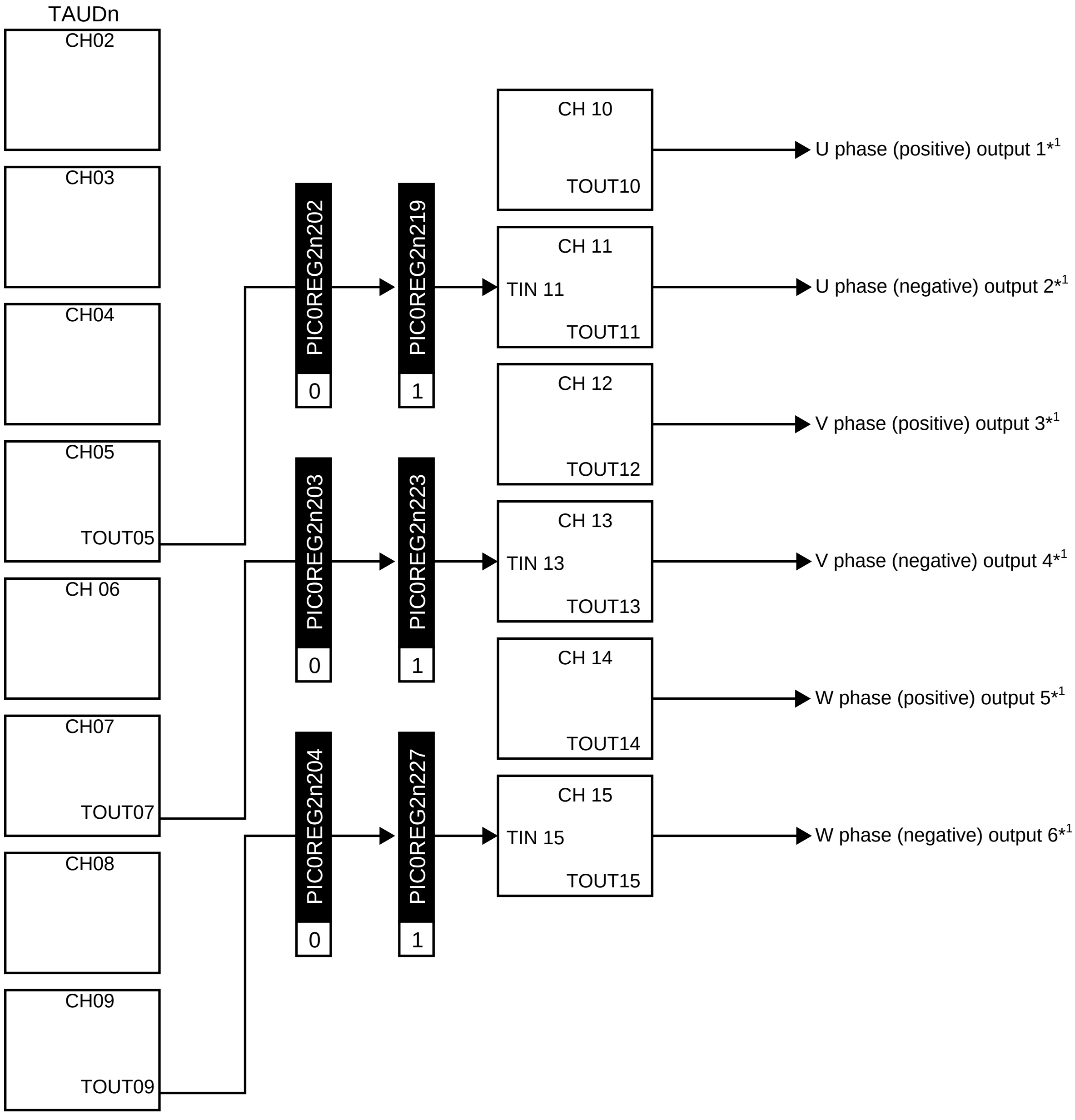
36.13.2.1 TAUDn Configuration

Because the CDRm value of CH3 does not affect TOUT0 to TOUT15, the INTm signal of CH3 can also be used for other purposes such as A/D conversion trigger generation.

Table 36.69 TAUDn configuration

CH	Function Name	M/S*[1]	CDR Setting	Description
2	Delay pulse output function (CH2 is the master channel for CH3 to CH9.)	M	Cycle	
3		S		Reserved
4		S	Delay (U phase)	
5		S	Pulse width (U phase)	
6		S	Delay (V phase)	
7		S	Pulse width (V phase)	
8		S	Delay (W phase)	
9		S	Pulse width (W phase)	
10	Any feature that does not use TOUTm	S		TOUT: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT: UB phase output
12	Any feature that does not use TOUTm	S		TOUT: V phase output
13	One-phase PWM output	S	Dead time (V phase)	TOUT: VB phase output
14	Any feature that does not use TOUTm	S		TOUT: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT: WB phase output

Note 1. M: Master channel, S: Slave channel



Note 1. For the connection destination, see **Figure 36.25, Block Diagram: Motor Output Buffer Control**.

Figure 36.43 Block Diagram: Delay Pulse Output with Dead Time

36.13.3 Registers

36.13.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

This register selects CHm input signals of the TAUDn timer. This section describes the bits to be used in the delay pulse output with dead time.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n227	—	—	—	PIC0REG2n223	—	—	—	PIC0REG2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 36.70 PIC0REG2n2 Register Contents

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31 to 28</td><td>Reserved</td><td>*1</td></tr>
<tr><td>27</td><td>PIC0REG2n227</td><td>Select the TIN input signal to TAUDTTIN15.
<table>
<tr><th>PIC0REG2n227</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n204 bit</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>26 to 24</td><td>Reserved</td><td>*1</td></tr>
<tr><td>23</td><td>PIC0REG2n223</td><td>Select the TIN input signal to TAUDTTIN13.
<table>
<tr><th>PIC0REG2n223</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n203 bit</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>22 to 20</td><td>Reserved</td><td>*1</td></tr>
<tr><td>19</td><td>PIC0REG2n219</td><td>Select the TIN input signal to TAUDTTIN11.
<table>
<tr><th>PIC0REG2n219</th><th>Input Signal</th></tr>
<tr><td>1</td><td>Signal selected by the PIC0REG2n202 bit</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>18 to 5</td><td>Reserved</td><td>*1</td></tr>
<tr><td>4</td><td>PIC0REG2n204</td><td>Select the signal supplied to TAUDTTIN15.
0: Select TAUDTTOUT9.
1: Setting prohibited</td></tr>
<tr><td>3</td><td>PIC0REG2n203</td><td>Select the signal supplied to TAUDTTIN13.
0: Select TAUDTTOUT7.
1: Setting prohibited</td></tr>
</table>

Table 36.70 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
2	PIC0REG2n202	Select the signal supplied to TAUDTTIN11. 0: Select TAUDTTOUT5. 1: Setting prohibited
1, 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

36.13.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: PIC0HIZCEN0: FFDD 0080$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 36.71 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	RH850/F1KH-D8 Selects whether to enable or disable Hi-Z output control by the WDTA1NMI and the WDTA2NMI interrupt signal. 0: Disable 1: Enable RH850/F1KM-S4, RH850/F1KM-S1 Select whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

36.13.4 Operation Example

This is achieved by combining the following TAUD features:

- Delay pulse output function
- One-phase PWM output

The delay pulse output feature generates a PWM signal that is later than the cycle timing by an amount equal to the delay amount. Next, a one-phase PWM signal to which dead time has been added is output for the delayed PWM signal by one-phase PWM output feature.

A delay pulse with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference between phases is the assigned channel, only one phase (the U phase) is described below.

36.13.4.1 Delay Pulse Output Function

By using a combination of CH2, CH4, and CH5, a basic PWM signal for one-phase PWM is output from TOUT05 delayed by the amount specified by CH4 with respect to the cycle specified by CH2.

Note that CH3 is a reserved timer for achieving this feature, so do not use it for other features.

CAUTION

Do not specify a delay amount that exceeds the cycle.

36.13.4.2 One-phase PWM Output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM with dead time, while the W phase uses CH14 and CH15.

CAUTION

Specify the same clock for each TAUDn channel that uses the delay pulse output and one-phase PWM output features.

For details about the TAUD functions, see **Section 32, Timer Array Unit D (TAUD)**.

The differences between the delay pulse output with dead time and the three-phase PWM output with dead time are described below.

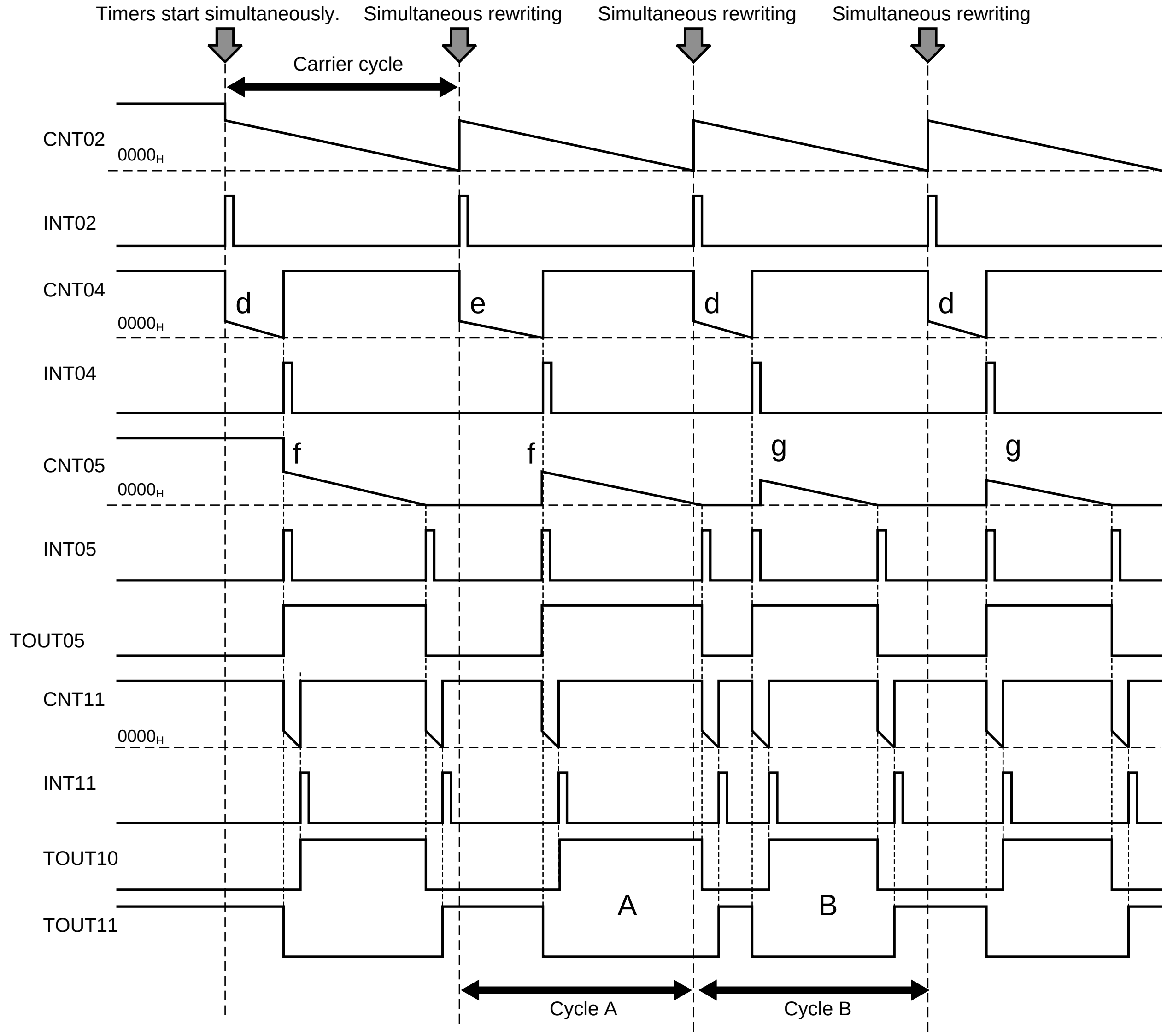


Figure 36.44 PWM Output by Outputting a Delay Pulse with Dead Time

In **Figure 36.44, PWM Output by Outputting a Delay Pulse with Dead Time**, PWM waveform A is supposed to be output before cycle A ends, but because the delay timing is too long, the PWM clear position is after the end of cycle A. Next, PWM waveform B, which is for cycle B, is output.

The operations shown below occur when an attempt is made to achieve the operations shown in **Figure 36.44, PWM Output by Outputting a Delay Pulse with Dead Time** by the three-phase PWM output with dead time.

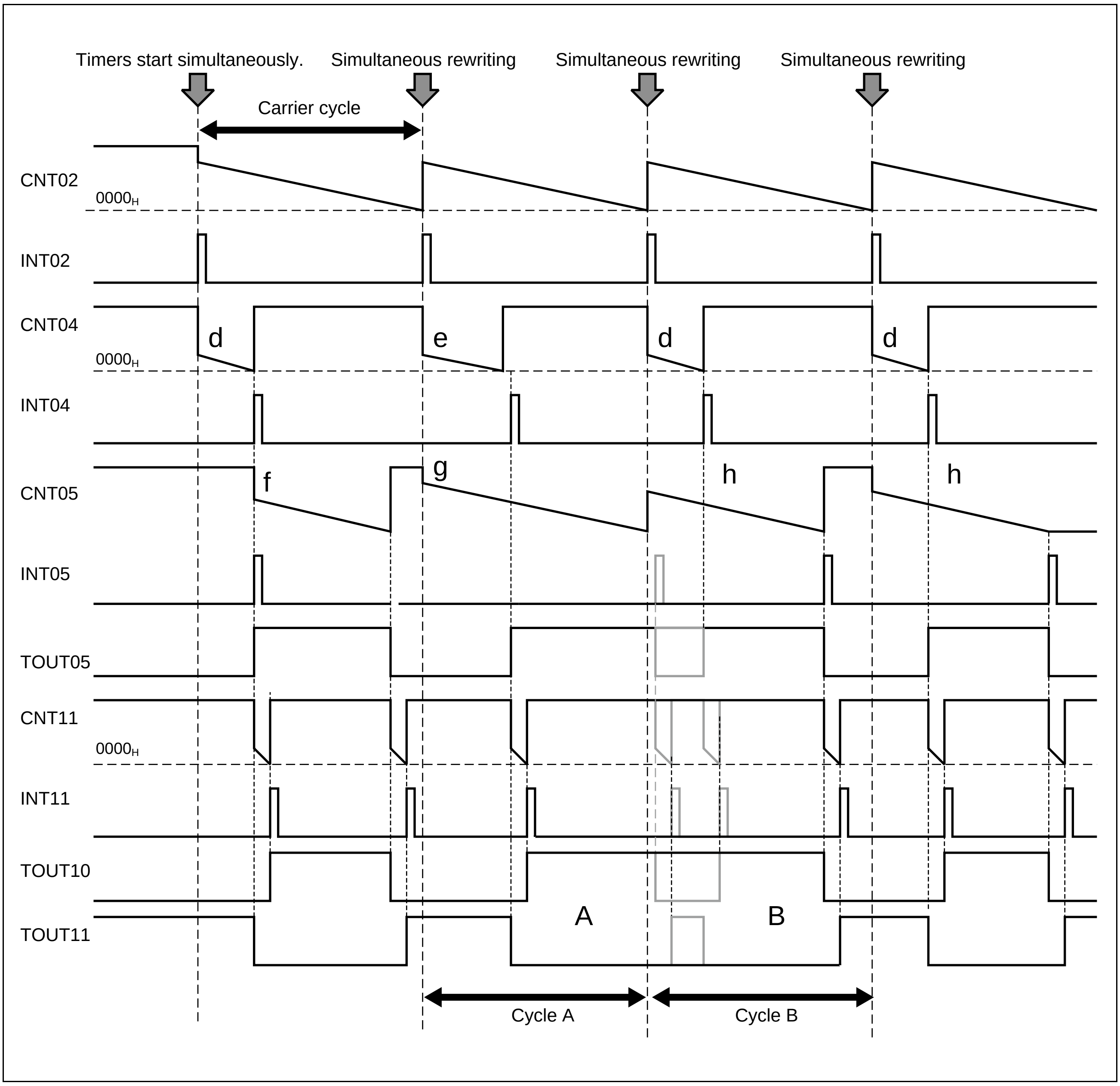


Figure 36.45 Output of a Three-Phase PWM Signal with Dead Time (1)

Figure 36.45, Output of a Three-Phase PWM Signal with Dead Time (1) shows an example in which the output PWM signal does not end before carrier cycle A because the set timing for outputting a three-phase PWM signal with dead time is delayed and the clear timing is after the end of the carrier cycle.

For cycle A, the set timing of PWM waveform A is the same as that in the figure on the previous page, but because the clear timing is after the end of cycle A, a reload operation occurs in cycle A before PWM waveform A is cleared, and the clear timing for PWM waveform A does not occur.

In addition, the set timing of PWM waveform B for cycle B is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle B, and a waveform that combines PWM waveform A and PWM waveform B is output.

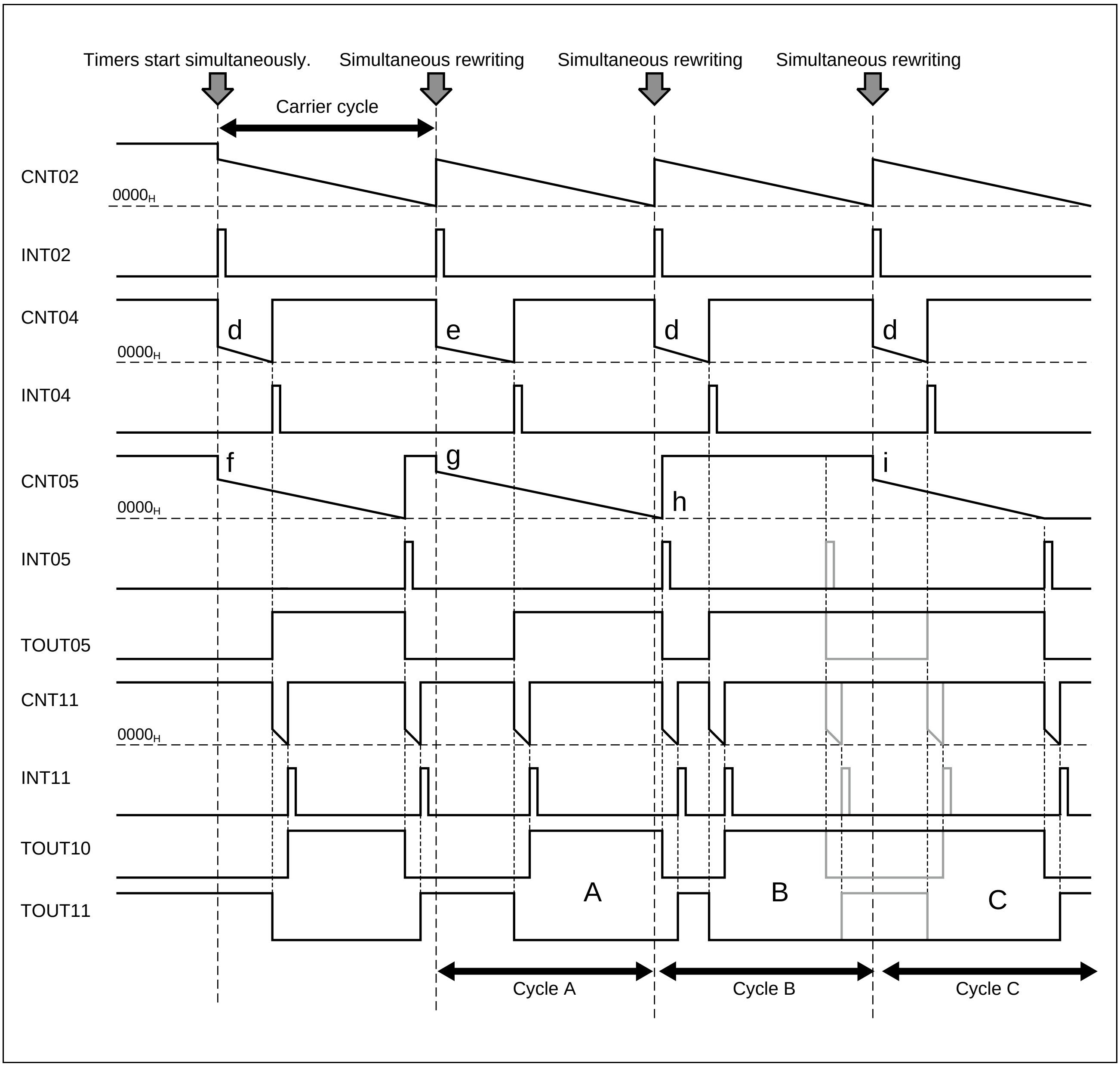


Figure 36.46 Output of a Three-Phase PWM Signal with Dead Time (2)

Figure 36.46, Output of a Three-Phase PWM Signal with Dead Time (2) shows an example of outputting a three-phase PWM signal with dead time in which counter operation for which the clear timing is longer than cycle A is continued in cycle B, and PWM output A is cleared at the beginning of cycle B.

The output of PWM waveform A for cycle A is the same as the output of a delay pulse with dead time, but because the clear timing is used at the beginning of cycle B, the clear timing of PWM output B, which is supposed to be output during cycle B, does not occur.

In addition, the set timing of PWM waveform C for cycle C is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle C, and a waveform that combines PWM waveform B and PWM waveform C is output.

In this way, it is possible to achieve freer PWM output timing when outputting a delay pulse with dead time than when outputting a three-phase PWM signal with dead time.

The peripheral interconnections provide a connection for using the PWM output timing of delay pulse output as input for one-phase PWM output.

Figure 36.47, Output of a Delay Pulse with Dead Time shows a timing chart for outputting a delay pulse with dead time.

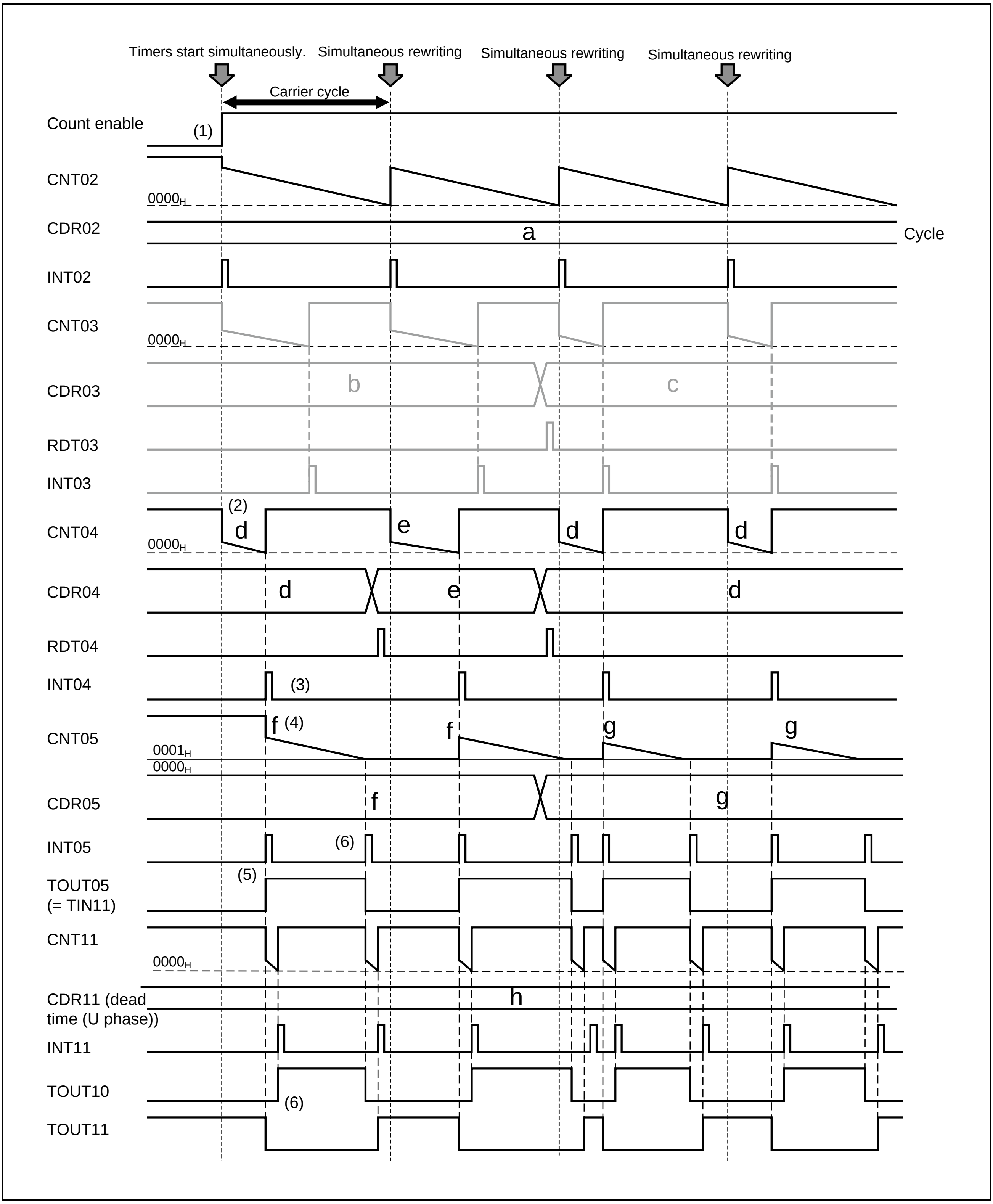


Figure 36.47 Output of a Delay Pulse with Dead Time

The output of a delay pulse with dead time shown in **Figure 36.47, Output of a Delay Pulse with Dead Time** is described below.

(1) CH2 (the carrier cycle timer) and CH4 (the delay timing timer) are started simultaneously by starting timers simultaneously.
CH5 (the PWM duty timer) and CH11 (the dead time timer) are also enabled, but no counting operations are performed until the edges of INT04, which indicates the count start timing for CH5, and TIN11, which indicates the count start timing for CH11, are detected.
Because CH3 does not affect PWM output for this function, the channel is not described.

(2) For CH4, when there is a CH2 underflow, the settings from CDR04 are reloaded to CNT04.

(3) The CH4 underflow generates the delay timing signal (INT04).

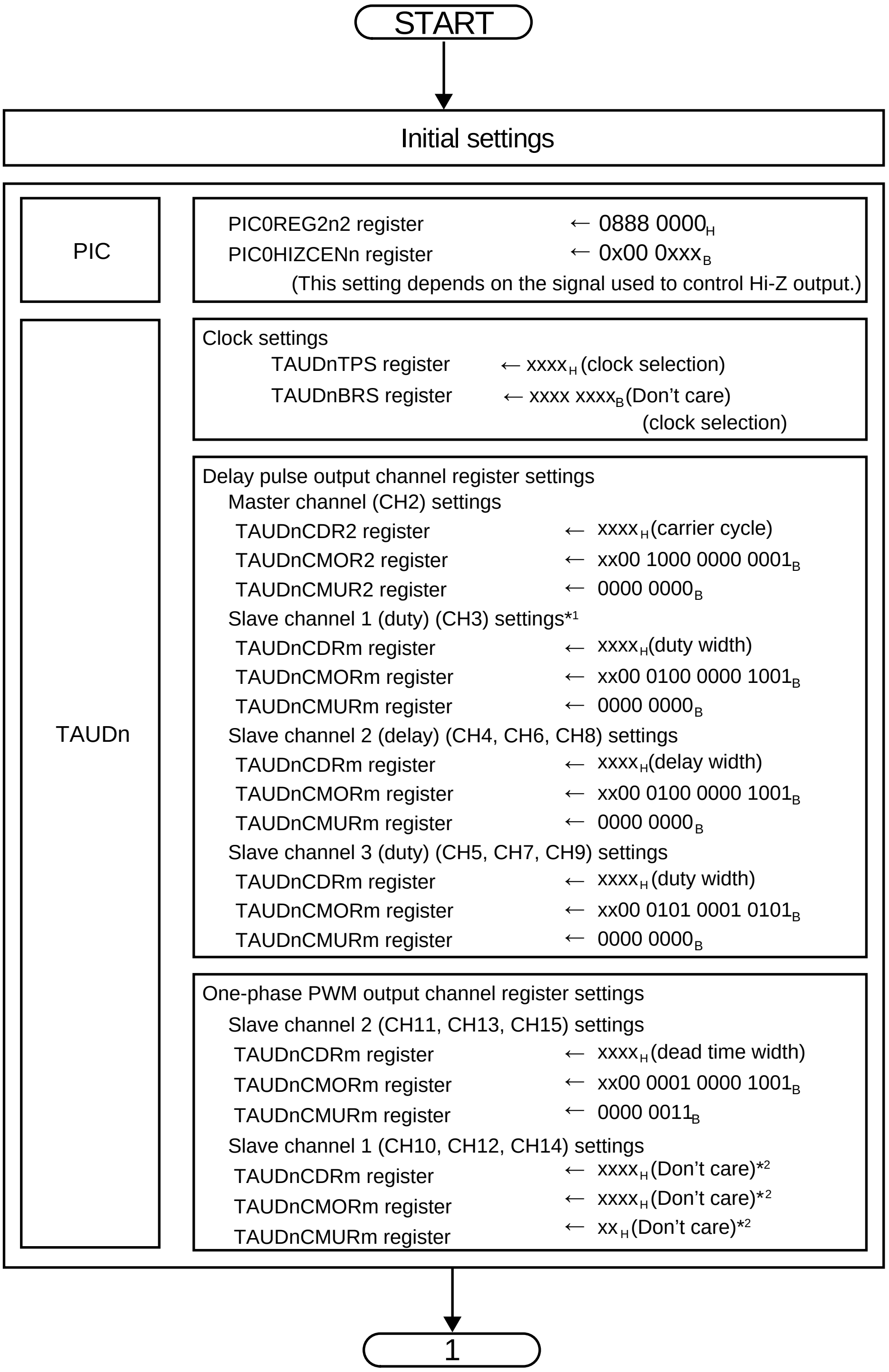
(4) When INT04 is generated, the settings from CDR05 are reloaded to CNT05, and then the CH5 (the PWM duty timer) operation starts.

(5) At this time, INT05 is generated and the TOUT05 output level changes to the active level.

(6) Due to the CH5 underflow, INT05 is generated again, and TOUT05 changes to the inactive level. TOUT05, which is changed by the CH4 and CH5 underflow, is supplied to the TIN11 input of one-phase PWM output.

(7) During one-phase PWM output, a PWM waveform with dead time is output by detecting a TIN11 edge.

36.13.5 Setup Flow



PIC0REG2n2 register ← $0888\ 0000_H$
PIC0HIZCENn register ← $0x00\ 0xxx_B$
(This setting depends on the signal used to control Hi-Z output.)

Clock settings
TAUDnTPS register ← $xxxx_H$ (clock selection)
TAUDnBRS register ← $xxxx\ xxxx_B$ (Don’t care)
(clock selection)

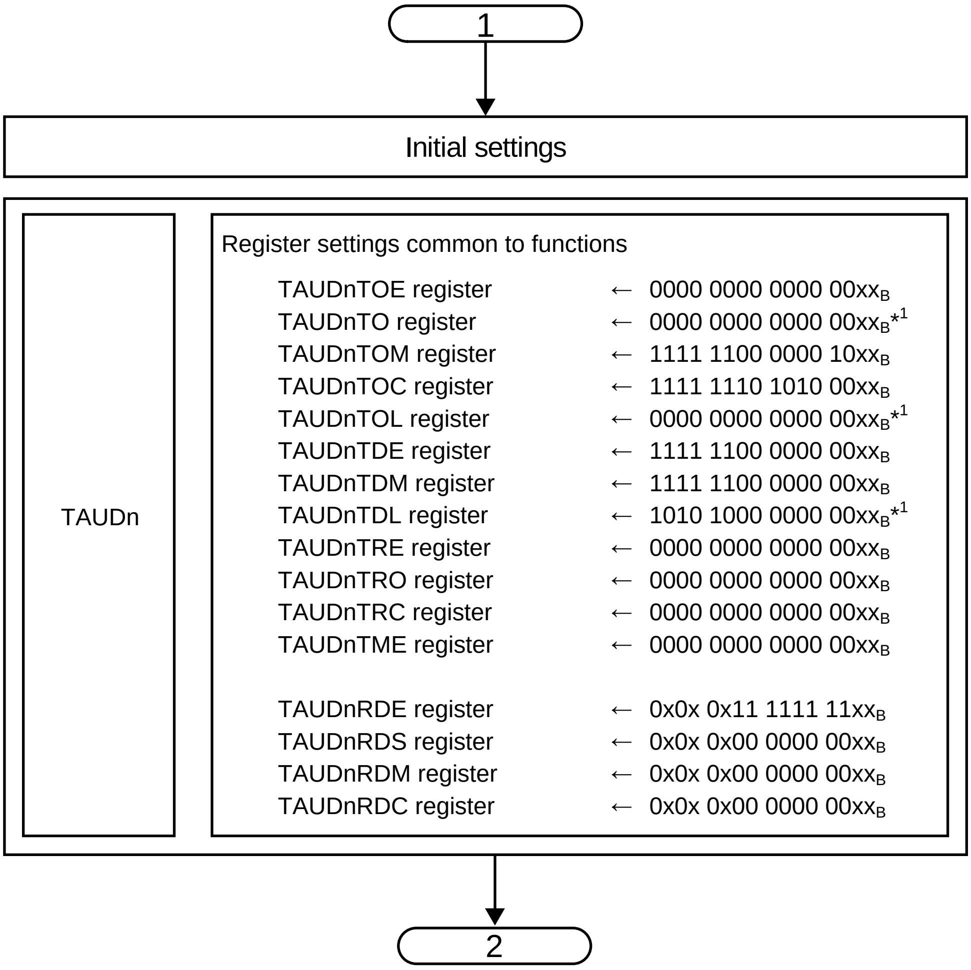
Delay pulse output channel register settings
Master channel (CH2) settings
TAUDnCDR2 register ← $xxxx_H$ (carrier cycle)
TAUDnCMOR2 register ← $xx00\ 1000\ 0000\ 0001_B$
TAUDnCMUR2 register ← $0000\ 0000_B$
Slave channel 1 (duty) (CH3) settings*1
TAUDnCDRm register ← $xxxx_H$ (duty width)
TAUDnCMORm register ← $xx00\ 0100\ 0000\ 1001_B$
TAUDnCMURm register ← $0000\ 0000_B$
Slave channel 2 (delay) (CH4, CH6, CH8) settings
TAUDnCDRm register ← $xxxx_H$ (delay width)
TAUDnCMORm register ← $xx00\ 0100\ 0000\ 1001_B$
TAUDnCMURm register ← $0000\ 0000_B$
Slave channel 3 (duty) (CH5, CH7, CH9) settings
TAUDnCDRm register ← $xxxx_H$ (duty width)
TAUDnCMORm register ← $xx00\ 0101\ 0001\ 0101_B$
TAUDnCMURm register ← $0000\ 0000_B$

One-phase PWM output channel register settings
Slave channel 2 (CH11, CH13, CH15) settings
TAUDnCDRm register ← $xxxx_H$ (dead time width)
TAUDnCMORm register ← $xx00\ 0001\ 0000\ 1001_B$
TAUDnCMURm register ← $0000\ 0011_B$
Slave channel 1 (CH10, CH12, CH14) settings
TAUDnCDRm register ← $xxxx_H$ (Don’t care)*2
TAUDnCMORm register ← $xxxx_H$ (Don’t care)*2
TAUDnCMURm register ← xx_H (Don’t care)*2

Note 1. For this feature, the slave channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

Note 2. Specify a feature that does not use TOUTm.

Figure 36.48 Setup Flow (Active High Example)



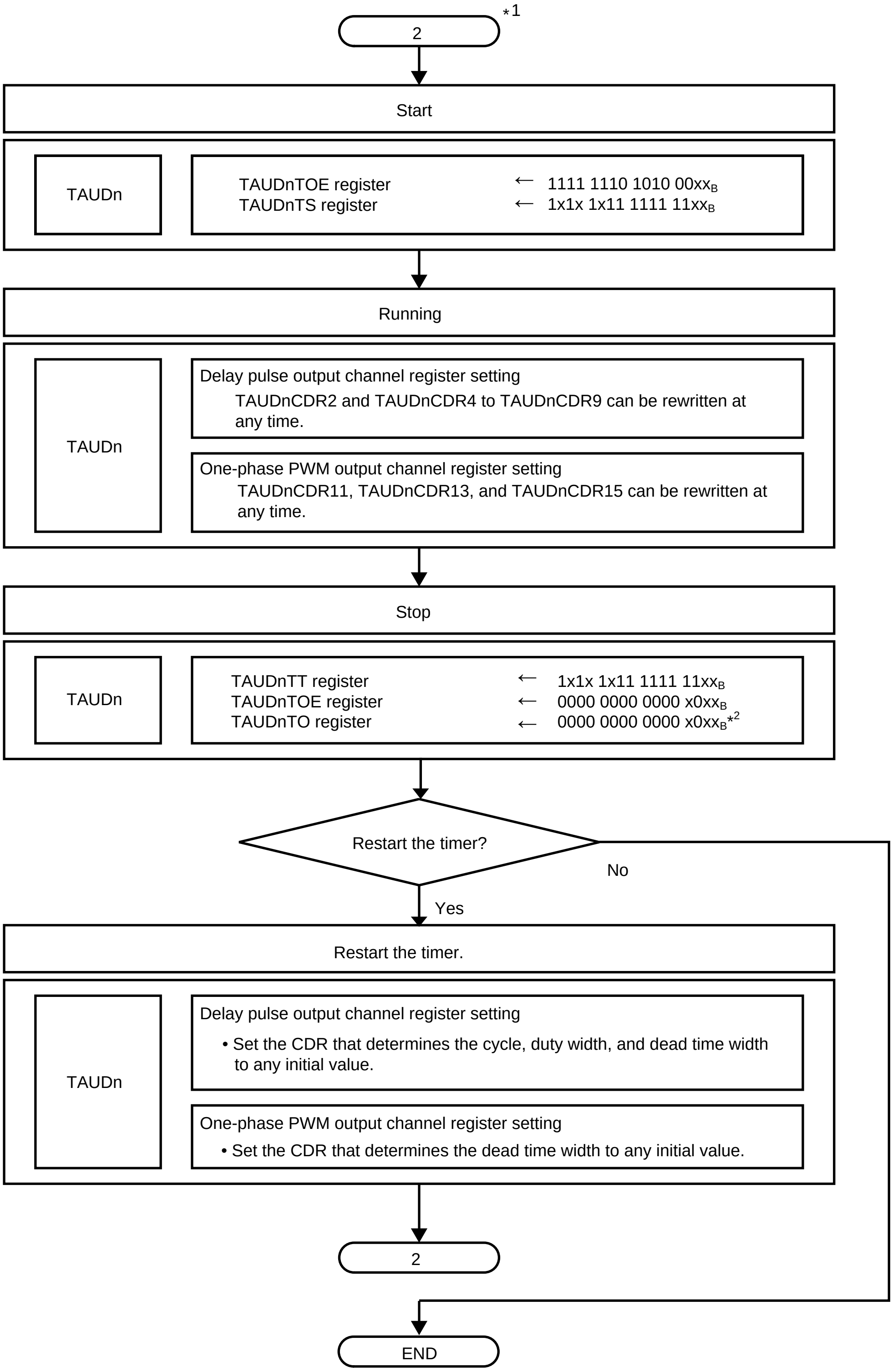
Register settings common to functions

TAUDnTOE register ← 0000 0000 0000 00xx$_B$
TAUDnTO register ← 0000 0000 0000 00xx$_B$*1
TAUDnTOM register ← 1111 1100 0000 10xx$_B$
TAUDnTOC register ← 1111 1110 1010 00xx$_B$
TAUDnTOL register ← 0000 0000 0000 00xx$_B$*1
TAUDnTDE register ← 1111 1100 0000 00xx$_B$
TAUDnTDM register ← 1111 1100 0000 00xx$_B$
TAUDnTDL register ← 1010 1000 0000 00xx$_B$*1
TAUDnTRE register ← 0000 0000 0000 00xx$_B$
TAUDnTRO register ← 0000 0000 0000 00xx$_B$
TAUDnTRC register ← 0000 0000 0000 00xx$_B$
TAUDnTME register ← 0000 0000 0000 00xx$_B$

TAUDnRDE register ← 0x0x 0x11 1111 11xx$_B$
TAUDnRDS register ← 0x0x 0x00 0000 00xx$_B$
TAUDnRDM register ← 0x0x 0x00 0000 00xx$_B$
TAUDnRDC register ← 0x0x 0x00 0000 00xx$_B$

Note 1. Change settings according to the active level of the PWM to be output.

Figure 36.49 Setup Flow (Active High Example) (continued)



Note 1. Specify the selection register and output port to use after specifying the initial settings for PIC and the timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 36.50 Setup Flow (Active High Example) (continued)

36.13.6 Setting Examples for Operation Functions

This section provides example settings for each register.

36.13.6.1 TAUDn Settings

Table 36.72 TAUDn: CH2-related (Master Channel used to Output a Delay Pulse*1)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don’t care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	Output INTm at the start of operation.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 36.73 TAUDn: CH3-related (Slave Channel used to Output a Delay Pulse*1,*2)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR3	15, 14	TAUDnCKS[1:0]	Don’t care*3	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMUR3	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Note 3. For this feature, the channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Table 36.74 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Delay Pulse*[1]) (m = 4, 6, or 8)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Table 36.75 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Delay Pulse*[1]) (m = 5, 7, or 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*[2]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	101	Start trigger: INTm detection on an upper channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

Table 36.76 TAUDn: CH11, CH13, and CH15-related (One-Phase PWM Output) (m = 11, 13, or 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*[1]	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	Start trigger: Detection of a TINm-input valid edge
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 32, Timer Array Unit D (TAUD)**.

CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output).

Table 36.77 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9	TAUDnTOE09	0 1	Disable the timer. Enable the timer.
	8	TAUDnTOE08	0	This is fixed to 0 because TOUT08 is not used.
	7	TAUDnTOE07	0 1	Disable the timer. Enable the timer.
	6	TAUDnTOE06	0	This is fixed to 0 because TOUT06 is not used
	5	TAUDnTOE05	0 1	Disable the timer. Enable the timer.
	4	TAUDnTOE04	0	This is fixed to 0 because TOUT04 is not used
	3	TAUDnTOE03	0	This is fixed to 0 because TOUT03 is not used
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*[1]	Output a low-level signal to TOUT15 to TOUT10.
	9 to 2	TAUDnTO09 to TAUDnTO02	0	Output a low-level signal to TOUT09 to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	

Table 36.77 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	1	Synchronous operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	0	Independent operation mode
	3	TAUDnTOM03	1	Synchronous operation mode
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don’t care	
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1, 0, 1, 0, 1, 0	CH5, CH7, CH9: Operation mode 2 CH4, CH6, CH8: Operation mode 1
	3	TAUDnTOC03	0	Operation mode 1
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don’t care	
TAUDnTOL	15 to 10	TAUDnTOL15 to TAUDnTOL10	0*1	Positive logic output (active high)
	9 to 2	TAUDnTOL09 to TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don’t care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control.*2
	9 to 2	TAUDnTDE09 to TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don’t care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 2	TAUDnTDM09 to TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don’t care	
TAUDnTDL	15	TAUDnTDL15	1*1	Add dead time to the negative W phase period.
	14	TAUDnTDL14	0*1	Add dead time to the positive W phase period.
	13	TAUDnTDL13	1*1	Add dead time to the negative V phase period.
	12	TAUDnTDL12	0*1	Add dead time to the positive V phase period.
	11	TAUDnTDL11	1*1	Add dead time to the negative U phase period.
	10	TAUDnTDL10	0*1	Add dead time to the positive U phase period.
	9 to 2	TAUDnTDL09 to TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don’t care	
TAUDnTRE	15 to 2	TAUDnTRE15 to TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don’t care	

Table 36.77 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRO	15 to 2	TAUDnTRO15 to TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't care	
TAUDnTRC	15 to 2	TAUDnTRC15 to TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't care	
TAUDnTME	15 to 2	TAUDnTME15 to TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't care	
	9 to 2	TAUDnRDE09 to TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 2	TAUDnRDS09 to TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't care	
	9 to 2	TAUDnRDM09 to TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't care	

Table 36.77 Common TAUDn Channel Settings

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't care	
	9 to 2	TAUDnRDC09 to TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 32, Timer Array Unit D (TAUD)**.

36.13.6.2 Peripheral Interconnections Settings

Table 36.78 Peripheral Interconnections Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	0	Select TAUDTTOUT9.
	3	PIC0REG2n203	0	Select TAUDTTOUT7.
	2	PIC0REG2n202	0	Select TAUDTTOUT5.

Section 37 PWM Output/Diagnostic (PWM-Diag)

This section contains a generic description of the PWM output/diagnostic function (PWM-Diag).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of PWM-Diag.

37.1 Features of RH850/F1KH, RH850/F1KM PWM-Diag

37.1.1 Number of Units

The PWM-Diag unit consists of a PWBA block for generating clock signals, PWGA blocks that generate PWM signals, and a PWSA block for generating triggers for A/D conversion. The numbers of individual units are listed below.

Each PWGA unit has single PWM channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 37.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
PWBA			
Number of Units	1	1	1
Name	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)
PWGA			
Number of Units	72	80	96
Name	PWGAn (n = 0 to 71)	PWGAn (n = 0 to 79)	PWGAn (n = 0 to 95)
PWSA			
Number of Units	1	1	1
Name	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)

Table 37.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
PWBA					
Number of Units	1	1	1	1	1
Name	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)
PWGA					
Number of Units	44	64	72	80	96
Name	PWGAn (n = 0 to 13, 16 to 23, 26 to 47)	PWGAn (n = 0 to 63)	PWGAn (n = 0 to 71)	PWGAn (n = 0 to 79)	PWGAn (n = 0 to 95)
PWSA					
Number of Units	1	1	1	1	1
Name	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)

Table 37.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
PWBA				
Number of Units	1	1	1	1
Name	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)	PWBAn (n = 0)
PWGA				
Number of Units	13	24	24	48
Name	PWGAn (n = 0 to 12)	PWGAn (n = 0 to 23)	PWGAn (n = 0 to 23)	PWGAn (n = 0 to 47)
PWSA				
Number of Units	1	1	1	1
Name	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)	PWSAn (n = 0)

Table 37.4 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m”; for example, PWBAnBRSm (m = 0 to 3) indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x_y = 00_01, 02_03, ..., 94_95).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj (j = 0 to 7) register.
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk (k = 0 to 2) register.
z	An A/D conversion result correspond to PWSAnPVCRx_y register is identified by the index “z”; for example, PWSAnPWDDIRz (z = 00, 01, ..., 95).
h	Sets of registers where each has the same function are identified by the index “h”; for example, the PWGAINTFhk (h = 0, 1) register.

Table 37.5 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m”; for example, PWBAnBRSm (m = 0 to 3) indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x_y = 00_01, 02_03, ..., 94_95).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj (j = 0 to 7) register.
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk (k = 0 to 2) register.
z	An A/D conversion result correspond to PWSAnPVCRx_y register is identified by the index “z”; for example, PWSAnPWDDIRz (z = 00, 01, ..., 95).
h	Sets of registers where each has the same function are identified by the index “h”; for example, the PWGAINTFhk (h = 0) register.

Table 37.6 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m”; for example, PWBAnBRSm (m = 0 to 3) indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x_y = 00_01, 02_03, ..., 46_47).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj (j = 0 to 7) register.
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk (k = 0, 1) register.
z	An A/D conversion result correspond to PWSAnPVCRx_y register is identified by the index “z”; for example, PWSAnPWDDIRz (z = 00, 01, ..., 47).
h	Sets of registers where each has the same function are identified by the index “h”; for example, the PWGAINTFhk (h = 0) register.

The following table shows values indicated by the indices of each product.

Table 37.7 Indices of Products (RH850/F1KH-D8)

Indices of each product		
176 Pins	233 Pins	324 Pins
x = 00, 02, ,,, 70 y = 01, 03, ,,, 71 z = 00, 01, ,,, 71	x = 00, 02, ,,, 78 y = 01, 03, ,,, 79 z = 00, 01, ,,, 79	x = 00, 02, ,,, 94 y = 01, 03, ,,, 95 z = 00, 01, ,,, 95
j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0 to 2	k = 0 to 2	k = 0 to 2
h = 0, 1	h = 0, 1	h = 0, 1

Table 37.8 Indices of Products (RH850/F1KM-S4)

Indices of each Product				
100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
x = 00, 02, ,,, 12, 16, ,,, 22, 26, ,,,46 y = 01, 03, ,,, 13, 17, ,,, 23, 27, ,,,47 z = 00, 01, ,,, 13, 16, ,,, 23, 26, ,,, 47	x = 00, 02, ,,, 62 y = 01, 03, ,,, 63 z = 00, 01, ,,, 63	x = 00, 02, ,,, 70 y = 01, 03, ,,, 71 z = 00, 01, ,,, 71	x = 00, 02, ,,, 78 y = 01, 03, ,,, 79 z = 00, 01, ,,, 79	x = 00, 02, ,,, 94 y = 01, 03, ,,, 95 z = 00, 01, ,,, 95
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0, 1	k = 0, 1	k = 0 to 2	k = 0 to 2	k = 0 to 2
h = 0	h = 0	h = 0	h = 0	h = 0

Table 37.9 Indices of Products (RH850/F1KM-S1)

Indices of each Product			
48 Pins	64 Pins	80 Pins	100 Pins
x = 00, 02, ,,, 12 y = 01, 03, ,,, 13*[1] z = 00, 01, ,,, 12	x = 00, 02, ,,, 22 y = 01, 03, ,,, 23 z = 00, 01, ,,, 23	x = 00, 02, ,,, 22 y = 01, 03, ,,, 23 z = 00, 01, ,,, 23	x = 00, 02, ,,, 46 y = 01, 03, ,,, 47 z = 00, 01, ,,, 47
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0	k = 0	k = 0	k = 0, 1
h = 0	h = 0	h = 0	h = 0

Note 1. Channels 0 to 12 are provided in PWM-Diag.

37.1.2 Register Base Addresses

PWM-Diag base addresses are listed in the following table.

PWM-Diag register addresses are given as offsets from the base addresses.

Table 37.10 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<PWBAn_base>	FFE7 2800_H
<PWGAn_base>	FFE7 1000_H + 40_H × n
<PWSAn_base>	FFE7 0000_H
<SLPW_base>	FFE7 3000_H
<PWGAINTF_base>	FFE7 3100_H

37.1.3 Clock Supply

The PWM-Diag clock supply is shown in the following table.

Table 37.11 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
PWBAn	PCLK	CKSCLK_IPERI2	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI2	Bus clock
PWGAn	PCLK	CKSCLK_IPERI2	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI2	Bus clock
PWSAn	PCLK	CKSCLK_IPERI2	Module clock
	Register access clock	CPUCLK_L, CKSCLK_IPERI2	Bus clock

37.1.4 Interrupt Requests

PWM-Diag interrupt requests are listed in the following table.
Regarding interrupt connection image, refer to **Figure 37.1, PWGA Interrupt Connection Image (PWGA 96 Channels Embedded)**.

Table 37.12 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	92, 95	—
PWGA_INT1	PWGA1 interrupt	92, 95	—
PWGA_INT2	PWGA2 interrupt	92, 95	—
PWGA_INT3	PWGA3 interrupt	92, 95	—
PWGA_INT4	PWGA4 interrupt	92, 95	—
PWGA_INT5	PWGA5 interrupt	92, 95	—
PWGA_INT6	PWGA6 interrupt	92, 95	—
PWGA_INT7	PWGA7 interrupt	92, 95	—
PWGA_INT8	PWGA8 interrupt	92, 95	—
PWGA_INT9	PWGA9 interrupt	92, 95	—
PWGA_INT10	PWGA10 interrupt	92, 95	—
PWGA_INT11	PWGA11 interrupt	92, 95	—
PWGA_INT12	PWGA12 interrupt	92, 95	—
PWGA_INT13	PWGA13 interrupt	92, 95	—
PWGA_INT14	PWGA14 interrupt	92, 95	—
PWGA_INT15	PWGA15 interrupt	92, 95	—
PWGA_INT16	PWGA16 interrupt	92, 95	—
PWGA_INT17	PWGA17 interrupt	92, 95	—
PWGA_INT18	PWGA18 interrupt	92, 95	—
PWGA_INT19	PWGA19 interrupt	92, 95	—
PWGA_INT20	PWGA20 interrupt	92, 95	—
PWGA_INT21	PWGA21 interrupt	92, 95	—
PWGA_INT22	PWGA22 interrupt	92, 95	—
PWGA_INT23	PWGA23 interrupt	92, 95	—
PWGA_INT24	PWGA24 interrupt	92, 95	—
PWGA_INT25	PWGA25 interrupt	92, 95	—
PWGA_INT26	PWGA26 interrupt	92, 95	—
PWGA_INT27	PWGA27 interrupt	92, 95	—

Table 37.12 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT28	PWGA28 interrupt	92, 95	—
PWGA_INT29	PWGA29 interrupt	92, 95	—
PWGA_INT30	PWGA30 interrupt	92, 95	—
PWGA_INT31	PWGA31 interrupt	92, 95	—
PWGA_INT32	PWGA32 interrupt	93, 96	—
PWGA_INT33	PWGA33 interrupt	93, 96	—
PWGA_INT34	PWGA34 interrupt	93, 96	—
PWGA_INT35	PWGA35 interrupt	93, 96	—
PWGA_INT36	PWGA36 interrupt	93, 96	—
PWGA_INT37	PWGA37 interrupt	93, 96	—
PWGA_INT38	PWGA38 interrupt	93, 96	—
PWGA_INT39	PWGA39 interrupt	93, 96	—
PWGA_INT40	PWGA40 interrupt	93, 96	—
PWGA_INT41	PWGA41 interrupt	93, 96	—
PWGA_INT42	PWGA42 interrupt	93, 96	—
PWGA_INT43	PWGA43 interrupt	93, 96	—
PWGA_INT44	PWGA44 interrupt	93, 96	—
PWGA_INT45	PWGA45 interrupt	93, 96	—
PWGA_INT46	PWGA46 interrupt	93, 96	—
PWGA_INT47	PWGA47 interrupt	93, 96	—
PWGA_INT48	PWGA48 interrupt	93, 96	—
PWGA_INT49	PWGA49 interrupt	93, 96	—
PWGA_INT50	PWGA50 interrupt	93, 96	—
PWGA_INT51	PWGA51 interrupt	93, 96	—
PWGA_INT52	PWGA52 interrupt	93, 96	—
PWGA_INT53	PWGA53 interrupt	93, 96	—
PWGA_INT54	PWGA54 interrupt	93, 96	—
PWGA_INT55	PWGA55 interrupt	93, 96	—
PWGA_INT56	PWGA56 interrupt	93, 96	—
PWGA_INT57	PWGA57 interrupt	93, 96	—
PWGA_INT58	PWGA58 interrupt	93, 96	—
PWGA_INT59	PWGA59 interrupt	93, 96	—
PWGA_INT60	PWGA60 interrupt	93, 96	—
PWGA_INT61	PWGA61 interrupt	93, 96	—
PWGA_INT62	PWGA62 interrupt	93, 96	—
PWGA_INT63	PWGA63 interrupt	93, 96	—
PWGA_INT64	PWGA64 interrupt	94, 97	—
PWGA_INT65	PWGA65 interrupt	94, 97	—
PWGA_INT66	PWGA66 interrupt	94, 97	—
PWGA_INT67	PWGA67 interrupt	94, 97	—
PWGA_INT68	PWGA68 interrupt	94, 97	—
PWGA_INT69	PWGA69 interrupt	94, 97	—
PWGA_INT70	PWGA70 interrupt	94, 97	—
PWGA_INT71	PWGA71 interrupt	94, 97	—
PWGA_INT72	PWGA72 interrupt	94, 97	—

Table 37.12 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT73	PWGA73 interrupt	94, 97	—
PWGA_INT74	PWGA74 interrupt	94, 97	—
PWGA_INT75	PWGA75 interrupt	94, 97	—
PWGA_INT76	PWGA76 interrupt	94, 97	—
PWGA_INT77	PWGA77 interrupt	94, 97	—
PWGA_INT78	PWGA78 interrupt	94, 97	—
PWGA_INT79	PWGA79 interrupt	94, 97	—
PWGA_INT80	PWGA80 interrupt	94, 97	—
PWGA_INT81	PWGA81 interrupt	94, 97	—
PWGA_INT82	PWGA82 interrupt	94, 97	—
PWGA_INT83	PWGA83 interrupt	94, 97	—
PWGA_INT84	PWGA84 interrupt	94, 97	—
PWGA_INT85	PWGA85 interrupt	94, 97	—
PWGA_INT86	PWGA86 interrupt	94, 97	—
PWGA_INT87	PWGA87 interrupt	94, 97	—
PWGA_INT88	PWGA88 interrupt	94, 97	—
PWGA_INT89	PWGA89 interrupt	94, 97	—
PWGA_INT90	PWGA90 interrupt	94, 97	—
PWGA_INT91	PWGA91 interrupt	94, 97	—
PWGA_INT92	PWGA92 interrupt	94, 97	—
PWGA_INT93	PWGA93 interrupt	94, 97	—
PWGA_INT94	PWGA94 interrupt	94, 97	—
PWGA_INT95	PWGA95 interrupt	94, 97	—
PWSA_INT_QFULL	PWSA queue full interrupt	91	—

Table 37.13 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	92	—
PWGA_INT1	PWGA1 interrupt	92	—
PWGA_INT2	PWGA2 interrupt	92	—
PWGA_INT3	PWGA3 interrupt	92	—
PWGA_INT4	PWGA4 interrupt	92	—
PWGA_INT5	PWGA5 interrupt	92	—
PWGA_INT6	PWGA6 interrupt	92	—
PWGA_INT7	PWGA7 interrupt	92	—
PWGA_INT8	PWGA8 interrupt	92	—
PWGA_INT9	PWGA9 interrupt	92	—
PWGA_INT10	PWGA10 interrupt	92	—
PWGA_INT11	PWGA11 interrupt	92	—
PWGA_INT12	PWGA12 interrupt	92	—
PWGA_INT13	PWGA13 interrupt	92	—
PWGA_INT14	PWGA14 interrupt	92	—
PWGA_INT15	PWGA15 interrupt	92	—
PWGA_INT16	PWGA16 interrupt	92	—
PWGA_INT17	PWGA17 interrupt	92	—
PWGA_INT18	PWGA18 interrupt	92	—
PWGA_INT19	PWGA19 interrupt	92	—
PWGA_INT20	PWGA20 interrupt	92	—
PWGA_INT21	PWGA21 interrupt	92	—
PWGA_INT22	PWGA22 interrupt	92	—
PWGA_INT23	PWGA23 interrupt	92	—
PWGA_INT24	PWGA24 interrupt	92	—
PWGA_INT25	PWGA25 interrupt	92	—
PWGA_INT26	PWGA26 interrupt	92	—
PWGA_INT27	PWGA27 interrupt	92	—
PWGA_INT28	PWGA28 interrupt	92	—
PWGA_INT29	PWGA29 interrupt	92	—
PWGA_INT30	PWGA30 interrupt	92	—
PWGA_INT31	PWGA31 interrupt	92	—
PWGA_INT32	PWGA32 interrupt	93	—
PWGA_INT33	PWGA33 interrupt	93	—
PWGA_INT34	PWGA34 interrupt	93	—
PWGA_INT35	PWGA35 interrupt	93	—
PWGA_INT36	PWGA36 interrupt	93	—
PWGA_INT37	PWGA37 interrupt	93	—
PWGA_INT38	PWGA38 interrupt	93	—
PWGA_INT39	PWGA39 interrupt	93	—
PWGA_INT40	PWGA40 interrupt	93	—
PWGA_INT41	PWGA41 interrupt	93	—
PWGA_INT42	PWGA42 interrupt	93	—
PWGA_INT43	PWGA43 interrupt	93	—
PWGA_INT44	PWGA44 interrupt	93	—

Table 37.13 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT45	PWGA45 interrupt	93	—
PWGA_INT46	PWGA46 interrupt	93	—
PWGA_INT47	PWGA47 interrupt	93	—
PWGA_INT48	PWGA48 interrupt	93	—
PWGA_INT49	PWGA49 interrupt	93	—
PWGA_INT50	PWGA50 interrupt	93	—
PWGA_INT51	PWGA51 interrupt	93	—
PWGA_INT52	PWGA52 interrupt	93	—
PWGA_INT53	PWGA53 interrupt	93	—
PWGA_INT54	PWGA54 interrupt	93	—
PWGA_INT55	PWGA55 interrupt	93	—
PWGA_INT56	PWGA56 interrupt	93	—
PWGA_INT57	PWGA57 interrupt	93	—
PWGA_INT58	PWGA58 interrupt	93	—
PWGA_INT59	PWGA59 interrupt	93	—
PWGA_INT60	PWGA60 interrupt	93	—
PWGA_INT61	PWGA61 interrupt	93	—
PWGA_INT62	PWGA62 interrupt	93	—
PWGA_INT63	PWGA63 interrupt	93	—
PWGA_INT64	PWGA64 interrupt	94	—
PWGA_INT65	PWGA65 interrupt	94	—
PWGA_INT66	PWGA66 interrupt	94	—
PWGA_INT67	PWGA67 interrupt	94	—
PWGA_INT68	PWGA68 interrupt	94	—
PWGA_INT69	PWGA69 interrupt	94	—
PWGA_INT70	PWGA70 interrupt	94	—
PWGA_INT71	PWGA71 interrupt	94	—
PWGA_INT72	PWGA72 interrupt	94	—
PWGA_INT73	PWGA73 interrupt	94	—
PWGA_INT74	PWGA74 interrupt	94	—
PWGA_INT75	PWGA75 interrupt	94	—
PWGA_INT76	PWGA76 interrupt	94	—
PWGA_INT77	PWGA77 interrupt	94	—
PWGA_INT78	PWGA78 interrupt	94	—
PWGA_INT79	PWGA79 interrupt	94	—
PWGA_INT80	PWGA80 interrupt	94	—
PWGA_INT81	PWGA81 interrupt	94	—
PWGA_INT82	PWGA82 interrupt	94	—
PWGA_INT83	PWGA83 interrupt	94	—
PWGA_INT84	PWGA84 interrupt	94	—
PWGA_INT85	PWGA85 interrupt	94	—
PWGA_INT86	PWGA86 interrupt	94	—
PWGA_INT87	PWGA87 interrupt	94	—
PWGA_INT88	PWGA88 interrupt	94	—
PWGA_INT89	PWGA89 interrupt	94	—

Table 37.13 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT90	PWGA90 interrupt	94	—
PWGA_INT91	PWGA91 interrupt	94	—
PWGA_INT92	PWGA92 interrupt	94	—
PWGA_INT93	PWGA93 interrupt	94	—
PWGA_INT94	PWGA94 interrupt	94	—
PWGA_INT95	PWGA95 interrupt	94	—
INTQFULL	PWSA queue full interrupt	91	—

Table 37.14 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	92	—
PWGA_INT1	PWGA1 interrupt	92	—
PWGA_INT2	PWGA2 interrupt	92	—
PWGA_INT3	PWGA3 interrupt	92	—
PWGA_INT4	PWGA4 interrupt	92	—
PWGA_INT5	PWGA5 interrupt	92	—
PWGA_INT6	PWGA6 interrupt	92	—
PWGA_INT7	PWGA7 interrupt	92	—
PWGA_INT8	PWGA8 interrupt	92	—
PWGA_INT9	PWGA9 interrupt	92	—
PWGA_INT10	PWGA10 interrupt	92	—
PWGA_INT11	PWGA11 interrupt	92	—
PWGA_INT12	PWGA12 interrupt	92	—
PWGA_INT13	PWGA13 interrupt	92	—
PWGA_INT14	PWGA14 interrupt	92	—
PWGA_INT15	PWGA15 interrupt	92	—
PWGA_INT16	PWGA16 interrupt	92	—
PWGA_INT17	PWGA17 interrupt	92	—
PWGA_INT18	PWGA18 interrupt	92	—
PWGA_INT19	PWGA19 interrupt	92	—
PWGA_INT20	PWGA20 interrupt	92	—
PWGA_INT21	PWGA21 interrupt	92	—
PWGA_INT22	PWGA22 interrupt	92	—
PWGA_INT23	PWGA23 interrupt	92	—
PWGA_INT24	PWGA24 interrupt	92	—
PWGA_INT25	PWGA25 interrupt	92	—
PWGA_INT26	PWGA26 interrupt	92	—
PWGA_INT27	PWGA27 interrupt	92	—
PWGA_INT28	PWGA28 interrupt	92	—
PWGA_INT29	PWGA29 interrupt	92	—
PWGA_INT30	PWGA30 interrupt	92	—
PWGA_INT31	PWGA31 interrupt	92	—
PWGA_INT32	PWGA32 interrupt	93	—
PWGA_INT33	PWGA33 interrupt	93	—
PWGA_INT34	PWGA34 interrupt	93	—
PWGA_INT35	PWGA35 interrupt	93	—
PWGA_INT36	PWGA36 interrupt	93	—
PWGA_INT37	PWGA37 interrupt	93	—
PWGA_INT38	PWGA38 interrupt	93	—
PWGA_INT39	PWGA39 interrupt	93	—
PWGA_INT40	PWGA40 interrupt	93	—
PWGA_INT41	PWGA41 interrupt	93	—
PWGA_INT42	PWGA42 interrupt	93	—
PWGA_INT43	PWGA43 interrupt	93	—
PWGA_INT44	PWGA44 interrupt	93	—

Table 37.14 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
PWGA_INT45	PWGA45 interrupt	93	—
PWGA_INT46	PWGA46 interrupt	93	—
PWGA_INT47	PWGA47 interrupt	93	—
INTQFULL	PWSA queue full interrupt	91	—

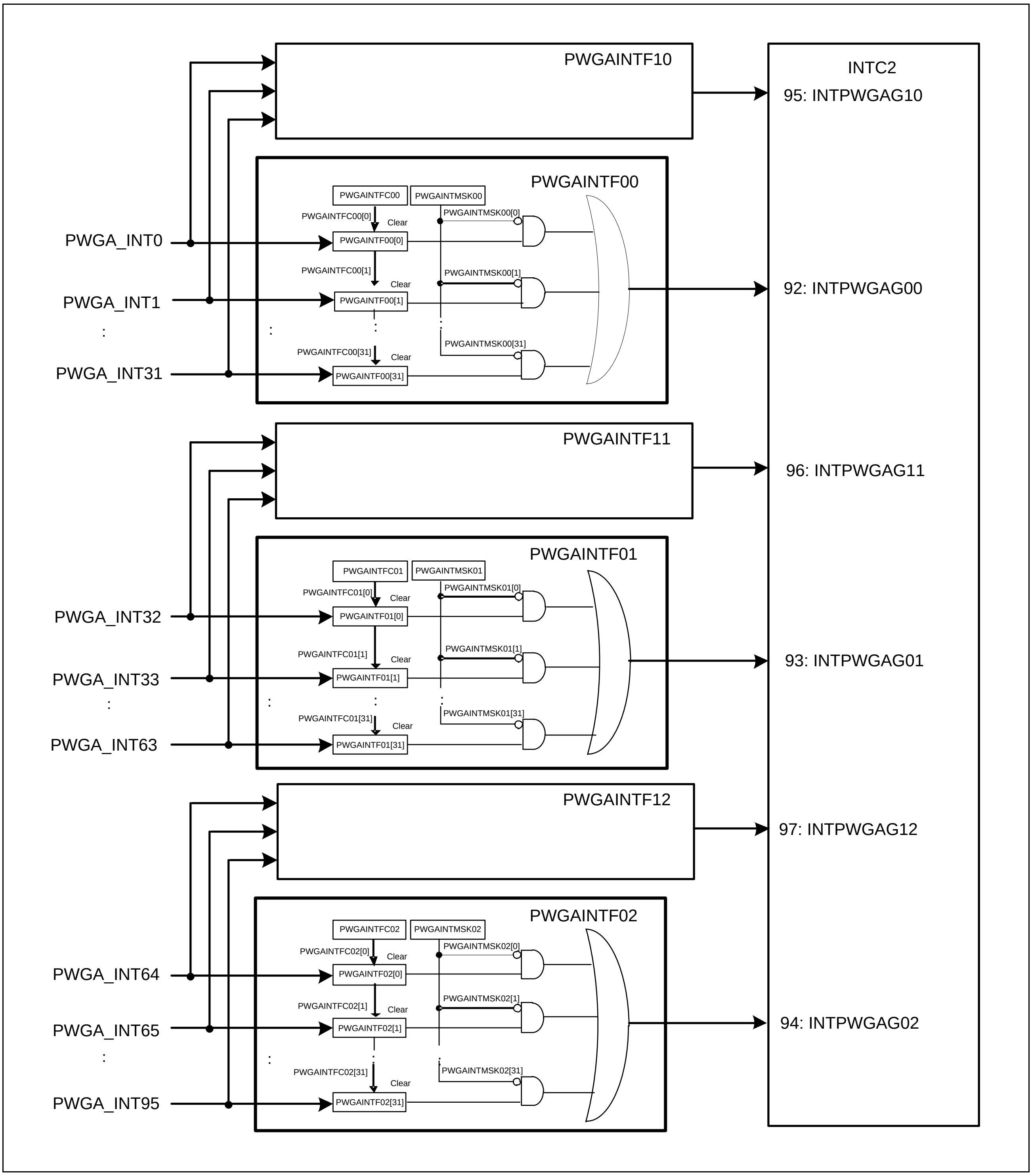


Figure 37.1 PWGA Interrupt Connection Image (PWGA 96 Channels Embedded)

37.1.5 Reset Sources

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

Table 37.15 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
PWBAn	All reset sources (ISORES)
PWGAn	
PWSAn	

37.1.6 External Input/Output Signals

External input/output signals of the PWM-Diag are listed below.

Table 37.16 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
PWGA_TOUTn (Unit: PWGA)	PWGA unit n output	PWGAnO

CAUTION

When port P8_6 is used as PWGA38O, port P8_6 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$, Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.

37.1.7 Internal Output Signals

The I/O signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

Table 37.17 Internal Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Connected to
PWBA0		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
PWGAn		
PWGA_TRGOUTn	PWGAn trigger	PWSA0
PWSA0		
PWSA_ADTRG[1:0]	A/D converter unit select signal	ADCA0, ADCA1
PWSA_PVCR_VALUE[13:0]	A/D converter control signal	ADCA0, ADCA1
ADCAn		
ADC_CONV_ENDn	A/D conversion completion signal	PWSA0
ADC_PWDDRn[15:0]	A/D conversion result signal	PWSA0

Table 37.18 Internal Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Connected to
PWBA0		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
PWGAn		
PWGA_TRGOUTn	PWGAn trigger	PWSA0
PWSA0		
PWSA_ADTRG[1:0]	A/D converter unit select signal	ADCA0, ADCA1
PWSA_PVCR_VALUE[13:0]	A/D converter control signal	ADCA0, ADCA1
ADCAn		
ADC_CONV_ENDn	A/D conversion completion signal	PWSA0
ADC_PWDDRn[15:0]	A/D conversion result signal	PWSA0

Table 37.19 Internal Output Signals (RH850/F1KM-S1)

Unit Signal Name	Description	Connected to
PWBA0		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
PWGAn		
PWGA_TRGOUTn	PWGAn trigger	PWSA0
PWSA0		
PWSA_ADTRG[0]	A/D converter unit select signal	ADCA0
PWSA_PVCR_VALUE[13:0]	A/D converter control signal	ADCA0
ADCAn		
ADC_CONV_ENDn	A/D conversion completion signal	PWSA0
ADC_PWDDRn[15:0]	A/D conversion result signal	PWSA0

37.1.8 Functional Overview

This function is comprised of four types of units: clock divider (PWBA), PWM generator (PWGA), A/D conversion trigger select function (PWSA), and A/D converter (ADCA).

PWBA

- Clock divider

  PWBA generates a PWMCLKm count clock signal by frequency division of PCLK and supplies it to the PWM generator PWGA.
  The cycle of the PWMCLKm count clock signal can be calculated from the setting of the PWBAnBRSm register by the equation below.

  (When the PWBAnBRSm = 1 to 2047)
  PWMCLKm count clock cycle = (PWBAnBRSm value × 2) × PCLK cycle

  (When the PWBAnBRSm = 0)
  PWMCLKm count clock cycle = PCLK cycle

  In addition, PWBA can control operation when the on-chip debugger is in use by using the PWBAnEMU register.

PWGA

PWGA outputs PWM waveforms and A/D conversion trigger to PWSA by using the clock PWMCLKm input from PWBA.

- PWM waveform output PWGA_TOUTn

  This generator outputs PWM waveforms from the PWGA_TOUTn pin. The PWM cycle is controlled by the match timing of PWGAnCNT register value and the PWGA_PERIOD input value. PWGA_PERIOD input value can be set by PWGAPRD and PWGAPRDSLk register setting. Set the high-level period of PWM output in the PWGAnCSDR and PWGAnCRDR registers.

  The PWM waveform cycle and duty can be calculated by the equations below.

  PWM waveform cycle = (PWGA_PERIOD + 1) × PWMCLKm count clock cycle

  When PWGAnCRDR[11:0] > PWGAnCSDR[11:0],

  High-level period of PWM waveform =
  (PWGAnCRDR register value – PWGAnCSDR register value) × PWMCLKm count clock cycle

  PWM waveform duty (%) = High-level period of PWM waveform/PWM waveform cycle × 100 = (PWGAnCRDR register value – PWGAnCSDR register value)/(PWGA_PERIOD + 1) × 100

  Note that the PWM output is fixed to the low level when the PWGAnCRDR register value is equal to the PWGAnCSDR register value.

  When $1xxx_H$ or the value higher than PWGA_PERIOD is set in the PWGAnCRDR register (i.e. bit 12 is set to 1), the PWM output is fixed to the high level.

- A/D conversion trigger output PWGA_TRGOUTn

  The A/D conversion trigger signal PWGA_TRGOUTn for PWSA is generated when the PWGAnCTDR register value and the PWGAnCNT register value match and its timing can be set by PWGAnOCL.
  The output enable/disable of PWGA_TRGOUTn is controlled by PWGAnTCR register. The timing can be calculated by the equation below.
  A/D conversion trigger signal generation timing = PWGAnCTDR register value × PWMCLKm count clock cycle

- PWGA interrupt request signal PWGA_INTn

  PWGA generates the interrupt request signal PWGA_INTn at the falling edge of the PWM output PWGA_TOUTn. When the PWM output is fixed to the low level, PWGA_INTn is generated when the PWGAnCRDR register value and the PWGAnCNT register value match; when the PWM output is fixed to the high level, it is generated when the PWGAnCNT register value and the PWGA_PERIOD input value match.

PWSA

PWSA transmits the required setting information to the A/D converter and outputs the A/D conversion start trigger, based on the A/D conversion trigger signal PWGA_TRGOUTn from the PWM generator (PWGA).

- A/D conversion control by PWSA

  PWSA outputs the information required for the A/D conversion, which is set in the corresponding PWSAnPVCRx_y register for the channel number of the trigger input from PWGAn, (i.e., information on ADC physical channel, external MPX control, and error detection level selection) to the A/D converter.
  At the same timing, A/D conversion trigger (PWSA_ADTRG) is output to ADCA0 or ADCA1. (A maximum of eight input trigger signal PWGA_TRGOUTn data received during A/D conversion are stored and kept in PWSAnQUE.)
  The setting information to be output to the A/D converter is kept until the next trigger is generated.
  When the A/D conversion triggered by the PWM-Diag function is completed in the A/D converter, PWSA triggers the next A/D conversion based on the data stored in the PWSAnQUE register.

- Queuing of A/D conversion triggers from PWGA

  The A/D conversion trigger signal (PWGA_TRGOUTn) input from PWGAn is stored in the PWSAnQUEj register as a channel number. The PWSAnQUEj register stores a maximum of eight channel numbers of the A/D conversion trigger signal PWGA_TRGOUTn received during A/D conversion in a queue structure.
  A PWSA queue full interrupt occurs in the following states, when the queue of the PWSAnQUEj register becomes full

  - A trigger number is written to PWSAnQUE7
  - A trigger number has already been written to PWSAnQUE7 and cannot be written when PWGA_TRGOUTn is input.

- Storing A/D conversion result

  The A/D conversion result is stored in PWSAnPWDDIRz register.

ADCA

A/D conversion is executed upon receipt of information required for A/D conversion and A/D conversion trigger from PWSA.

A/D conversion is executed using the PWM-Diag-dedicated scan group; on completion of the A/D conversion, it is reported to the PWSA.

For the basic operation of the A/D converter, see **Section 38, A/D Converter (ADCA)**.

For the A/D converter operation with the PWM-Diag function, see **Section 38.4.7.1, A/D Conversion with PWM-Diag Enabled**.

37.1.9 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.

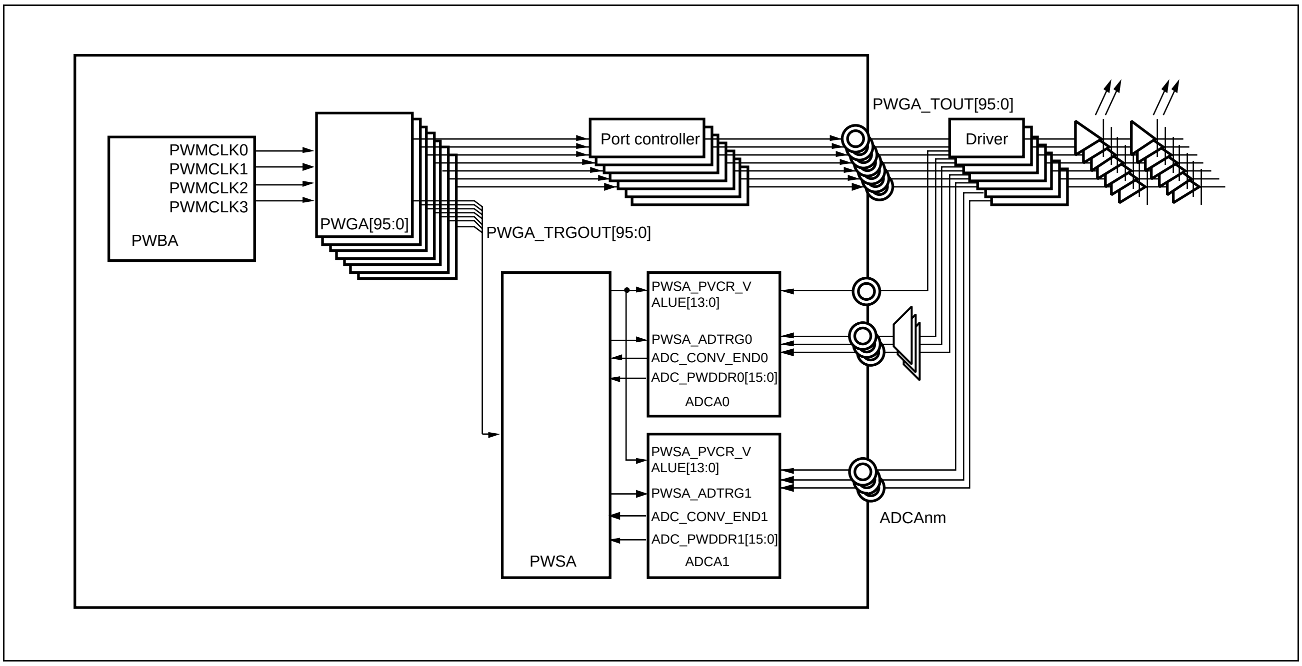


Figure 37.2 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter

37.2 Registers

37.2.1 List of Registers

PWM Output/Diagnostic registers are listed in the following table.

<PWBAn_base>, <PWSAn_base>, and <PWGAn_base> are defined in **Section 37.1.2, Register Base Addresses**.

Table 37.20 List of Registers

Module Name	Register Name	Symbol	Address
PWBAn	PWMCLKm cycle configuration register	PWBAnBRSm	<PWBAn_base> + 0004_H × m
	PWMCLKm enable status register	PWBAnTE	<PWBAn_base> + 0010_H
	PWMCLKm start trigger register	PWBAnTS	<PWBAn_base> + 0014_H
	PWMCLKm stop trigger register	PWBAnTT	<PWBAn_base> + 0018_H
	PWBA emulation register	PWBAnEMU	<PWBAn_base> + $001C_H$
PWGAn	PWM output set condition register	PWGAnCSDR	<PWGAn_base> + 0000_H
	PWM output reset condition register	PWGAnCRDR	<PWGAn_base> + 0004_H
	PWGA_TRGOUTn generation condition register	PWGAnCTDR	<PWGAn_base> + 0008_H
	Buffer register reload trigger register	PWGAnRDT	<PWGAn_base> + $000C_H$
	Buffer register reload status register	PWGAnRSF	<PWGAn_base> + 0010_H
	PWM cycle count register	PWGAnCNT	<PWGAn_base> + 0014_H
	PWGA_TRGOUTn control register	PWGAnTCR	<PWGAn_base> + 0018_H
	PWGAnTCR buffer register	PWGAnTCBR	<PWGAn_base> + $001C_H$
	PWGA control register	PWGAnCTL	<PWGAn_base> + 0020_H
	PWGAnCSDR buffer register	PWGAnCSBR	<PWGAn_base> + 0024_H
	PWGAnCRDR buffer register	PWGAnCRBR	<PWGAn_base> + 0028_H
	PWGAnCTDR buffer register	PWGAnCTBR	<PWGAn_base> + $002C_H$
SLPWG	PWGA synchronous trigger register	SLPWGAk	<SLPW_base> + k × 4_H
	PWGA period setting register	PWGAPRD	<SLPW_base> + $000C_H$
	PWGA period selection register	PWGAPRDSLk	<SLPW_base> + 0010_H + k × 4_H
PWSAn	PWSA control register	PWSAnCTL	<PWSAn_base> + 0000_H
	Trigger queue status register	PWSAnSTR	<PWSAn_base> + 0004_H
	Trigger queue status clear register	PWSAnSTC	<PWSAn_base> + 0008_H
	Trigger queue register	PWSAnQUEj	<PWSAn_base> + 0020_H + j × 4_H
	PWM-Diag mode A/D setting register	PWSAnPVCRx_y	<PWSAn_base> + 0040_H + x × 2_H
	PWM-Diag data supplementary information register	PWSAnPWDDIRz	<PWSAn_base> + 0200_H + z × 4_H
	PWSA emulation control register	PWSAnEMU	<PWSAn_base> + $000C_H$
PWGA_INTF	PWGA interrupt factor register	PWGAINTFhk	<PWGAINTF_base> + h × 30_H + k × 10_H
	PWGA interrupt mask register	PWGAINTMSKhk	<PWGAINTF_base> + h × 30_H + k × 10_H + 4_H
	PWGA interrupt factor clear register	PWGAINTFChk	<PWGAINTF_base> + h × 30_H + k × 10_H + 8_H

37.2.1.1 PWBAnBRSm — PWMCLKm Cycle Configuration Register

This register sets the clock cycle of PWMCLKm.

Access: This register can be read or written in 16-bit units.

Address: <PWBAn_base> + 0004_H × m

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWBAnBRSm[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.21 PWBAnBRSm Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	PWBAnBRSm[10:0]	Register for setting the clock cycle of PWMCLKm. – PWBAnBRSm = 0: PWMCLKm = PCLK – PWBAnBRSm = 1: PWMCLKm = PCLK / (2 × 1) – PWBAnBRSm = 2: PWMCLKm = PCLK / (2 × 2) ... – PWBAnBRSm = n: PWMCLKm = PCLK / (2 × n) (n = 1 to 2047) These bits can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).

37.2.1.2 PWBAnTE — PWMCLKm Enable Status Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWBAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTE3	PWBAnTE2	PWBAnTE1	PWBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.22 PWBAnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	PWBAnTE3	A status flag indicating the operation status of PWMCLK3 0: Not operating 1: Operating
2	PWBAnTE2	A status flag indicating the operation status of PWMCLK2 0: Not operating 1: Operating
1	PWBAnTE1	A status flag indicating the operation status of PWMCLK1 0: Not operating 1: Operating
0	PWBAnTE0	A status flag indicating the operation status of PWMCLK0 0: Not operating 1: Operating

37.2.1.3 PWBAnTS — PWMCLKm Start Trigger Register

This register is a start trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0014_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTS3	PWBAnTS2	PWBAnTS1	PWBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 37.23 PWBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTS3	Start Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Starts the output of PWMCLK0.

37.2.1.4 PWBAnTT — PWMCLKm Stop Trigger Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTT3	PWBAnTT2	PWBAnTT1	PWBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 37.24 PWBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTT3	Stop Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Stops the output of PWMCLK0.

37.2.1.5 PWBAnEMU — PWBA Emulation Register

This register sets the operation during emulation.

Access: This register can be read or written in 8-bit units.

Address: <PWBAn_base> + $001C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PWBAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 37.25 PWBAnEMU Register Contents

Bit Position	Bit Name	Function
7	PWBAnSVSDIS	(When the EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: The count clock is stopped when the debugger is controlling the microcontroller (by using break points, etc.). 1: The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.). This bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

37.2.1.6 PWGAnCTL — PWGA Control Register

PWGAnCTL is used to select the count clock from PWBA and setting output condition of PWGA_TRGOUTn.

Access: This register can be read or written in 8-bit units.

Address: <PWGAn_base> + 0020$_{\text{H}}$

Value after reset: 00$_{\text{H}}$

Bit	7	6	5	4	3	2	1	0
	PWGAnOCL	—	PWGAnTCUT[1:0]		—	—	PWGAnCKS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R/W	R/W

Table 37.26 PWGAnCTL Register Contents

Bit Position	Bit Name	Function
7	PWGAnOCL	This bit select the PWGA_TRGOUTn output condition related with PWGA_TOUTn. 0: PWGA_TRGOUTn output at the condition of high level of PWGA_TOUTn 1: PWGA_TRGOUTn output at the condition of both high and low level of PWGA_TOUTn
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	PWGAnTCUT[1:0]	These bits select the update timing of PWGAnTCBR register. 00: No update 01: Update at rising edge of PWGA_TOUTn at the condition of PWGAnTCBR rewrite in progress (PWGAnRSFT = 1) 10: Update at falling edge of PWGA_TOUTn at the condition of PWGAnTCBR rewrite in progress (PWGAnRSFT = 1) 11: Update immediately (PWGAnTCBR rewrite request trigger of PWGAnRDTT = 1 setting is invalid)
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PWGAnCKS[1:0]	Count Clock Enable Input PWMCLK3 to PWMCLK0 Select 00: Uses PWMCLK0 as count clock 01: Uses PWMCLK1 as count clock 10: Uses PWMCLK2 as count clock 11: Uses PWMCLK3 as count clock These bits can only be rewritten when the PWGAn operation is stopped (SLPWGAk.SLPWGA[31:0] = 0).

37.2.1.7 PWGAnCNT — PWM Cycle Count Register

This is a count register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0014_H

Value after reset: $0FFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCNT[11:0]											
Value after reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.27 PWGAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCNT[11:0]	12-bit counter value

37.2.1.8 PWGAnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGA_TOUTn output.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.28 PWGAnCSDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCSDR[11:0]	These bits set the setting condition for PWM output. The set value is reflected to the PWGAnCSBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

NOTE

In case of changing PWM period by PWGAPRD and PWGAPRDSLk registers, the setting of PWGAnCSDR > PWGA_PERIOD is prohibited.

37.2.1.9 PWGAnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGA_TOUTn output.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRDR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.29 PWGAnCRDR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 0	PWGAnCRDR[12:0]	These bits set the reset condition for PWM output. The set value is reflected to the PWGAnCRBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

37.2.1.10 PWGAnCTDR — PWGA_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGA_TRGOUTn.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.30 PWGAnCTDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCTDR[11:0]	These bits set the A/D conversion trigger generation condition for PWSAn. The set value is reflected to the PWGAnCTBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

37.2.1.11 PWGAnTCR — PWGA_TRGOUTn Control Register

This register controls enable/disable of PWGA_TRGOUTn.

Access: This register can be read or written in 8-bit units.

Address: <PWGAn_base> + 0018$_H$

Value after reset: 01$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnTOE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Table 37.31 PWGAnTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWGAnTOE	This bit set the output enable/disable of PWGA_TRGOUTn 0: Output disable of PWGA_TRGOUTn. 1: Output enable of PWGA_TRGOUTn. This register value is reflected to the PWGAnTCBR at the update timing selected by PWGAnTCUT[1:0].

NOTE

PWGAnTOE bit has to be set before count operation. In addition, in case of PWGAnTCUT=01B or 10B, PWGAnTOE bit has to be set before PWGAnTCBR rewrite request.

37.2.1.12 PWGAnCSBR — PWGAnCSDR Buffer Register

This is a buffer register for the PWGAnCSDR register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0024$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.32 PWGAnCSBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCSBR[11:0]	The PWGAnCSDR register value is reflected to this register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven high.

37.2.1.13 PWGAnCRBR — PWGAnCRDR Buffer Register

This is a buffer register for the PWGA_TOUTn reset condition.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0028_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRBR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.33 PWGAnCRBR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12 to 0	PWGAnCRBR[12:0]	The PWGAnCRDR register value is reflected to this register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven low.

37.2.1.14 PWGAnCTBR — PWGAnCTDR Buffer Register

This is a buffer register for the PWGA_TRGOUTn generation condition.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + $002C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.34 PWGAnCTBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCTBR[11:0]	The PWGAnCTDR register value is reflected to this register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, a trigger is transmitted to PWSAn.

37.2.1.15 PWGAnTCBR — PWGAnTCR Buffer Register

This is a buffer register for the PWGAnTCR register.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWGAn_base> + $001C_H$

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnTOBE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 37.35 PWGAnTCBR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	PWGAnTOBE	The PWGAnTCR register value is reflected to this register at the update timing selected by PWGAnCTRL.TCUT[1:0]. 0: Output disable of PWGA_TRGOUTn. 1: Output enable of PWGA_TRGOUTn.

37.2.1.16 PWGAnRSF — Buffer Register Reload Status Register

This register is a status register for simultaneous rewrite control.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWGAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWGAnRSFT	PWGAnRSF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.36 PWGAnRSF Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWGAnRSFT	PWGAnTCBR Rewrite Control Status 0: PWGAnTCBR rewrite is enabled. This value indicates the completion of PWGAnTCBR rewrite after the generation of a PWGAnTCBR rewrite trigger signal. In case of PWGAnTCUT = 00_B or 11_B, this bit shows always “0”. 1: PWGAnTCBR rewrite is in progress. This value indicates the waiting state for completion.
0	PWGAnRSF	Compare Buffer Register Simultaneous Rewrite Control Status 0: Compare buffer register simultaneous rewrite is enabled. This value indicates the completion of compare buffer register simultaneous rewrite after the generation of a compare buffer register simultaneous rewrite trigger signal. 1: Compare buffer register simultaneous rewrite is in progress. This value indicates the waiting state for completion.

37.2.1.17 PWGAnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWGAn_base> + $000C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWGAnRDTT	PWGAnRDT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 37.37 PWGAnRDT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWGAnRDTT	PWGAnTCBR Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the rewrite request for PWGAnTCBR, and sets PWGA0RSF.PWGA0RSFT to 1. In case of PWGAnTCUT = 00_B or 11_B, write to this bit “1” is invalid.
0	PWGAnRDT	Compare Buffer Register Simultaneous Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the simultaneous rewrite request for the compare registers (PWGAnCSDR, PWGAnCRDR, and PWGAnCTDR), and sets PWGAnRSF.PWGAnRSF to 1.

37.2.1.18 SLPWGAk — PWGA Synchronous Trigger Register (k = 0 to 2)

This register triggers start and stop for multiple channels simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <SLPW_base> + k × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SLPWGA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLPWGA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.38 SLPWGAk Register Contents

Bit Position	Bit Name	Function
31 to 0	SLPWGA[31:0]	Trigger start and stop to multiple channels simultaneously. 0: Stops the corresponding channels. 1: Starts the corresponding channels. The bits correspond to the following channels. SLPWGA0.SLPWGA[31:0]: PWGA31 - PWGA0 SLPWGA1.SLPWGA[31:0]: PWGA63 - PWGA32 SLPWGA2.SLPWGA[31:0]: PWGA95 - PWGA64

37.2.1.19 PWGAPRD — PWGA Period Setting Register

This register is used to set the PWGA counter period.

Access: This register can be read or written in 16-bit units.

Address: <SLPW_base> + $000C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAPRD[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.39 PWGAPRD Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAPRD[11:0]	These bits set the PWGA counter period. (= PWGA_PERIOD) The period of PWGA_TOUTn can be flexible changed by setting this register when PWGAPRDSLk is set to 1. If PWGAPRDSLk is set to 0, PWGA_PERIOD value is FFF_H.

NOTE

PWGAPRD register setting change is prohibited during count operation.

37.2.1.20 PWGAPRDSLk — PWGA Period Selection Register (k = 0 to 2)

This register is used to select the PWGA counter period either 12bit free-running or PWGAPRD register setting.

Access: This register can be read or written in 32-bit units.

Address: <SLPW_base> + 0010_H + k × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGAPRDSL[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGAPRDSL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.40 PWGAPRDSLk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGAPRDSL[31:0]	Select the PWGA counter period either 12 bit free-running or PWGAPRD register setting. 0: Select the 12 bit free-running 1: Select the PWGAPRD register setting The bits correspond to the following channels. PWGAPRDSL0.SLPWGA[31:0]: PWGA31 - PWGA0 PWGAPRDSL1.SLPWGA[31:0]: PWGA63 - PWGA32 PWGAPRDSL2.SLPWGA[31:0]: PWGA95 - PWGA64

37.2.1.21 PWSAnCTL — PWSA Control Register

This register is used to control operations of PWSA.

Access: This register can be read or written in 8-bit units.

Address: <PWSAn_base> + 0000$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	PWSAnARSE	—	—	—	—	—	—	PWSAnENBL
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W

Table 37.41 PWSAnCTL Register Contents

Bit Position	Bit Name	Function
7	PWSAnARSE	A/D conversion result storing control 0: Storing the ADCAnPWDDIR register to PWSAnPWDDIRz register is disabled 1: Storing the ADCAnPWDDIR register to PWSAnPWDDIRz register is enabled.
6 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnENBL	Operation Permission Control 0: Operation is prohibited (initial state). Writing 0 initializes PWSAnSTR, PWSAnQUEj, PWSAnOWE and PWSAnWFLG. 1: Operation is enabled.

NOTE

PWSAnARSE setting has to be changed during the state of trigger has not been input (PWSAnQUE0 = 7F$_H$) and condition of not input the trigger from PWGA.

37.2.1.22 PWSAnSTR — Trigger Queue Status Register

This is a status register that indicates whether the number of a channel for which an A/D conversion trigger has been generated is stored in a PWSAnQUEj register.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSAn_base> + 0004$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnQFL	PWSAnQNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.42 PWSAnSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWSAnQFL	Indicates the queuing state of the A/D conversion trigger. 0: Some PWSAnQUEj registers do not store a channel number. 1: All of the PWSAnQUEj registers store a channel number.
0	PWSAnQNE	Bit indicating that there is a trigger in the trigger queue 0: A channel number is not stored in a PWSAnQUEj register, or A/D conversion is in progress while only PWSAnQUE0 stores a channel number. 1: The number of the channel waiting for conversion is stored in j = 1 and subsequent PWSAnQUEj registers.

37.2.1.23 PWSAnSTC — Trigger Queue Status Clear Register

This register clears the status of the PWSAnSTR register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWSAn_base> + 0008$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnCLFL	PWSAnCLNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 37.43 PWSAnSTC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWSAnCLFL	PWSAnQFL Clear 0: PWSAnQFL retains the status (Writing 0 has no effect). 1: PWSAnQFL is cleared to 0.
0	PWSAnCLNE	PWSAnQNE Clear 0: PWSAnQNE retains the status (Writing 0 has no effect). 1: PWSAnQNE is cleared to 0.

37.2.1.24 PWSAnQUEj — Trigger Queue Register (j = 0 to 7)

This register stores the channel number that received the trigger from PWGAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSAn_base> + 0020_H + j × 4_H

Value after reset: $7F_H$

Bit	7	6	5	4	3	2	1	0
	—	PWSAnQUEj[6:0]						
Value after reset	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R

Table 37.44 PWSAnQUEj Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	PWSAnQUEj[6:0]	These bits hold the channel number (0 to 95) of the PWGA for which a trigger was generated in order from PWSAnQUE0 to PWSAnQUE7. After the A/D conversion of PWSAnQUE0 is completed, the values in PWSAnQUE1 to PWSAnQUE7 shift to PWSAnQUE0 to PWSAnQUE6.

NOTE

If a trigger occurs simultaneously for multiple channels, the trigger with the smaller channel number has priority.

37.2.1.25 PWSAnPVCRx_y — PWM-Diag Mode A/D Setting Register

This register is used to set the corresponding A/D converter for each channel.

Two consecutive channels are set such as PWSA0PVCR02_03, and the 16 higher-order bits of each register correspond to an odd-numbered channel while the 16 lower-order bits correspond to an even- numbered channel.

At the generation of a trigger, the set value is transmitted to the ADCAnPWDVCR register of the A/D converter.

For the ADCAnPWDVCR register, see **Section 38, A/D Converter (ADCA)**.

Access:	PWSAnPVCRx_y register can be read or written in 32-bit units. PWSAnPVCRx_yL, PWSAnPVCRx_yH registers can be read or written in 16-bit units.
Address:	PWSAnPVCRx_y: <PWSAn_base> + 0040_H + x × 2_H PWSAnPVCRx_yL: <PWSAn_base> + 0040_H + x × 2_H PWSAnPVCRx_yH: <PWSAn_base> + 0040_H + x × 2_H + 2_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PWSAnVRDTy [29:28]		PWSAn SLADy	PWSAn VRDTy [27]	PWSAnVRDTy[26:24]			PWSAnVRDTy [23:22]		PWSAnVRDTy[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PWSAnVRDTx [13:12]		PWSAn SLADx	PWSAn VRDTx [11]	PWSAnVRDTx[10:8]			PWSAnVRDTx [7:6]		PWSAnVRDTx[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.45 PWSAnPVCRx_y Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30, 29	PWSAnVRDTy[29:28]	These bits indicate the set value of the ADCAnPWDVCR.ULS[3:2] bits (odd-numbered channel).
28	PWSAnSLADy	RH850/F1KM-S4 100-pin products: RH850/F1KM-S1 all products: When writing, write the value after reset. RH850/F1KH-D8 all products, RH850/F1KM-S4 144/176/233/272-pin products: ADCA Select (odd-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
27	PWSAnVRDTy[27]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit[*1] (odd-numbered channel).
26 to 24	PWSAnVRDTy[26:24]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits[*1] (odd-numbered channel).
23, 22	PWSAnVRDTy[23:22]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits (odd-numbered channel).
21 to 16	PWSAnVRDTy[21:16]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits (odd-numbered channel).
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14, 13	PWSAnVRDTx[13:12]	These bits indicate the set value of the ADCAnPWDVCR.ULS[3:2] bits. (even-numbered channel)

Table 37.45 PWSAnPVCRx_y Register Contents

Bit Position	Bit Name	Function
12	PWSAnSLADx	RH850/F1KM-S4 100-pin products: RH850/F1KM-S1 all products: When writing, write the value after reset. RH850/F1KH-D8 all products, RH850/F1KM-S4 144/176/233/272-pin products: ADCA Select (even-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
11	PWSAnVRDTx[11]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit*[1]. (even-numbered channel)
10 to 8	PWSAnVRDTx[10:8]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits*[1]. (even-numbered channel)
7, 6	PWSAnVRDTx[7:6]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits. (even-numbered channel)
5 to 0	PWSAnVRDTx[5:0]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits. (even-numbered channel)

Note 1. These bits are only supported for ADCA0.

37.2.1.26 PWSAnPWDDIRz — PWM-Diag Data Supplementary Information Register

This register stores the A/D conversion result correspond to PWSAnPVCRx_y register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PWSAn_base> + 0200_H + z × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWSAn MPXE	PWSAnMPXV[2:0]			—	PWSAn OWE	PWSAn WFLG	—	—	—	PWSAnID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWSAnPWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.46 PWSAnPWDDIRz Register Contents

Bit Position	Bit Name	Function
31	PWSAnMPXE	This bit mirrors PWSAnVRDTx[11] or PWSAnVRDTy[27]
30 to 28	PWSAnMPXV[2:0]	These bits mirror PWSAnVRDTx[10:8] or PWSAnVRDTy[26:24]
27	Reserved	When read, the value after reset is returned.
26	PWSAnOWE	This bit indicates overwrite error flag of the A/D conversion result storing. Set timing is same as PWSAnPWDDR stored timing at the condition of PWSAnWFLG = 1. 0: An overwrite error is not detected. 1: An overwrite error is detected Setting condition: PWSAnWFLG = 1, and new A/D conversion result is written to PWSAnPWDDR[15:0] Clearing condition: Reading this register, or clearing the PWSAnENBL bit.
25	PWSAnWFLG	This bit indicates write flag of the A/D conversion result storing. Set timing is same as PWSAnPWDDR stored timing. 0: This register is read, or A/D conversion is not finished. 1: The A/D conversion result is stored (not read yet). Setting condition: The A/D conversion result is written to PWSAnPWDDR[15:0] Clearing condition: Reading this register, or clearing the PWSAnENBL bit.
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	PWSAnID[5:0]	These bits mirror PWSAnVRDTx[5:0] or PWSAnVRDTy[21:16]
15 to 0	PWSAnPWDDR[15:0]	These bits indicate the A/D conversion result which is sent by ADC_PWDDRn[15:0] pins from ADCAn. The data format is same as ADCAnPWDTSNDR.PWDDR[15:0].

37.2.1.27 PWSAnEMU — PWSA Emulation Control Register

This register is used to set the operation for emulation.

Access: This register can be read or written in 8-bit units.

Address: <PWSAn_base> + 000C$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWSAnSVSDIS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.47 PWSAnEMU Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnSVSDIS	(When the EPC.SVSTOP bit = 0) The operation continues when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: When the debugger is controlling the microcontroller (by using break points, etc.); – The output state to A/D is retained, the ADC_CONV_ENDn input at a break point is internally retained, and PWSAnQUEj is updated after break release. – The PWGA_TRGOUTn input is accepted even at a break, and INTQFULL is also output. – Reading and writing to the register is possible. 1: The operation continues when the debugger is controlling the microcontroller (by using break points, etc.). The above bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0), the operation of all channels PWGAn has stopped (SLPWGAk.SLPWGA), and no trigger has been generated from any of the channels PWGAn (PWSAnQUE0 is the value after reset).

37.2.1.28 PWGAINTFhk — PWGA Interrupt Factor Register (k = 0 to 2)

These registers contain information about which PWGAn interrupt (PWGA_INTn) has been generated without depending on PWGAINTMSKhk setting. Regarding block diagram image, see **Figure 37.1, PWGA Interrupt Connection Image**.

Access:	These registers are read-only registers that can be read in 32-bit units.
Address:	<PWGAINTF_base> + h × 30_H + k × 10_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGAINTFhk[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGAINTFhk[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.48 PWGAINTFhk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGAINTFhk [31:0]	PWGA_INTn interrupt occurrence 0: No interrupt occurred 1: interrupt has occurred The bits correspond to the following channels. PWGAINTFh0[31:0]: PWGA31 - PWGA0 PWGAINTFh1[31:0]: PWGA63 - PWGA32 PWGAINTFh2[31:0]: PWGA95 - PWGA64

37.2.1.29 PWGAINTMSKhk — PWGA Interrupt Mask Register (k = 0 to 2)

These registers mask PWGAn interrupt output to INTC2 by each channels. Regarding block diagram image, see **Figure 37.1, PWGA Interrupt Connection Image**.

Access: These registers can be read or written in 32-bit units.

Address: <PWGAINTF_base> + h × 30_H + k × 10_H + 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGAINTMSKhk[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGAINTMSKhk[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.49 PWGAINTMSKhk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGAINTMSKhk [31:0]	PWGA interrupt output to INTC2 mask 0: Not masked 1: Masked The bits correspond to the following channels. PWGAINTMSKh0[31:0]: PWGA31 - PWGA0 PWGAINTMSKh1[31:0]: PWGA63 - PWGA32 PWGAINTMSKh2[31:0]: PWGA95 - PWGA64

37.2.1.30 PWGAINTFChk — PWGA Interrupt Factor Clear Register (k = 0 to 2)

These registers clear the bits of PWGA interrupt factor register (PWGAINTFhk). Regarding block diagram image, see **Figure 37.1, PWGA Interrupt Connection Image**.

Access:	These registers are write-only registers that can be written in 32-bit units.
Address:	<PWGAINTF_base> + h × 30_H + k × 10_H + 8_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGAINTFChk[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGAINTFChk[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 37.50 PWGAINTFChk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGAINTFChk [31:0]	PWGAINTFhk[31:0] flag clear 0: — 1: Clear The bits correspond to the following channels. PWGAINTFCh0[31:0]: PWGA31 - PWGA0 PWGAINTFCh1[31:0]: PWGA63 - PWGA32 PWGAINTFCh2[31:0]: PWGA95 - PWGA64

37.3 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.

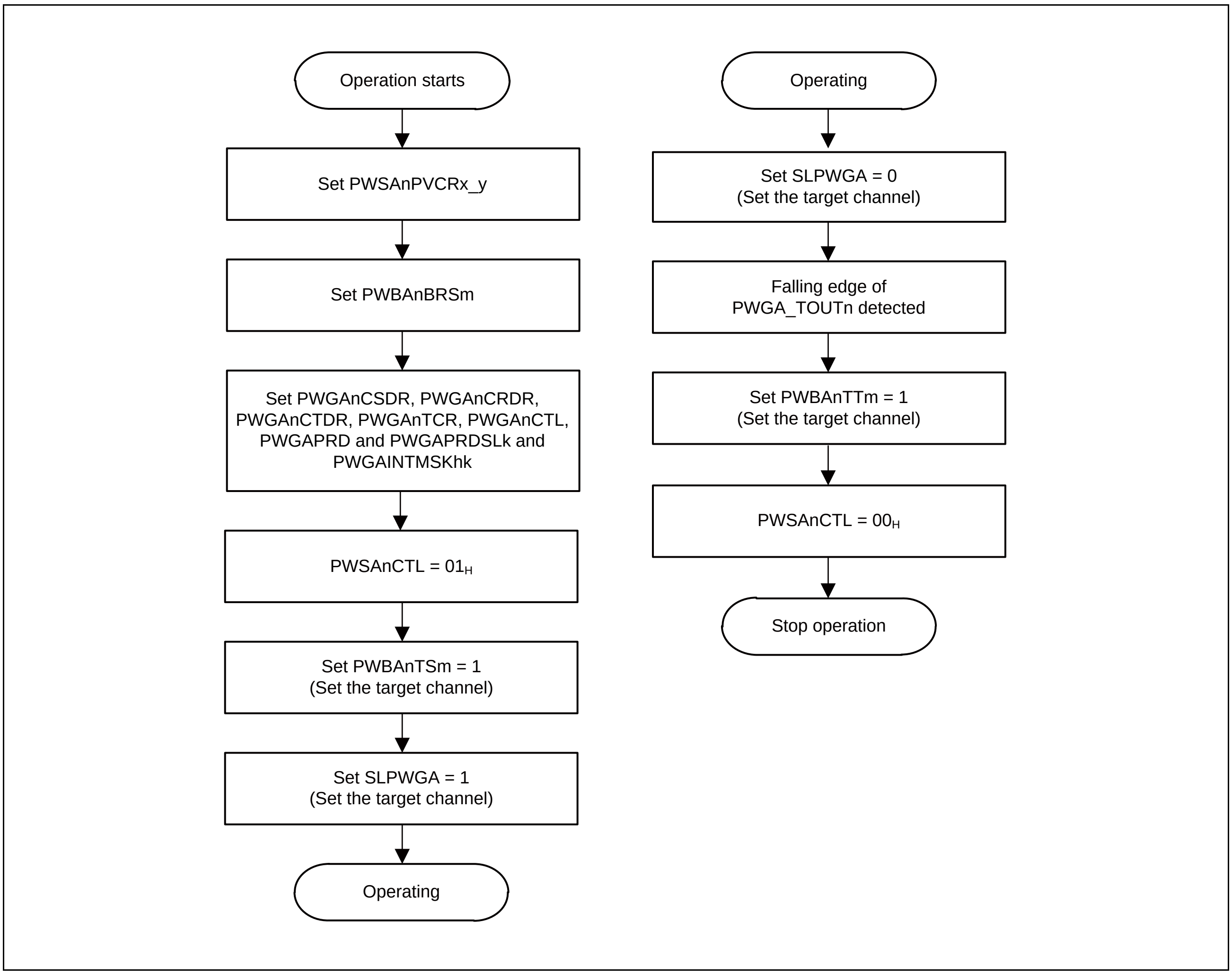
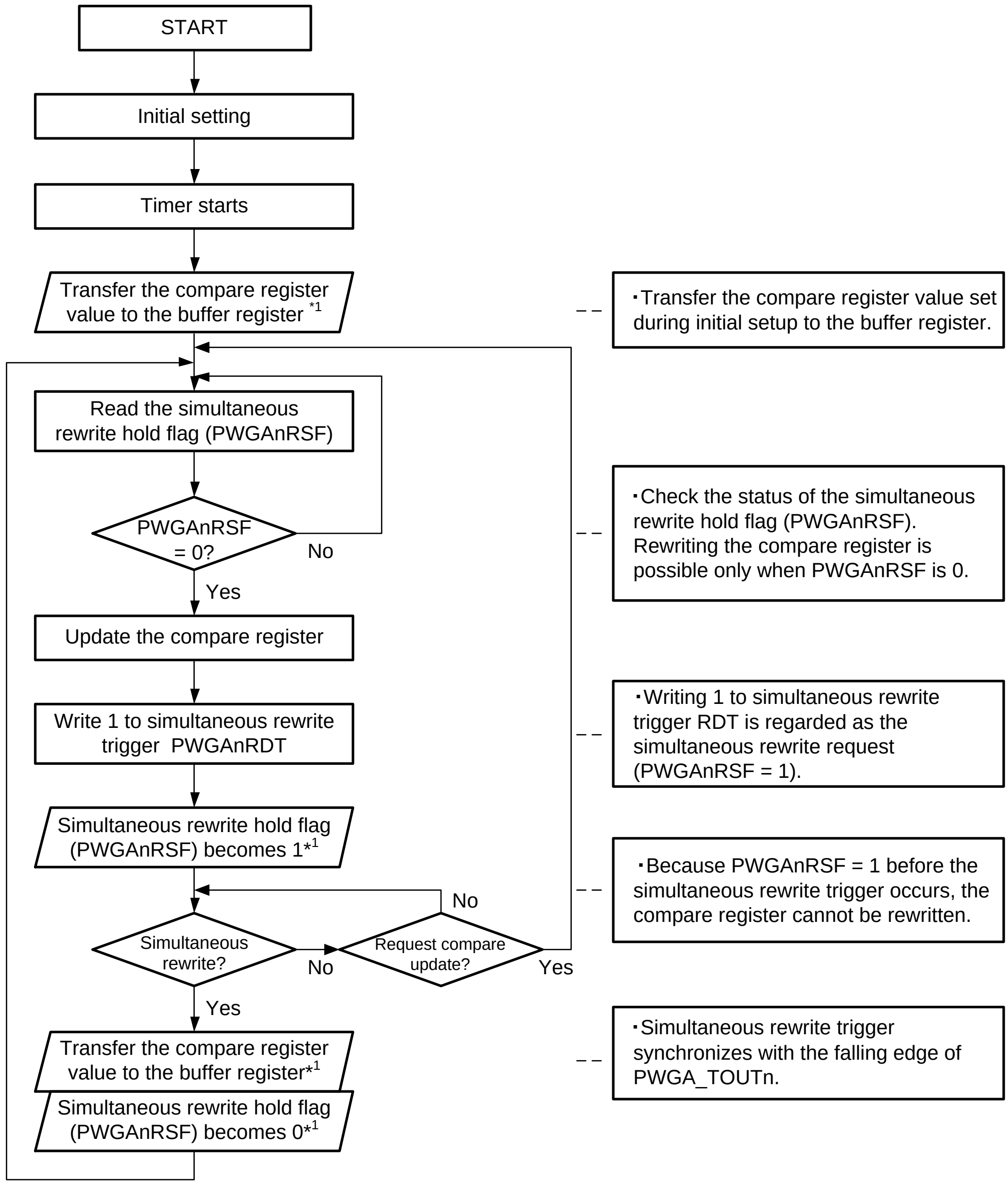


Figure 37.3 PWM-Diag Operating Procedure

Procedures for compare buffer register simultaneous rewrite of PWGA are illustrated below.

The described term “compare register” indicates PWGAnCSDR, PWGAnCRDR, or PWGAnCTDR.

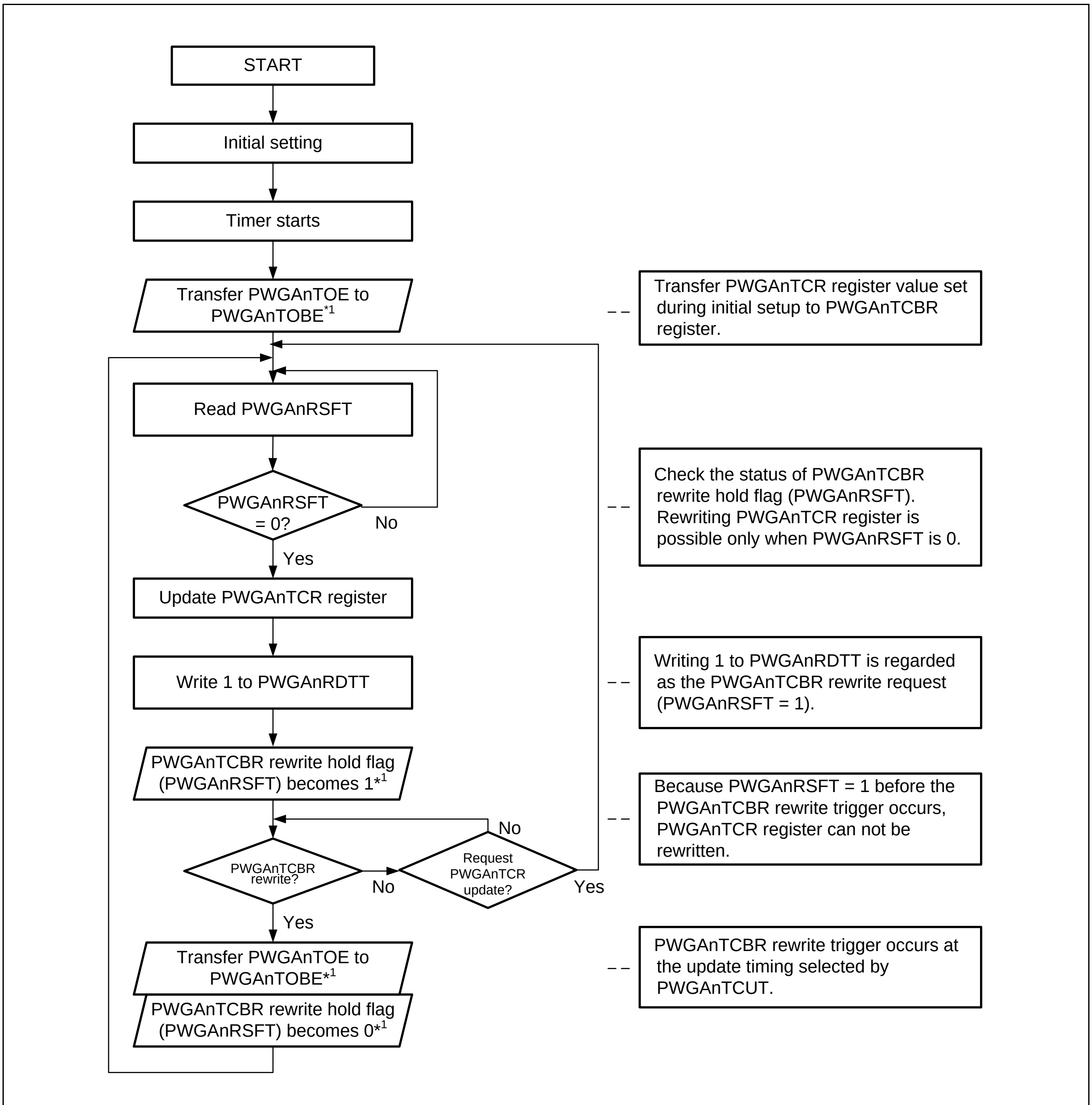
In addition, the described term “buffer register” indicates PWGAnCSBR, PWGAnCRBR, or PWGAnCTBR.



Note 1. Indicates the transition by hardware.

Figure 37.4 Compare Buffer Register Simultaneous Rewrite Procedure

Procedures for PWGAnTCBR register rewrite of PWGA are illustrated below.



Note 1. Indicates the transition by hardware.

Figure 37.5 PWGAnTCBR Register Rewrite Procedure

37.4 Operation Waveform of PWM-Diag

37.4.1 PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output

37.4.1.1 Basic Operation Waveform of PWGA

The basic operation waveforms of PWGA are illustrated below.

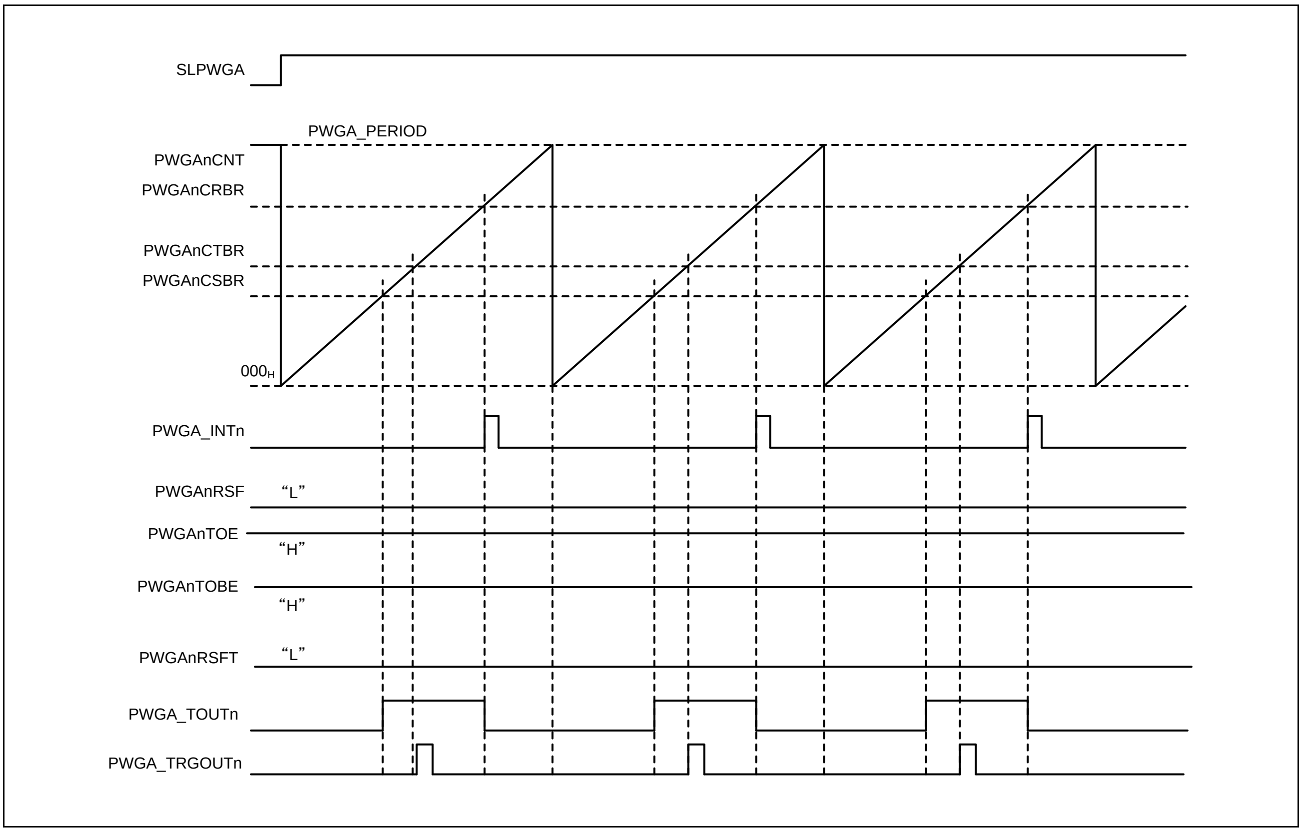


Figure 37.6 Basic Waveform

37.4.1.2 Operation Waveform when Compare Buffer Register Simultaneous Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when compare buffer register simultaneous rewrite for PWGA is executed.

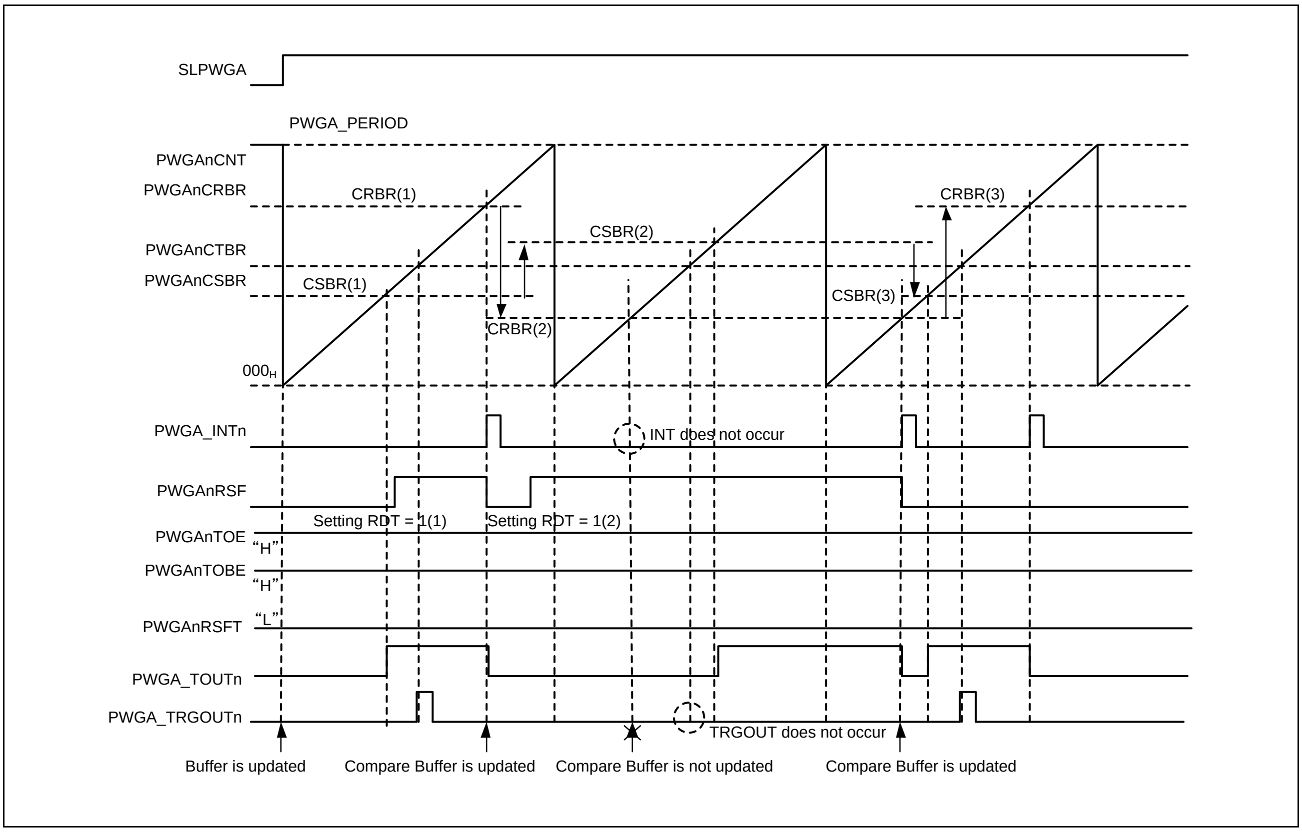


Figure 37.7 Waveform when Compare Buffer Register Simultaneous Rewrite is Executed (PWGAnOCL = 0)

Compare buffer register simultaneous rewrite is executed by re-setting the PWGAnCSDR and PWGAnCRDR registers, then setting either the PWGAnRDT or SLPWGAk register.

Moreover, if the relationship between set values in one interval is PWGAnCSDR > PWGAnCRDR, a falling edge in that interval is meaningless, and the falling edge in the next interval is valid.

In case of PWGAnOCL = 1, PWGAn_TRGOUTn is also occurred at the timing of "TRGOUT does not occur" in **Figure 37.7, Waveform when Compare Buffer Register Simultaneous Rewrite is Executed (PWGAnOCL = 0)**.

37.4.1.3 Operation Waveform when PWGAnTCBR Register Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when PWGAnTCBR register rewrite for PWGA is executed.

PWGAnTCBR register rewrite is executed by re-setting the PWGAnTCR register after setting of PWGAnRDTT.

The update timing of PWGAnTCBR register requested by PWGAnRDTT is depended on PWGAnTCUT setting.

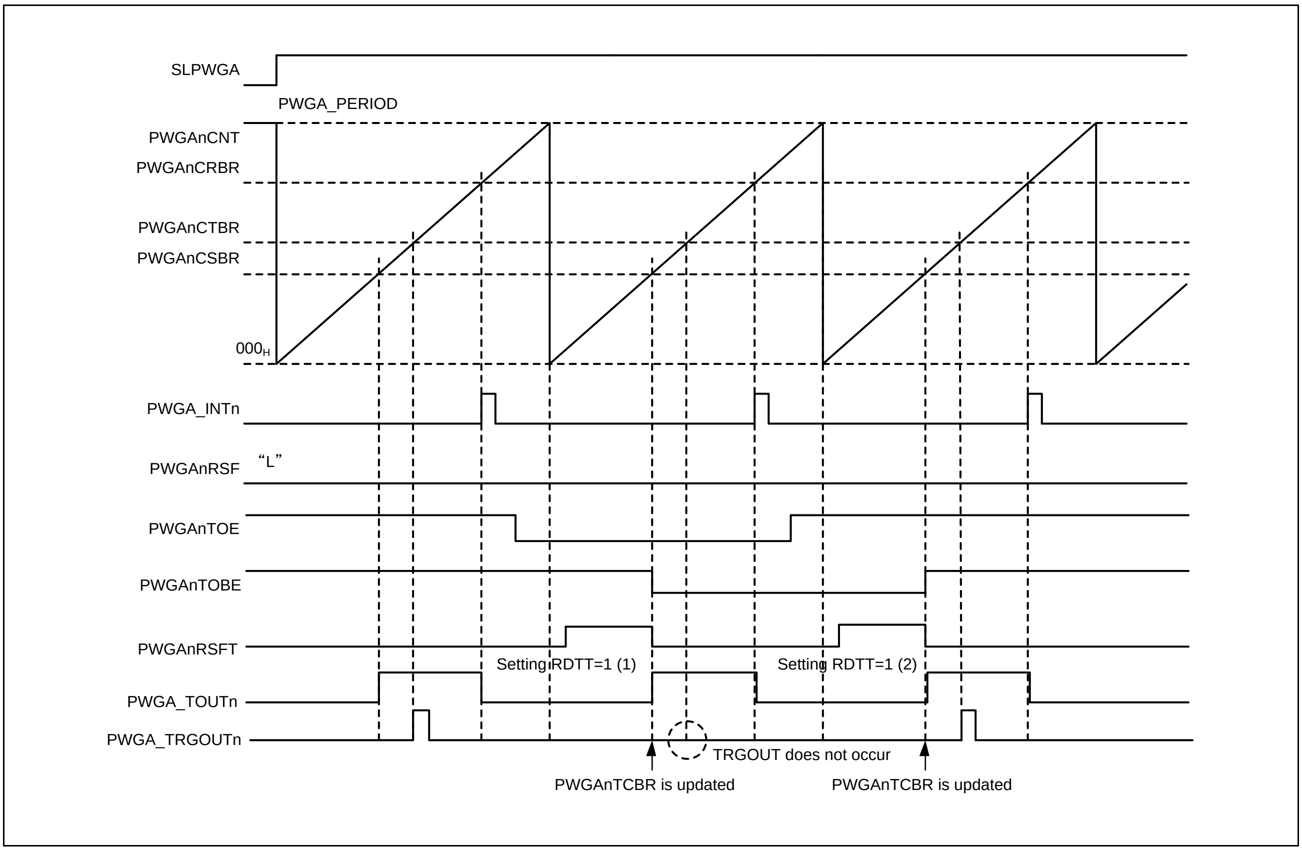


Figure 37.8 Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 01_B)

PWGAn_TRGOUTn is not occurred at the timing of "TRGOUT does not occur" in **Figure 37.8, Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 01B)** because PWGAnTOBE = 0 even if PWGAnOCL = 1.

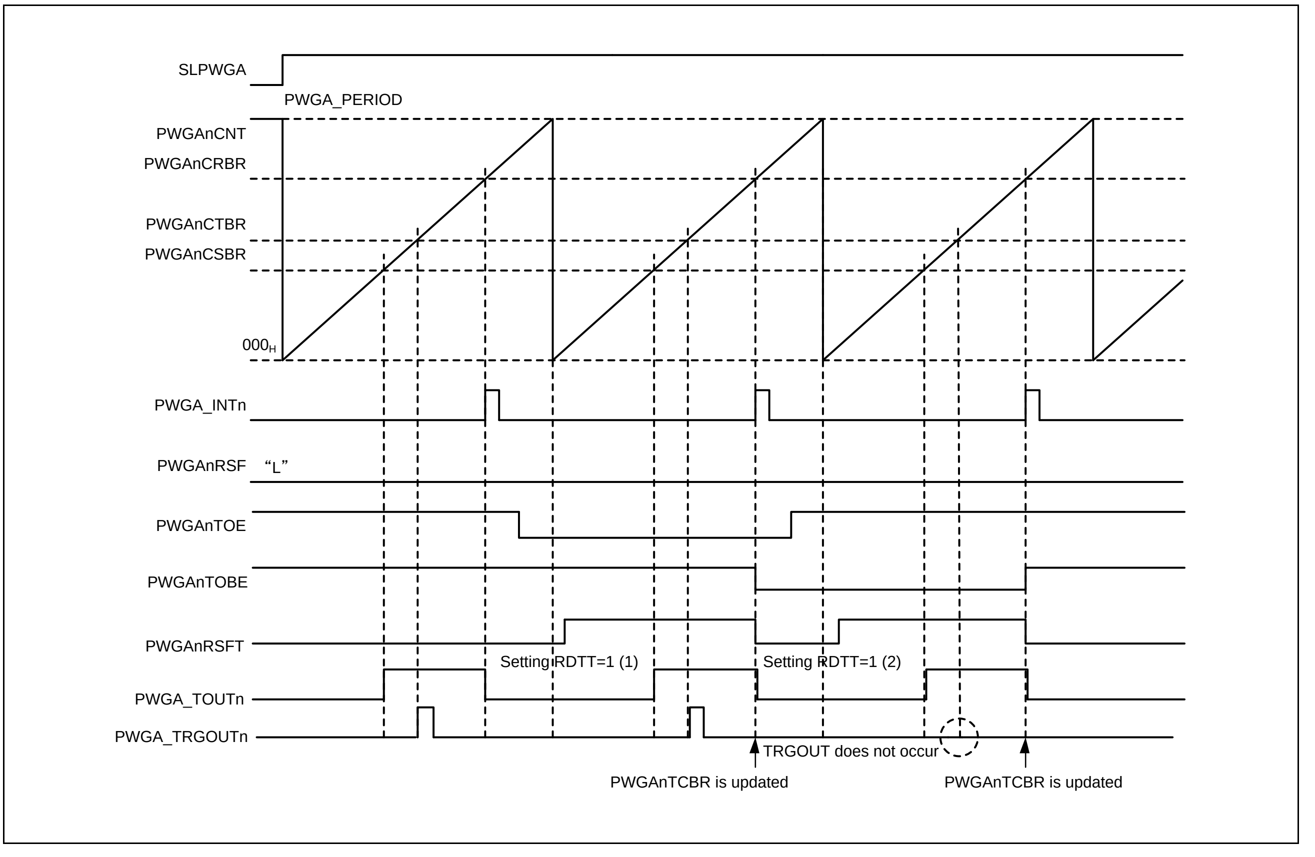


Figure 37.9 Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 10_B)

PWGAn_TRGOUTn is not occurred at the timing of “TRGOUT does not occur” in **Figure 37.9, Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 10B)** because PWGAnTOBE = 0 even if PWGAnOCL = 1.

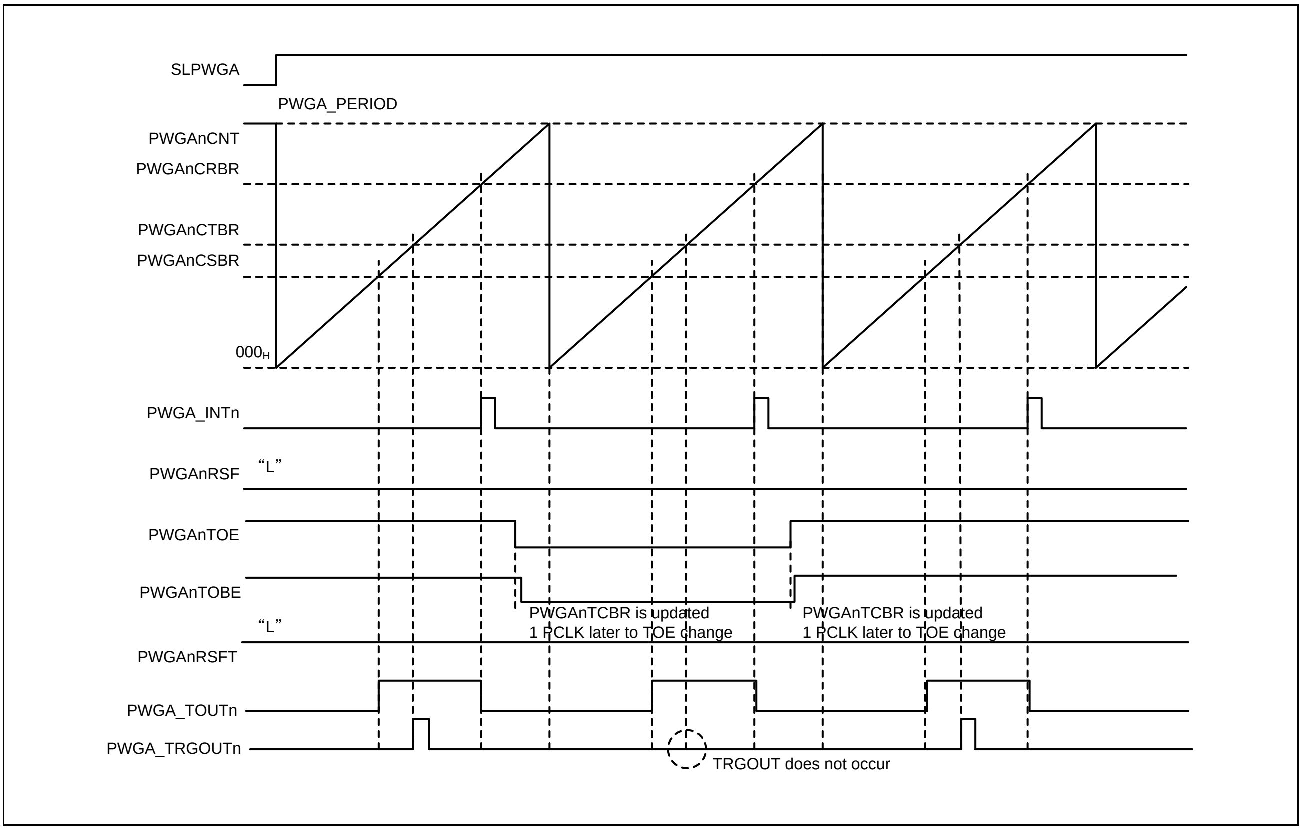


Figure 37.10 Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 11_B)

PWGAn_TRGOUTn is not occurred at the timing of "TRGOUT does not occur" in **Figure 37.10, Waveform when PWGAnTCBR Register Rewrite is Executed (PWGAnOCL = 0, PWGAnTCUT = 11B)** because PWGAnTOBE = 0 even if PWGAnOCL = 1.

37.4.1.4 Operation Waveform when Stopping and Restarting PWGA Operation

The following figure illustrates the operation waveforms when stopping and restarting PWGA operation.

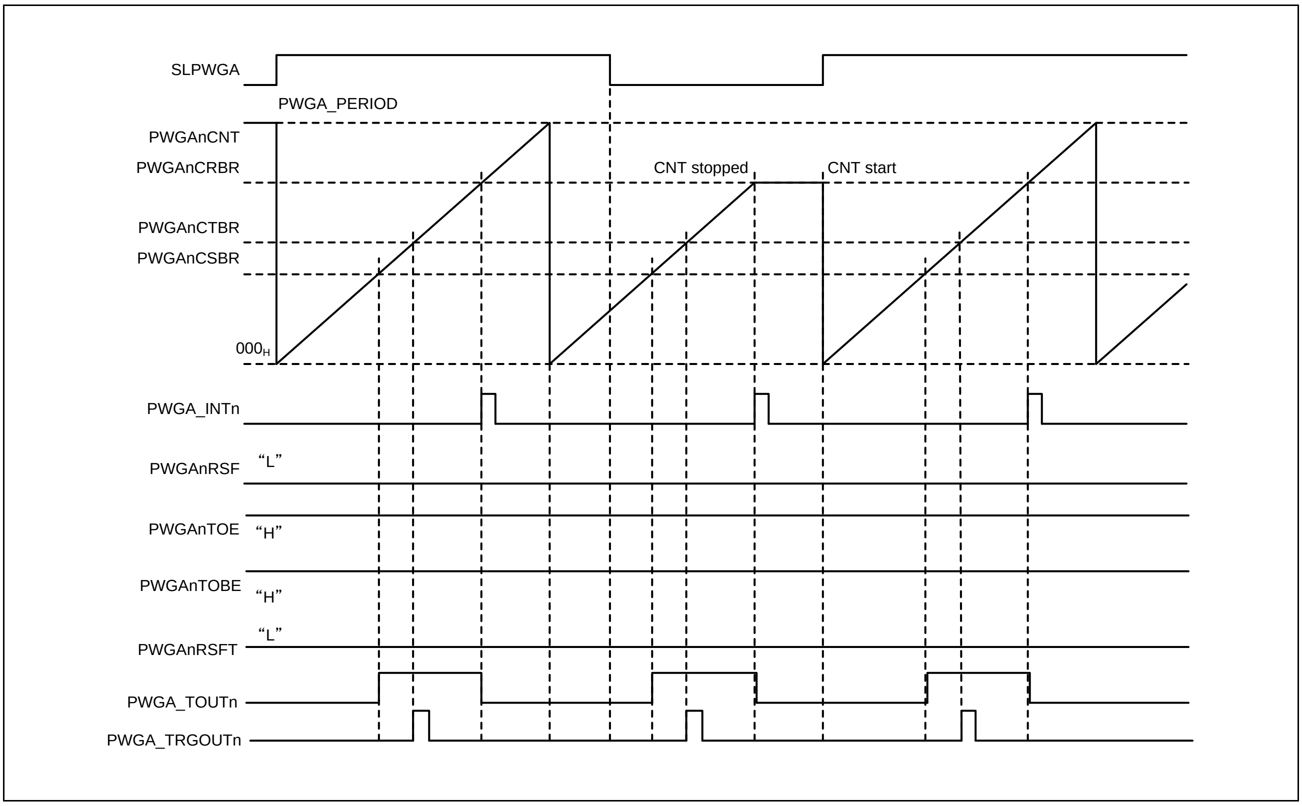


Figure 37.11 Stopping and Resuming Operation (1)

After the setting of SLPWGA has been changed from 1 to 0, PWGAnCNT stops operation because PWGA_INTn is generated.

After PWGA_INTn has been generated, by changing the setting of SLPWGA from 0 to 1, PWGAnCNT resumes counting from 000_H.

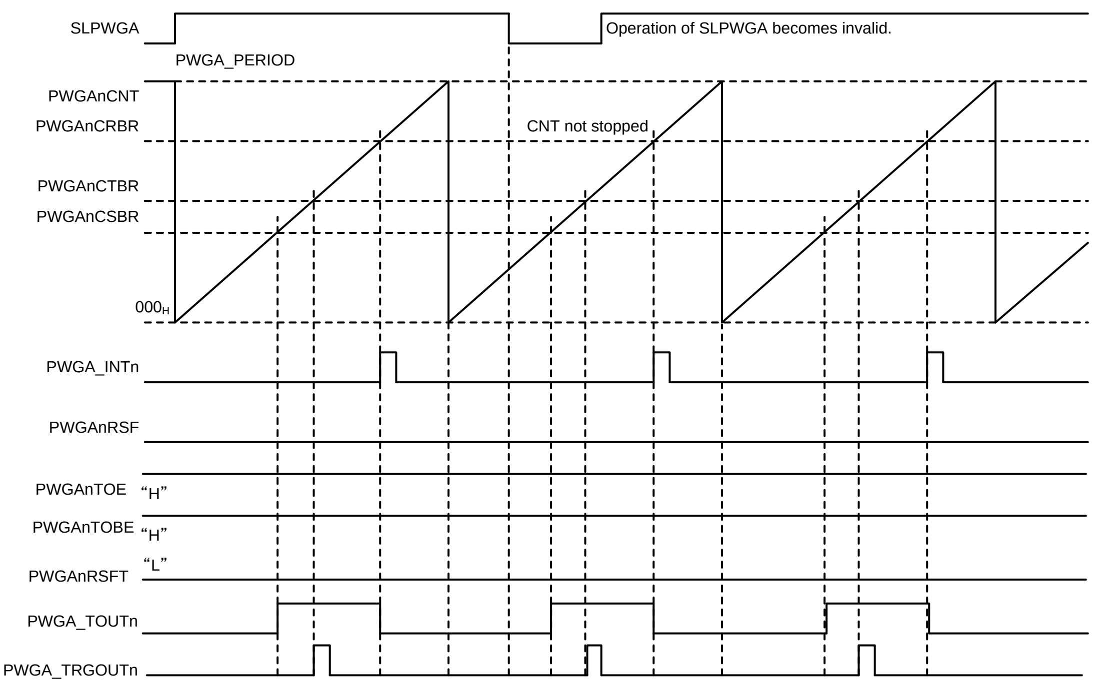


Figure 37.12 Stopping and Resuming Operation (2)

After the setting of SLPWGA has been changed from 1 to 0, if the setting of SLPWGA is changed from 0 to 1 before PWGA_INTn is generated, operations of SLPWGA become invalid, and PWGAnCNT continues counting.

37.4.1.5 Waveforms of PWGA Operation with Specific Settings

The following figures illustrate the waveforms of PWGA operation with specific settings.

When PWGAnOCL is set to 0 and PWGAnCSBR is equal to PWGAnCRBR, PWGA_TOUTn is fixed to the low level and PWGA_TRGOUTn does not become valid as illustrated.

When PWGAnOCL is set to 1, PWGA_TRGOUTn occurs at the timing PWGAnCNT matches PWGAnCTBR regardless of PWGA_TOUTn level.

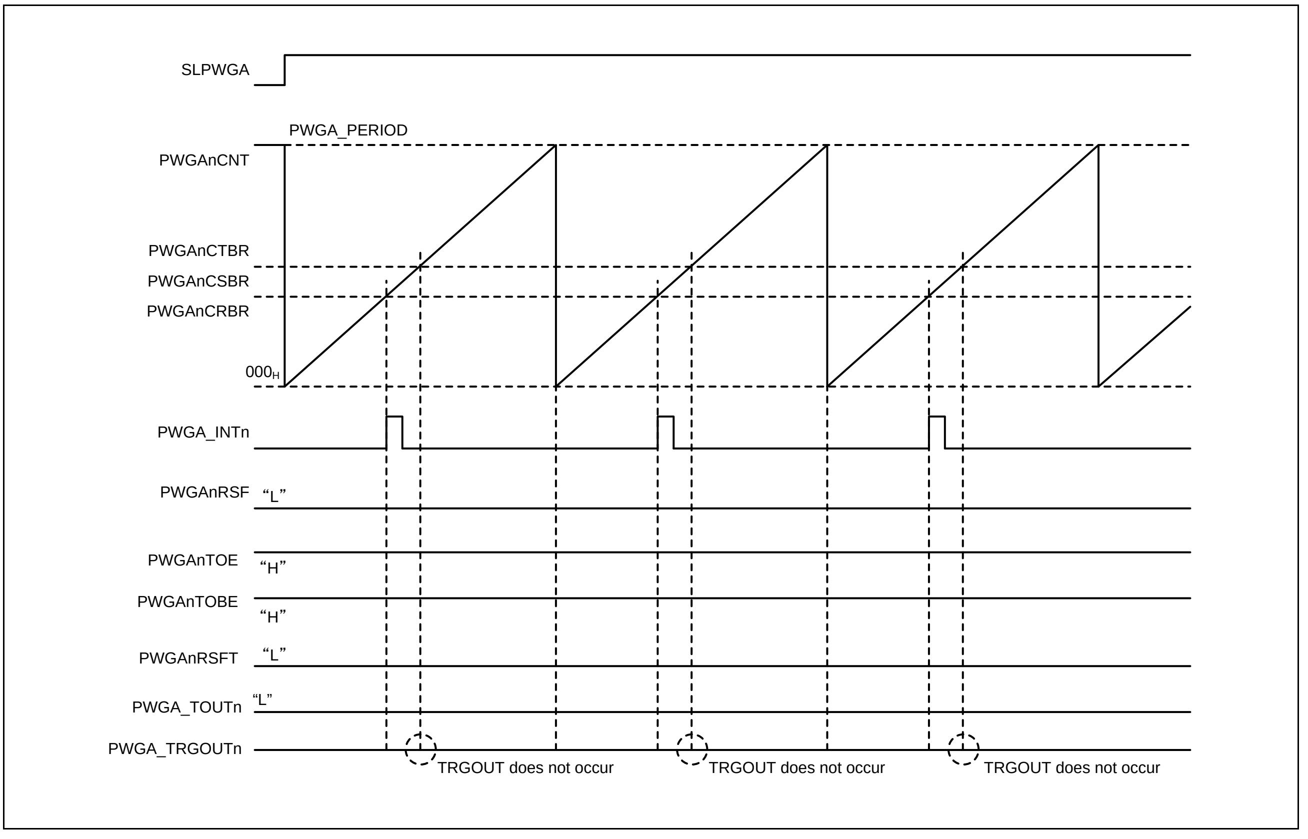


Figure 37.13 PWGA_TOUTn = 0% Output Waveform

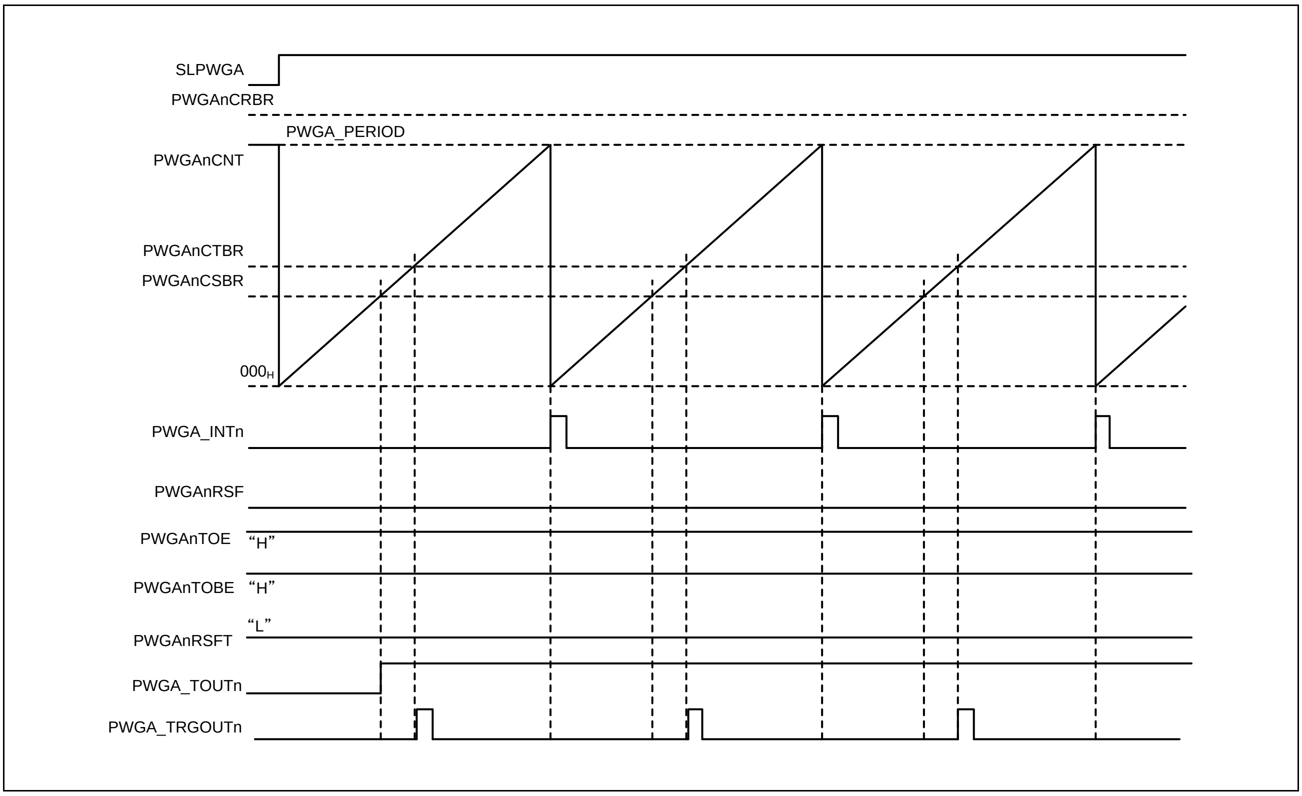
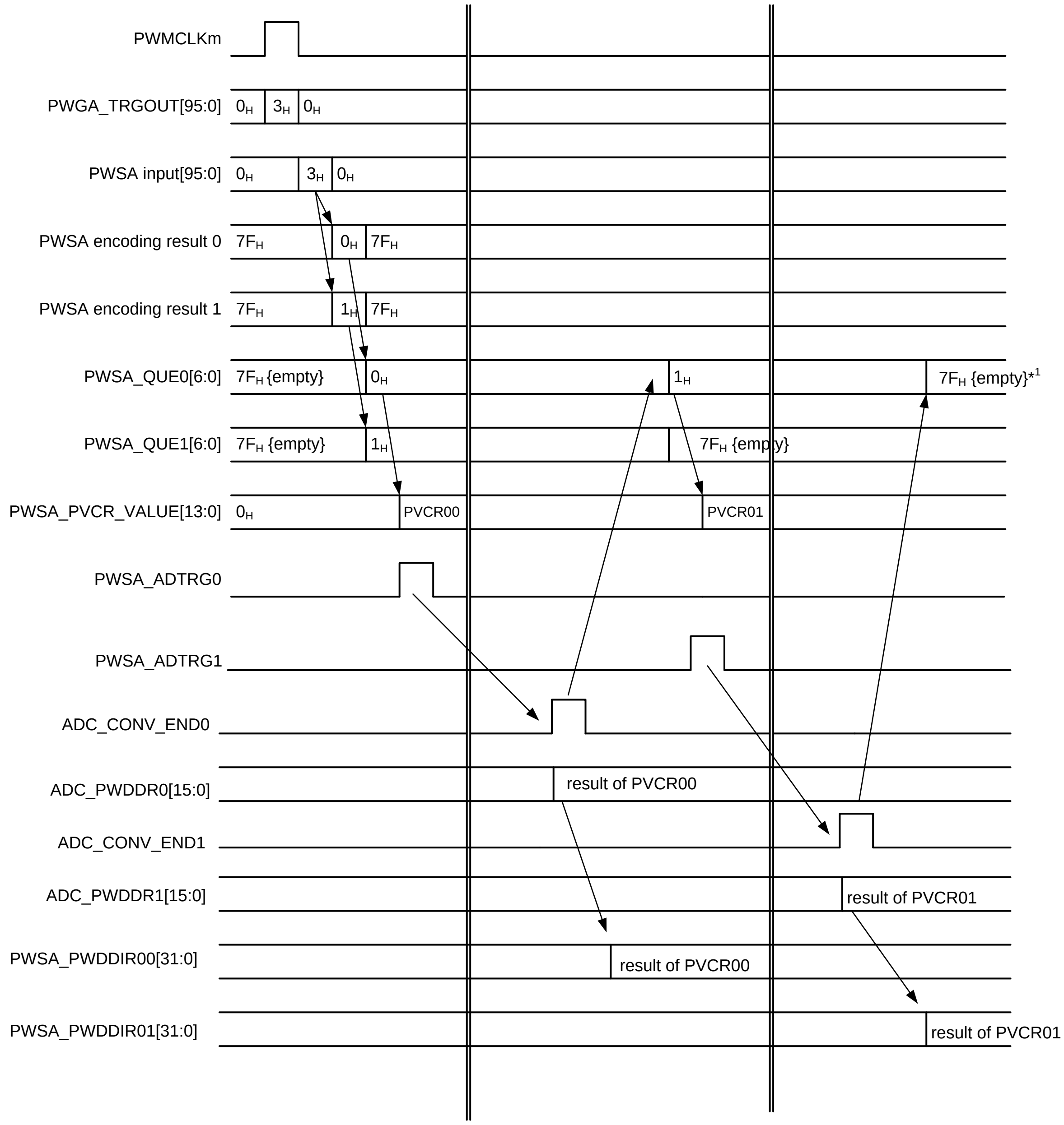


Figure 37.14 PWGA_TOUTn = 100% Output Waveform

37.4.2 Operation Waveform when A/D Conversion Trigger Occurs in PWSA

An example of the PWSA operation is shown below.



Note 1. QUE0 indicates the PWGA_TRG number while the A/D conversion is executed. The initial value (while the A/D conversion is not executed) is $7F_H$.

Figure 37.15 Example of PWSA Operation

(1) Triggers occur simultaneously in channels 0 and 1 of PWGA. Channel 0 with the smaller channel number is stored in PWSAnQUE0, and channel 1 with the larger channel number is stored in PWSAnQUE1. The lower 16 bits data of PWSAnPVCR00_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.
The A/D converter whether ADCA0 or ADCA1 is selected by PWSAnSLAD00 bit of PWSAnPVCR00_01. This example shows that ADCA0 is selected when PWSAnSLAD00 is 0.
At this time, as the A/D conversion for channel 1 is in the waiting state, the PWSAnSTR.PWSAnQNE bit is set.

(2) On completion of A/D conversion executed in step (1), the result of A/D conversion is stored in the lower 16 bits of PWSAnPWDDIR00. The higher 16 bits of PWSAnPWDDIR00 mirrors the information of PWSAnPVCR00_01[15:0]. The channel number of PWSAnQUE1 shifts to PWSAnQUE0 and PWSAnQUE1 enters the empty state.
After that, as similar to step (1), the upper 16-bit data of PWSAnPVCR00_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter. This example shows that ADCA1 is selected when PWSAnSLAD01 is 1.

(3) On completion of A/D conversion executed in step (2), the result of A/D conversion is stored in the lower 16 bits of PWSAnPWDDIR01. The higher 16 bits of PWSAnPWDDIR01 mirrors the information of PWSAnPVCR00_01[31:16]. PWSAnQUE0 enters the empty state.

37.5 PWM-Diag Related Function in A/D Converter (ADCA)

This section describes the A/D converter used in the PWM-Diag function.

37.6 ADCA Registers when the PWM-Diag Function is Used

- Before starting PWSA operation, the A/D converter must be set using the following register.
  - PWM-Diag scan group control register (ADCAnPWDSGCR)
- When the PWM-Diag is running, the PWSAnPVCRx_y value corresponding to the channel under conversion is set in the following register of the A/D converter.
  - PWM-Diag virtual channel register (ADCAnPWDVCR)
- After completion of A/D conversion, the conversion result can be checked by reading the following registers.
  - PWM-Diag data register (ADCAnPWDTSNDR)
  - PWM-Diag data supplementary information register (ADCAnPWDDIR, PWSAnPWDDIRz*[1])
- When A/D conversion result is outside the expected range, it can be confirmed using the upper/lower limit error detection function. The upper/lower limit error detection function is set by the following register.
  - Upper limit/lower limit error register (ADCAnULER)
- The scan end flag of the PWM-Diag scan group can be cleared using the following register.
  - PWM-Diag scan end flag clear register (ADCAnPWDSGSEFCR)

Note 1. PWSAnPWDDIRz register is in PWSA.

Section 38 A/D Converter (ADCA)

This section contains a generic description of the A/D Converter (ADCA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of ADCA.

38.1 Features of RH850/F1KH, RH850/F1KM ADCA

38.1.1 Number of Units and Channels

This microcontroller has the following number of ADCA units.

Table 38.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	2	2	2
Name	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)

Table 38.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	2	2	2	2
Name	ADCAn (n = 0)	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)

Table 38.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)

An ADCAn unit has the same number of physical channels as the number of A/D input pins and the same number of virtual channels as the number of addresses where the results of A/D conversion will be stored. The numbers of channels on individual products are as listed below.

Table 38.4 Unit Configurations and Physical Channels (RH850/F1KH-D8)

Unit Name ADCAn		RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
ADCA0	12-bit pins for conversion*[1]	16	16	16
	10-bit pins for conversion*[2]	18	18	18
ADCA1	12-bit pins for conversion*[1]	16	16	16
	10-bit pins for conversion*[2]	8	20	20

Note 1. When 10-bit mode is selected, this pin can be used for 10-bit conversion.

Note 2. When 12-bit mode is selected but a pin is for 10-bit conversion, the 2 low-order bits of the result of conversion must be masked before use.

Table 38.5 Unit Configurations and Physical Channels (RH850/F1KM-S4)

Unit Name ADCAn		RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
ADCA0	12-bit pins for conversion*[1]	16	16	16	16	16
	10-bit pins for conversion*[2]	16	18	18	18	18
ADCA1	12-bit pins for conversion*[1]	—	8	16	16	16
	10-bit pins for conversion*[2]	—	4	8	20	20

Note 1. When 10-bit mode is selected, this pin can be used for 10-bit conversion.

Note 2. When 12-bit mode is selected but a pin is for 10-bit conversion, the 2 low-order bits of the result of conversion must be masked before use.

Table 38.6 Unit Configurations and Physical Channels (RH850/F1KM-S1)

Unit Name ADCAn		RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
ADCA0	12-bit pins for conversion*[1]	8	10	11	16
	10-bit pins for conversion*[2]	4	11	14	20

Note 1. When 10-bit mode is selected, this pin can be used for 10-bit conversion.

Note 2. When 12-bit mode is selected but a pin is for 10-bit conversion, the 2 low-order bits of the result of conversion must be masked before use.

Table 38.7 Unit Configurations and Virtual Channels (RH850/F1KH-D8)

Unit Name ADCAn	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
ADCA0	50	50	50
ADCA1	36	36	36

Table 38.8 Unit Configurations and Virtual Channels (RH850/F1KM-S4)

Unit Name ADCAn	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
ADCA0	50	50	50	50	50
ADCA1	—	36	36	36	36

Table 38.9 Unit Configurations and Virtual Channels (RH850/F1KM-S1)

Unit Name ADCAn	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
ADCA0	20	29	37	50

Table 38.10 Indices (RH850/F1KH-D8)

Index	Description
n	Throughout this section, the individual ADCA units are identified by the index “n”; for example, ADCAnPWDVCR (n = 0, 1) indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels (channels in the unit) of ADCAn are identified by the index “m”; for example, ANInm.
j	Throughout this section, the individual virtual channels of ADCAn are identified by the index “j”; for example, ADCAnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCAn are identified by the index “x”; for example, ADCAnSGSTCRx indicates the scan group x start control register (x = 1 to 3).
k	Throughout this section, the individual physical channel numbers for T&H are identified by the index “k”; for example, THkE (k = 0 to 5) is T&H enable bit of the T&H enable register (ADCAnTHER).

Table 38.11 Indices (RH850/F1KM-S4)

Index	Description
n	Throughout this section, the individual ADCA units are identified by the index “n”; for example, ADCAnPWDVCR (n = 0, 1) indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels (channels in the unit) of ADCAn are identified by the index “m”; for example, ANInm.
j	Throughout this section, the individual virtual channels of ADCAn are identified by the index “j”; for example, ADCAnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCAn are identified by the index “x”; for example, ADCAnSGSTCRx (x = 1 to 3) indicates the scan group x start control register.
k	Throughout this section, the individual physical channel numbers for T&H are identified by the index “k”; for example, THkE (k = 0 to 5) is T&H enable bit of the T&H enable register (ADCAnTHER).

Table 38.12 Indices (RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual ADCA units are identified by the index “n”; for example, ADCAnPWDVCR (n = 0) indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels (channels in the unit) of ADCAn are identified by the index “m”; for example, ANInm.
j	Throughout this section, the individual virtual channels of ADCAn are identified by the index “j”; for example, ADCAnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCAn are identified by the index “x”; for example, ADCAnSGSTCRx (x = 1 to 3) indicates the scan group x start control register.
k	Throughout this section, the individual physical channel numbers for T&H are identified by the index “k”; for example, THkE (k = 0 to 5) is T&H enable bit of the T&H enable register (ADCAnTHER).

The following table shows values indicated by the indices of each product.

Table 38.13 Indices of Products (RH850/F1KH-D8)

Indices of Each Product		
RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 23 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 35 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 35 (ADCA1)
j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)
x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)
k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)

Table 38.14 Indices of Products (RH850/F1KM-S4)

Indices of Each Product				
RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
m = 0 to 15, 18 to 27, 30 to 35 (ADCA0)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 7, 16 to 19 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 23 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 35 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 35 (ADCA1)
j = 00 to 49 (ADCA0)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 35 (ADCA1)
x = 1 to 3 (ADCA0)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)	x = 1 to 3 (ADCA0) x = 1 to 3 (ADCA1)
k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)	k = 0 to 5 (ADCA0)

Table 38.15 Indices of Products (RH850/F1KM-S1)

Indices of Each Product			
RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
m = 0 to 7, 16 to 19 (ADCA0)	m = 0 to 9, 16 to 26(ADCA0)	m = 0 to 10, 16 to 29 (ADCA0)	m = 0 to 35 (ADCA0)
j = 00 to 19*[1] (ADCA0)	j = 00 to 28*[1] (ADCA0)	j = 00 to 36 (ADCA0)	j = 00 to 49 (ADCA0)
x = 1 to 3 (ADCA0)	x = 1 to 3 (ADCA0)	x = 1 to 3 (ADCA0)	x = 1 to 3 (ADCA0)
k = 0, 2, 4 (ADCA0)	k = 0, 2, 4 (ADCA0)	k = 0, 2, 4 (ADCA0)	k = 0 to 5 (ADCA0)

Note 1. j = 33 to 35 are supported only in diagnosis of T&H circuit.

38.1.2 Register Base Addresses

ADCAn base addresses are listed in the following table.

ADCAn register addresses are given as offsets from the base addresses.

Table 38.16 Register Base Addresses (RH850/F1KH-D8)

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000$_{H}$
<ADCA1_base>	FFD6 D000$_{H}$

Table 38.17 Register Base Addresses (RH850/F1KM-S4)

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000$_{H}$
<ADCA1_base>	FFD6 D000$_{H}$

Table 38.18 Register Base Address (RH850/F1KM-S1)

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000$_{H}$

38.1.3 Clock Supply

The ADCAn clock supply is shown in the following table.

Table 38.19 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name	Description
ADCA0	ADCLK	CKSCLK_AADCA	Module clock
	Register access clock	CPUCLK_L, CKSCLK_AADCA	Bus clock
ADCA1	ADCLK	CKSCLK_IADCA	Module clock
	Register access clock	CPUCLK_L	Bus clock

Table 38.20 Clock Supply (RH850/F1KM-S4)

Unit Name	Unit Clock Name	Supply Clock Name	Description
ADCA0	ADCLK	CKSCLK_AADCA	Module clock
	Register access clock	CPUCLK_L, CKSCLK_AADCA	Bus clock
ADCA1	ADCLK	CKSCLK_IADCA	Module clock
	Register access clock	CPUCLK_L	Bus clock

Table 38.21 Clock Supply (RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
ADCA0	ADCLK	CKSCLK_AADCA	Module clock
	Register access clock	CPUCLK_L	Bus clock
		CKSCLK_AADCA	

38.1.4 Interrupt Requests

ADCAn interrupt requests are listed in the following table.

Table 38.22 Interrupt Requests (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Other Trigger Signals
ADCA0				
INT_ADE	A/D error interrupt	56	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	18	4	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	19	5	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	20, 32	6	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7	—
ADCA1				
INT_ADE	A/D error interrupt	212	—	—
INT_SG1	Scan group 1 (SG1) end interrupt	213	115	—
INT_SG2	Scan group 2 (SG2) end interrupt	214	116	—
INT_SG3	Scan group 3 (SG3) end interrupt	215	117	—
ADC_CONV_END1	Scan group 4 (SG4) A/D conversion end signal	—	118	—

Table 38.23 Interrupt Requests (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Other Trigger Signals
ADCA0				
INT_ADE	A/D error interrupt	56	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	18	4	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	19	5	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	20, 32	6	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7	—
ADCA1				
INT_ADE	A/D error interrupt	212	—	—
INT_SG1	Scan group 1 (SG1) end interrupt	213	115	—
INT_SG2	Scan group 2 (SG2) end interrupt	214	116	—
INT_SG3	Scan group 3 (SG3) end interrupt	215	117	—
ADC_CONV_END1	Scan group 4 (SG4) A/D conversion end signal	—	118	—

Table 38.24 Interrupt Requests (RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Other Trigger Signals
ADCA0				
INT_ADE	A/D error interrupt	56	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	18	4	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	19	5	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	20, 32	6	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7	—

38.1.5 Reset Sources

ADCAn reset sources are listed in the following table. ADCAn is initialized by these reset sources.

Table 38.25 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DeepSTOP mode (AWORES)
ADCA1	All reset sources (ISORES)

Table 38.26 Reset Sources (RH850/F1KM-S4)

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DeepSTOP mode (AWORES)
ADCA1	All reset sources (ISORES)

Table 38.27 Reset Sources (RH850/F1KM-S1)

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DeepSTOP mode (AWORES)

38.1.6 External Input/Output Signals

External input/output signals of ADCAn are listed below.

Table 38.28 ADCA0 Analog Input Signals (RH850/F1KH-D8)

Unit Signal Name	Alternative Port Pin Signal	Resolution	T&H	RH850/ F1KH-D8 176 Pins	RH850/ F1KH-D8 233 Pins	RH850/ F1KH-D8 272 Pins	RH850/ F1KH-D8 324 Pins
ANI000	ADCA0I0	12	✓	✓	✓	—	✓
ANI001	ADCA0I1	12	✓	✓	✓	—	✓
ANI002	ADCA0I2	12	✓	✓	✓	—	✓
ANI003	ADCA0I3	12	✓	✓	✓	—	✓
ANI004	ADCA0I4	12	✓	✓	✓	—	✓
ANI005	ADCA0I5	12	✓	✓	✓	—	✓
ANI006	ADCA0I6	12	—	✓	✓	—	✓
ANI007	ADCA0I7	12	—	✓	✓	—	✓
ANI008	ADCA0I8	12	—	✓	✓	—	✓
ANI009	ADCA0I9	12	—	✓	✓	—	✓
ANI010	ADCA0I10	12	—	✓	✓	—	✓
ANI011	ADCA0I11	12	—	✓	✓	—	✓
ANI012	ADCA0I12	12	—	✓	✓	—	✓
ANI013	ADCA0I13	12	—	✓	✓	—	✓
ANI014	ADCA0I14	12	—	✓	✓	—	✓
ANI015	ADCA0I15	12	—	✓	✓	—	✓
ANI016	ADCA0I0S	10	—	✓	✓	—	✓
ANI017	ADCA0I1S	10	—	✓	✓	—	✓
ANI018	ADCA0I2S	10	—	✓	✓	—	✓
ANI019	ADCA0I3S	10	—	✓	✓	—	✓
ANI020	ADCA0I4S	10	—	✓	✓	—	✓
ANI021	ADCA0I5S	10	—	✓	✓	—	✓
ANI022	ADCA0I6S	10	—	✓	✓	—	✓
ANI023	ADCA0I7S	10	—	✓	✓	—	✓
ANI024	ADCA0I8S	10	—	✓	✓	—	✓
ANI025	ADCA0I9S	10	—	✓	✓	—	✓
ANI026	ADCA0I10S	10	—	✓	✓	—	✓
ANI027	ADCA0I11S	10	—	✓	✓	—	✓
ANI028	ADCA0I12S	10	—	*1	*1	—	*1
ANI029	ADCA0I13S	10	—	*1	*1	—	*1
ANI030	ADCA0I14S	10	—	✓	✓	—	✓
ANI031	ADCA0I15S	10	—	✓	✓	—	✓
ANI032	ADCA0I16S	10	—	✓	✓	—	✓
ANI033	ADCA0I17S	10	—	✓	✓	—	✓
ANI034	ADCA0I18S	10	—	✓	✓	—	✓
ANI035	ADCA0I19S	10	—	✓	✓	—	✓

Note 1. This is not supported in RH850/F1KH-D8 product.

Table 38.29 ADCA0 Analog Input Signals (RH850/F1KM-S4)

Unit Signal Name	Alternative Port Pin Signal	Resolution	T&H	RH850/ F1KM-S4 100 Pins	RH850/ F1KM-S4 144 Pins	RH850/ F1KM-S4 176 Pins	RH850/ F1KM-S4 233 Pins	RH850/ F1KM-S4 272 Pins
ANI000	ADCA0I0	12	✓	✓	✓	✓	✓	✓
ANI001	ADCA0I1	12	✓	✓	✓	✓	✓	✓
ANI002	ADCA0I2	12	✓	✓	✓	✓	✓	✓
ANI003	ADCA0I3	12	✓	✓	✓	✓	✓	✓
ANI004	ADCA0I4	12	✓	✓	✓	✓	✓	✓
ANI005	ADCA0I5	12	✓	✓	✓	✓	✓	✓
ANI006	ADCA0I6	12	—	✓	✓	✓	✓	✓
ANI007	ADCA0I7	12	—	✓	✓	✓	✓	✓
ANI008	ADCA0I8	12	—	✓	✓	✓	✓	✓
ANI009	ADCA0I9	12	—	✓	✓	✓	✓	✓
ANI010	ADCA0I10	12	—	✓	✓	✓	✓	✓
ANI011	ADCA0I11	12	—	✓	✓	✓	✓	✓
ANI012	ADCA0I12	12	—	✓	✓	✓	✓	✓
ANI013	ADCA0I13	12	—	✓	✓	✓	✓	✓
ANI014	ADCA0I14	12	—	✓	✓	✓	✓	✓
ANI015	ADCA0I15	12	—	✓	✓	✓	✓	✓
ANI016	ADCA0I0S	10	—	—	✓	✓	✓	✓
ANI017	ADCA0I1S	10	—	—	✓	✓	✓	✓
ANI018	ADCA0I2S	10	—	✓	✓	✓	✓	✓
ANI019	ADCA0I3S	10	—	✓	✓	✓	✓	✓
ANI020	ADCA0I4S	10	—	✓	✓	✓	✓	✓
ANI021	ADCA0I5S	10	—	✓	✓	✓	✓	✓
ANI022	ADCA0I6S	10	—	✓	✓	✓	✓	✓
ANI023	ADCA0I7S	10	—	✓	✓	✓	✓	✓
ANI024	ADCA0I8S	10	—	✓	✓	✓	✓	✓
ANI025	ADCA0I9S	10	—	✓	✓	✓	✓	✓
ANI026	ADCA0I10S	10	—	✓	✓	✓	✓	✓
ANI027	ADCA0I11S	10	—	✓	✓	✓	✓	✓
ANI028	ADCA0I12S	10	—	*1	*1	*1	*1	*1
ANI029	ADCA0I13S	10	—	*1	*1	*1	*1	*1
ANI030	ADCA0I14S	10	—	✓	✓	✓	✓	✓
ANI031	ADCA0I15S	10	—	✓	✓	✓	✓	✓
ANI032	ADCA0I16S	10	—	✓	✓	✓	✓	✓
ANI033	ADCA0I17S	10	—	✓	✓	✓	✓	✓
ANI034	ADCA0I18S	10	—	✓	✓	✓	✓	✓
ANI035	ADCA0I19S	10	—	✓	✓	✓	✓	✓

Note 1. This is not supported in RH850/F1KM-S4 product.

Table 38.30 ADCA0 Analog Input Signals (RH850/F1KM-S1)

Unit Signal Name	Alternative Port Pin Signal	Resolution	T&H	RH850/ F1KM-S1 48 Pins	RH850/ F1KM-S1 64 Pins	RH850/ F1KM-S1 80 Pins	RH850/ F1KM-S1 100 Pins
ANI000	ADCA0I0	12	✓	✓	✓	✓	✓
ANI001	ADCA0I1	12	✓*1	✓	✓	✓	✓
ANI002	ADCA0I2	12	✓	✓	✓	✓	✓
ANI003	ADCA0I3	12	✓*1	✓	✓	✓	✓
ANI004	ADCA0I4	12	✓	✓	✓	✓	✓
ANI005	ADCA0I5	12	✓*1	✓	✓	✓	✓
ANI006	ADCA0I6	12	—	✓	✓	✓	✓
ANI007	ADCA0I7	12	—	✓	✓	✓	✓
ANI008	ADCA0I8	12	—	—	✓	✓	✓
ANI009	ADCA0I9	12	—	—	✓	✓	✓
ANI010	ADCA0I10	12	—	—	—	✓	✓
ANI011	ADCA0I11	12	—	—	—	—	✓
ANI012	ADCA0I12	12	—	—	—	—	✓
ANI013	ADCA0I13	12	—	—	—	—	✓
ANI014	ADCA0I14	12	—	—	—	—	✓
ANI015	ADCA0I15	12	—	—	—	—	✓
ANI016	ADCA0I0S	10	—	✓	✓	✓	✓
ANI017	ADCA0I1S	10	—	✓	✓	✓	✓
ANI018	ADCA0I2S	10	—	✓	✓	✓	✓
ANI019	ADCA0I3S	10	—	✓	✓	✓	✓
ANI020	ADCA0I4S	10	—	—	✓	✓	✓
ANI021	ADCA0I5S	10	—	—	✓	✓	✓
ANI022	ADCA0I6S	10	—	—	✓	✓	✓
ANI023	ADCA0I7S	10	—	—	✓	✓	✓
ANI024	ADCA0I8S	10	—	—	✓	✓	✓
ANI025	ADCA0I9S	10	—	—	✓	✓	✓
ANI026	ADCA0I10S	10	—	—	✓	✓	✓
ANI027	ADCA0I11S	10	—	—	—	✓	✓
ANI028	ADCA0I12S	10	—	—	—	✓	✓
ANI029	ADCA0I13S	10	—	—	—	✓	✓
ANI030	ADCA0I14S	10	—	—	—	—	✓
ANI031	ADCA0I15S	10	—	—	—	—	✓
ANI032	ADCA0I16S	10	—	—	—	—	✓
ANI033	ADCA0I17S	10	—	—	—	—	✓
ANI034	ADCA0I18S	10	—	—	—	—	✓
ANI035	ADCA0I19S	10	—	—	—	—	✓

Note 1. The track and hold circuit (T&H) for ANI0m (m = 01, 03, 05) is only included on RH850/F1KM-S1 devices with 100 pins.

Table 38.31 ADCA0 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA0		
ADCA0TRG0	External trigger pin (scan group 1)*1	ADCA0TRG0
ADCA0TRG1	External trigger pin (scan group 2)*1	ADCA0TRG1
ADCA0TRG2	External trigger pin (scan group 3)*1	ADCA0TRG2
ADCA0SEL0	External analog multiplexer (MPX) output pin 0	ADCA0SEL0
ADCA0SEL1	External analog multiplexer (MPX) output pin 1	ADCA0SEL1
ADCA0SEL2	External analog multiplexer (MPX) output pin 2	ADCA0SEL2

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector** and **Section 2C.12, Noise Filter & Edge/Level Detector**.

CAUTION

- When port P8_6 is used as ADCA0I8S, port P8_6 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.
  For details, see **Section 2A.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**, **Section 2B.11.1.1, P8_6: $\overline{\text{RESETOUT}}$** and **Section 2C.11.1.1, P8_6: $\overline{\text{RESETOUT}}$**.
- The RH850/F1KH-D8 and RH850/F1KM-S4 do not have ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S) pins, so do not attempt writing to the corresponding bits or controlling the signals.
  For descriptions in this section that mention ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S), read them as stating that ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S) are not supported.

Table 38.32 ADCA1 Analog Input Signals (RH850/F1KH-D8)

Unit Signal Name	Alternative Port Pin Signal	Resolution	T&H	RH850/ F1KH-D8 176 Pins	RH850/ F1KH-D8 233 Pins	RH850/ F1KH-D8 272 Pins	RH850/ F1KH-D8 324 Pins
ANI100	ADCA1I0	12	—	✓	✓	—	✓
ANI101	ADCA1I1	12	—	✓	✓	—	✓
ANI102	ADCA1I2	12	—	✓	✓	—	✓
ANI103	ADCA1I3	12	—	✓	✓	—	✓
ANI104	ADCA1I4	12	—	✓	✓	—	✓
ANI105	ADCA1I5	12	—	✓	✓	—	✓
ANI106	ADCA1I6	12	—	✓	✓	—	✓
ANI107	ADCA1I7	12	—	✓	✓	—	✓
ANI108	ADCA1I8	12	—	✓	✓	—	✓
ANI109	ADCA1I9	12	—	✓	✓	—	✓
ANI110	ADCA1I10	12	—	✓	✓	—	✓
ANI111	ADCA1I11	12	—	✓	✓	—	✓
ANI112	ADCA1I12	12	—	✓	✓	—	✓
ANI113	ADCA1I13	12	—	✓	✓	—	✓
ANI114	ADCA1I14	12	—	✓	✓	—	✓
ANI115	ADCA1I15	12	—	✓	✓	—	✓
ANI116	ADCA1I0S	10	—	✓	✓	—	✓
ANI117	ADCA1I1S	10	—	✓	✓	—	✓
ANI118	ADCA1I2S	10	—	✓	✓	—	✓
ANI119	ADCA1I3S	10	—	✓	✓	—	✓
ANI120	ADCA1I4S	10	—	✓	✓	—	✓
ANI121	ADCA1I5S	10	—	✓	✓	—	✓
ANI122	ADCA1I6S	10	—	✓	✓	—	✓
ANI123	ADCA1I7S	10	—	✓	✓	—	✓
ANI124	ADCA1I8S	10	—	—	✓	—	✓
ANI125	ADCA1I9S	10	—	—	✓	—	✓
ANI126	ADCA1I10S	10	—	—	✓	—	✓
ANI127	ADCA1I11S	10	—	—	✓	—	✓
ANI128	ADCA1I12S	10	—	—	✓	—	✓
ANI129	ADCA1I13S	10	—	—	✓	—	✓
ANI130	ADCA1I14S	10	—	—	✓	—	✓
ANI131	ADCA1I15S	10	—	—	✓	—	✓
ANI132	ADCA1I16S	10	—	—	✓	—	✓
ANI133	ADCA1I17S	10	—	—	✓	—	✓
ANI134	ADCA1I18S	10	—	—	✓	—	✓
ANI135	ADCA1I19S	10	—	—	✓	—	✓

Table 38.33 ADCA1 Analog Input Signals (RH850/F1KM-S4)

Unit Signal Name	Alternative Port Pin Signal	Resolution	T&H	RH850/ F1KM-S4 100 Pins	RH850/ F1KM-S4 144 Pins	RH850/ F1KM-S4 176 Pins	RH850/ F1KM-S4 233 Pins	RH850/ F1KM-S4 272 Pins
ANI100	ADCA1I0	12	—	—	✓	✓	✓	✓
ANI101	ADCA1I1	12	—	—	✓	✓	✓	✓
ANI102	ADCA1I2	12	—	—	✓	✓	✓	✓
ANI103	ADCA1I3	12	—	—	✓	✓	✓	✓
ANI104	ADCA1I4	12	—	—	✓	✓	✓	✓
ANI105	ADCA1I5	12	—	—	✓	✓	✓	✓
ANI106	ADCA1I6	12	—	—	✓	✓	✓	✓
ANI107	ADCA1I7	12	—	—	✓	✓	✓	✓
ANI108	ADCA1I8	12	—	—	—	✓	✓	✓
ANI109	ADCA1I9	12	—	—	—	✓	✓	✓
ANI110	ADCA1I10	12	—	—	—	✓	✓	✓
ANI111	ADCA1I11	12	—	—	—	✓	✓	✓
ANI112	ADCA1I12	12	—	—	—	✓	✓	✓
ANI113	ADCA1I13	12	—	—	—	✓	✓	✓
ANI114	ADCA1I14	12	—	—	—	✓	✓	✓
ANI115	ADCA1I15	12	—	—	—	✓	✓	✓
ANI116	ADCA1I0S	10	—	—	✓	✓	✓	✓
ANI117	ADCA1I1S	10	—	—	✓	✓	✓	✓
ANI118	ADCA1I2S	10	—	—	✓	✓	✓	✓
ANI119	ADCA1I3S	10	—	—	✓	✓	✓	✓
ANI120	ADCA1I4S	10	—	—	—	✓	✓	✓
ANI121	ADCA1I5S	10	—	—	—	✓	✓	✓
ANI122	ADCA1I6S	10	—	—	—	✓	✓	✓
ANI123	ADCA1I7S	10	—	—	—	✓	✓	✓
ANI124	ADCA1I8S	10	—	—	—	—	✓	✓
ANI125	ADCA1I9S	10	—	—	—	—	✓	✓
ANI126	ADCA1I10S	10	—	—	—	—	✓	✓
ANI127	ADCA1I11S	10	—	—	—	—	✓	✓
ANI128	ADCA1I12S	10	—	—	—	—	✓	✓
ANI129	ADCA1I13S	10	—	—	—	—	✓	✓
ANI130	ADCA1I14S	10	—	—	—	—	✓	✓
ANI131	ADCA1I15S	10	—	—	—	—	✓	✓
ANI132	ADCA1I16S	10	—	—	—	—	✓	✓
ANI133	ADCA1I17S	10	—	—	—	—	✓	✓
ANI134	ADCA1I18S	10	—	—	—	—	✓	✓
ANI135	ADCA1I19S	10	—	—	—	—	✓	✓

Table 38.34 ADCA1 External Input/Output Signals (RH850/F1KH-D8)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA1		
ADCA1TRG0	External trigger pin (scan group 1)*[1]	ADCA1TRG0
ADCA1TRG1	External trigger pin (scan group 2)*[1]	ADCA1TRG1
ADCA1TRG2	External trigger pin (scan group 3)*[1]	ADCA1TRG2

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**.

Table 38.35 ADCA1 External Input/Output Signals (RH850/F1KM-S4)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA1		
ADCA1TRG0	External trigger pin (scan group 1)*[1]	ADCA1TRG0
ADCA1TRG1	External trigger pin (scan group 2)*[1]	ADCA1TRG1
ADCA1TRG2	External trigger pin (scan group 3)*[1]	ADCA1TRG2

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, see **Section 2B.12, Noise Filter & Edge/Level Detector**

38.2 Overview

38.2.1 Functional Overview

ADCA has the following features.

- 10-bit/12-bit resolution
- Successive approximation conversion method
- Number of A/D input channels
  A/D conversion is available for a maximum of 34 ADCA0 channels and 36 ADCA1 channels. Additionally, ADCA0 supports the connection of an external analog multiplexer (MPX) to extend the number of analog input channels.
- Internal track and hold (T&H) circuit
  ANI000 to ANI005 (ADCA0I0 to ADCA0I5) of ADCA0 include the track and hold circuit. The track and hold circuit can sample up to 6 channels of analog input simultaneously.
- A/D conversion control by scan groups
  The A/D conversion channel or conversion mode (scan mode) can be set for each scan group.
- Two scan modes
  Multi-cycle scan mode: Specified number of scans are executed.
  Continuous scan mode: Scans are executed repeatedly and continuously.
- Asynchronous/synchronous suspend and resume function
  A processing for a scan group can be interrupted to run the processing for another scan group.
- Start/Stop trigger for each scan group
  Software, hardware, and external trigger can start processing of each scan group.
  Software can stop processing of each scan group.
- Scan end interrupt and DMA transfer are supported.
  For each scan group, an interrupt request to INTC can be issued or DMA transfer can be started, each time a processing for the virtual channel indicated by the end virtual channel pointer ends, or a virtual channel ends.
- A/D conversion channel repeat function
  A/D conversion is performed for the same channel two or four times sequentially, and the result is stored in the data register.
- Abundant safety functions
  Abundant safety functions are provided, such as A/D converter diagnosis, diagnosis of the channel multiplexer, diagnosis of open pins, diagnosis of the T&H circuit, 8 pairs of upper limit/lower limit check for the A/D converter and have error flag register for each virtual channel, overwrite check for data registers, and read and clear function for data registers.
- Shortest A/D conversion time per channel
  1.15 µs (when MPX is not used)
  2.30 µs (when MPX is used)
- Configurable stabilization time for MPX analog input
  Stabilization time can be selected by 8 patterns of counter setting for each physical channel.
  Min. 1us
  Max. 250us

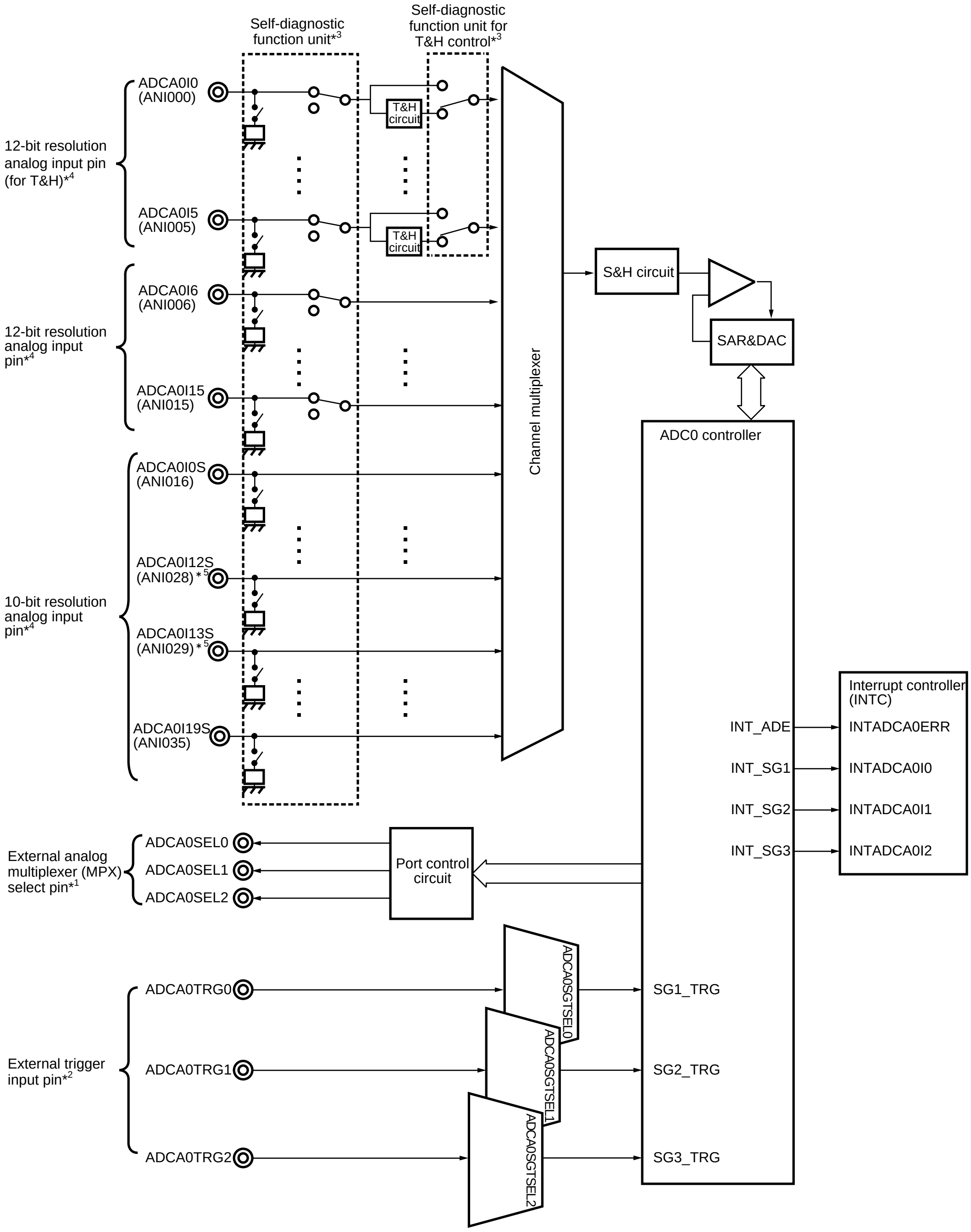
NOTE

- Physical channel (ANInm)
  Each A/D input channel of ADCA0 and ADCA1 units is called a physical channel. The physical channel of each unit is represented as ANI0m (m = 0 to 27, 30 to 35) for ADCA0 and ANI1m (m = 0 to 35) for ADCA1.
  In RH850/F1KH, RH850/F1KM, the alternative port pins for 12-bit resolution A/D input channel and 10-bit resolution A/D input channel are represented as ADCAnIm and ADCAnImS, respectively. In this section, the physical channels and the corresponding alternative port pins are listed together.
- Virtual channel (ADCAnVCRj)
  ADCA0 has a maximum of 50 virtual channels, and ADCA1 has a maximum of 36 virtual channels. The virtual channel specifies the physical channel to be scanned.
  Scans are executed in sequence from the smallest virtual channel number. The scan order can be arbitrarily-specified by using virtual channels. In addition, the scanned result is stored in the data register (ADCAnDRj) corresponding to the virtual channel.
- Scan group (SGx)
  ADCA has three scan groups (SG1, SG2, SG3) and one PWM-Diag group (SG4). A/D conversion is executed in scan group unit. The channel to be scanned can be selected for each group by specifying the scan range, that is, the conversion start virtual channel and the conversion end virtual channel.

38.2.2 Block Diagram

The block diagram of ADCA0 is shown in **Figure 38.1, ADCA0 Block Diagram**. The block diagram of ADCA1 is shown in **Figure 38.2, ADCA1 Block Diagram**.

(1) Configuration of ADCA0



Note 1. For an example of external analog multiplexer connection, see **Figure 38.4, Example of External Analog Multiplexer Connection**.

Note 2. For details, see **Figure 38.3, Internal Connection Diagram of External Trigger Input Pins**.

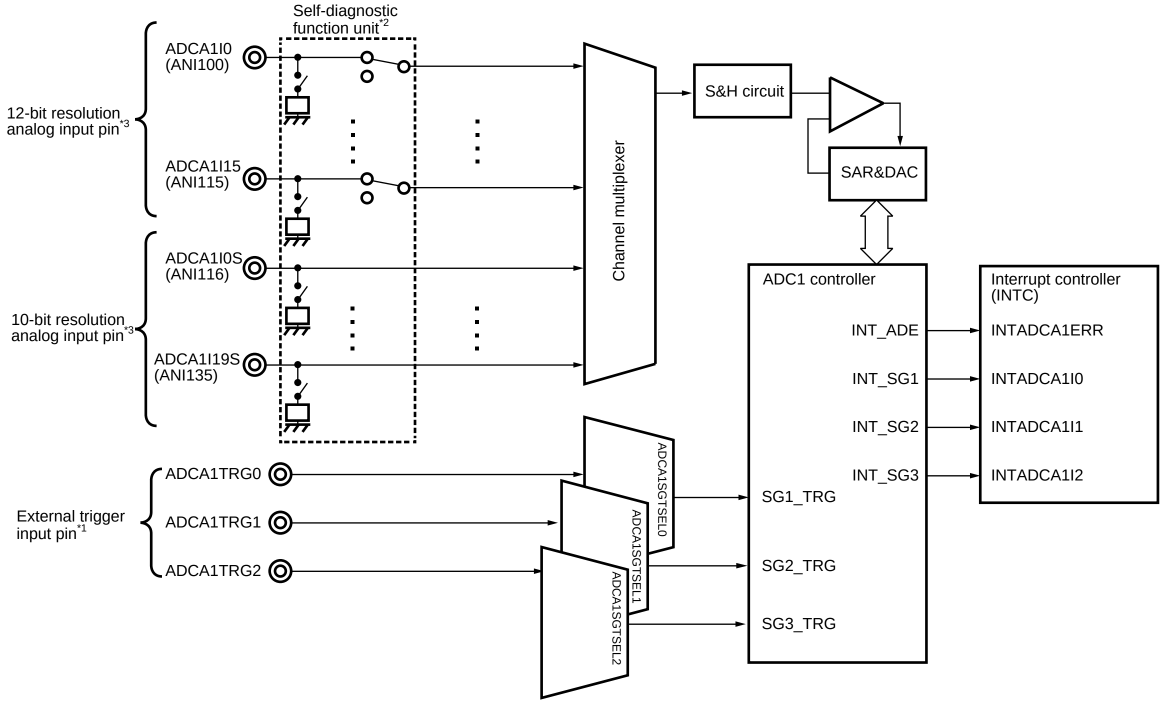
Note 3. For the self-diagnostic function unit, see **Section 38.5, Self-Diagnostic Function**.

Note 4. For the analog input of each product, see **Table 38.28, ADCA0 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.29, ADCA0 Analog Input Signals (RH850/F1KM-S4)**, **Table 38.30, ADCA0 Analog Input Signals (RH850/F1KM-S1)**.

Note 5. The RH850/F1KH-D8 and RH850/F1KM-S4 do not have ADCA0I12S (ANI028) and ADCA0I13S (ANI029) pins.

Figure 38.1 ADCA0 Block Diagram

(2) Configuration of ADCA1



Note 1. For details, see **Figure 38.3, Internal Connection Diagram of External Trigger Input Pins**.

Note 2. For the self-diagnostic function unit, see **Section 38.5, Self-Diagnostic Function**.

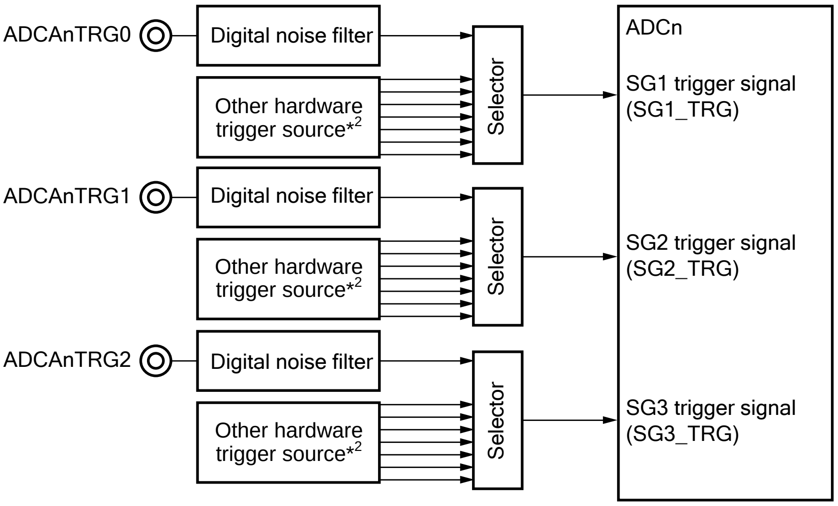
Note 3. For the analog input of each product, see **Table 38.32, ADCA1 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.33, ADCA1 Analog Input Signals (RH850/F1KM-S4)**.

Figure 38.2 ADCA1 Block Diagram

(3) Configuration of External Trigger Input Pins

An external trigger input pin is a hardware trigger source to activate ADCAn.

The configuration of external trigger input pins is shown below.



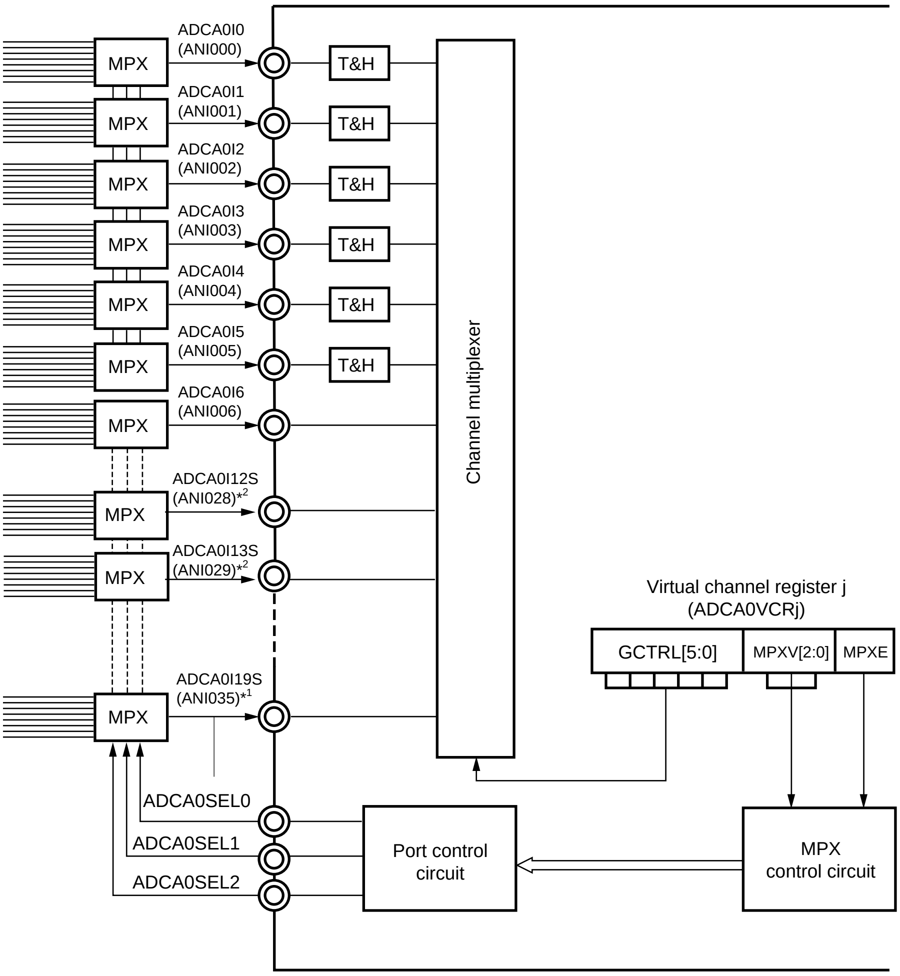
Note 1. For the condition setting to accept a trigger from an external trigger input pin, see **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector** and **Section 2C.12, Noise Filter & Edge/Level Detector**.

Note 2. For the hardware trigger sources, see **Table 38.78, List of A/D Conversion Hardware Triggers**.

Figure 38.3 Internal Connection Diagram of External Trigger Input Pins

(4) Configuration of External Analog Multiplexer (MPX)

The external analog multiplexer (MPX) can be connected to any input signal pins ADCA0I0 to ADCA0I19S. An example of the external analog multiplexer connection is shown below.



Note: The T&H function is disabled when the external analog multiplexer (MPX) is used

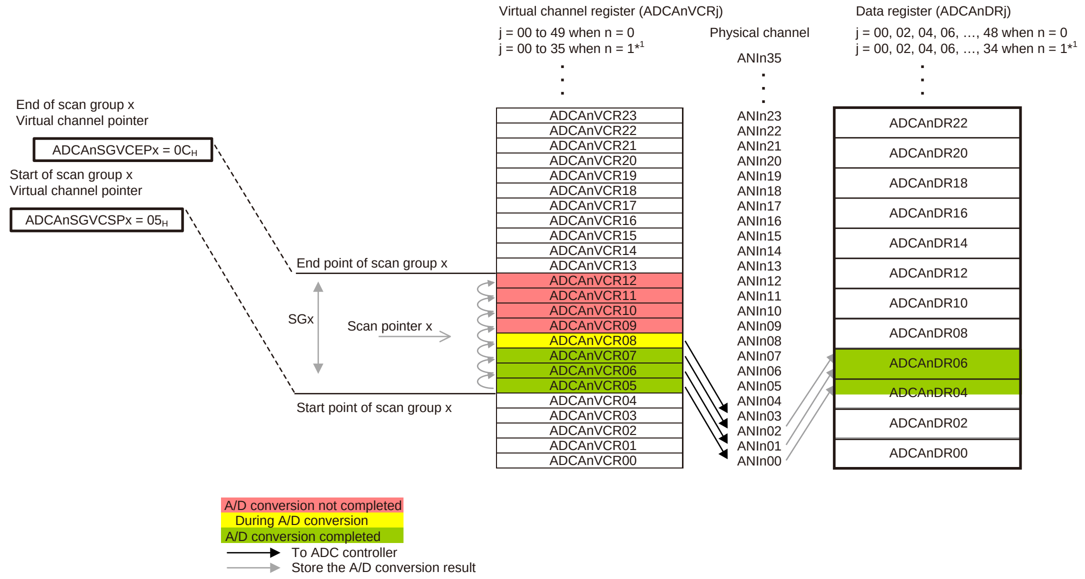
Note 1. For the analog input of each product, see **Table 38.28**, **ADCA0 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.29, ADCA0 Analog Input Signals (RH850/F1KM-S4)**, **Table 38.30, ADCA0 Analog Input Signals (RH850/F1KM-S1)**.

Note 2. The RH850/F1KH-D8 and RH850/F1KM-S4 do not have ADCA0I12S (ANI028) and ADCA0I13S (ANI029) pins.

Figure 38.4 Example of External Analog Multiplexer Connection

(5) Virtual Channel

The virtual channel specifies the physical address to be scanned.
The virtual channel is controlled by the ADCAnVCRj register.
A usage example of the virtual channel is shown below.



Note 1. This is not supported in RH850/F1KM-S4 with 100 pins and RH850/F1KM-S1.

Example) When scan group x starts scanning 5th to 12th virtual channels

1. The ADCAnVCRj register specifies the physical channel (ANInm) for which virtual channel j executes the A/D conversion. The A/D conversion pins can be set in any order. (The figure above shows an example when ANIn0 to ANIn7 are selected.)
2. The scan pointer x sequentially reads the physical channels specified in the ADCAnVCRj register from “start point of scan group x” to “end point of scan group x”, and starts the A/D conversion for the read physical address.
3. The scanned result of virtual channel j is stored in data register j.

Figure 38.5 Usage Example of Virtual Register

38.3 Registers

38.3.1 List of Registers

The ADCA registers are listed in the following table.

For details about <ADCAn_base>, see **Section 38.1.2, Register Base Addresses**.

Table 38.36 List of Registers

Module Name	Register Name	Symbol	Address
ADCA specific registers (virtual channel)			
ADCAn	Virtual channel register j	ADCAnVCRj	<ADCAn_base> + j × 4_H
	PWM-Diag virtual channel register	ADCAnPWDVCR	<ADCAn_base> + $0F4_H$
	Data register j	ADCAnDRj	<ADCAn_base> + 100_H + j × 2_H
	Data supplementary information register j	ADCAnDIRj	<ADCAn_base> + 200_H + j × 4_H
	PWM-Diag data register	ADCAnPWDTSNDR	<ADCAn_base> + 178_H
	PWM-Diag data supplementary information register	ADCAnPWDDIR	<ADCAn_base> + $2F4_H$
ADCA specific registers (control)			
ADCAn	A/D force halt register	ADCAnADHALTR	<ADCAn_base> + 300_H
	A/D control register	ADCAnADCR	<ADCAn_base> + 304_H
	MPX current register	ADCAnMPXCURR	<ADCAn_base> + $30C_H$
	T&H sampling start control register	ADCAnTHSMPSTCR	<ADCAn_base> + 314_H
	T&H control register	ADCAnTHCR	<ADCAn_base> + 318_H
	T&H group A hold start control register	ADCAnTHAHLDSTCR	<ADCAn_base> + $31C_H$
	T&H group B hold start control register	ADCAnTHBHLDSTCR	<ADCAn_base> + 320_H
	T&H group A control register	ADCAnTHACR	<ADCAn_base> + 324_H
	T&H group B control register	ADCAnTHBCR	<ADCAn_base> + 328_H
	T&H enable register	ADCAnTHER	<ADCAn_base> + $32C_H$
	T&H group select register	ADCAnTHGSR	<ADCAn_base> + 330_H
	Sampling control register	ADCAnSMPCR	<ADCAn_base> + 380_H
	MPX stabilization time selection register 0	ADCAnMPXSTBTSELR0	<ADCAn_base> + $5F0_H$
	MPX stabilization time selection register 1	ADCAnMPXSTBTSELR1	<ADCAn_base> + $5F4_H$
	MPX stabilization time selection register 2	ADCAnMPXSTBTSELR2	<ADCAn_base> + $5F8_H$
	MPX stabilization time selection register 3	ADCAnMPXSTBTSELR3	<ADCAn_base> + $5FC_H$
	MPX stabilization time selection register 4	ADCAnMPXSTBTSELR4	<ADCAn_base> + 600_H
	MPX stabilization time setting register 0	ADCAnMPXSTBTR0	<ADCAn_base> + 610_H
	MPX stabilization time setting register 1	ADCAnMPXSTBTR1	<ADCAn_base> + 614_H
	MPX stabilization time setting register 2	ADCAnMPXSTBTR2	<ADCAn_base> + 618_H
	MPX stabilization time setting register 3	ADCAnMPXSTBTR3	<ADCAn_base> + $61C_H$
	MPX stabilization time setting register 4	ADCAnMPXSTBTR4	<ADCAn_base> + 620_H
	MPX stabilization time setting register 5	ADCAnMPXSTBTR5	<ADCAn_base> + 624_H
	MPX stabilization time setting register 6	ADCAnMPXSTBTR6	<ADCAn_base> + 628_H
	MPX stabilization time setting register 7	ADCAnMPXSTBTR7	<ADCAn_base> + $62C_H$

Table 38.36 List of Registers

Module Name	Register Name	Symbol	Address
ADCA specific registers (safety-related)			
ADCAn	Safety control register	ADCAnSFTCR	<ADCAn_base> + 334_H
	Upper limit/lower limit table register 0	ADCAnULLMTBR0	<ADCAn_base> + 338_H
	Upper limit/lower limit table register 1	ADCAnULLMTBR1	<ADCAn_base> + $33C_H$
	Upper limit/lower limit table register 2	ADCAnULLMTBR2	<ADCAn_base> + 340_H
	Error clear register	ADCAnECR	<ADCAn_base> + 344_H
	Upper limit/lower limit error register	ADCAnULER	<ADCAn_base> + 348_H
	Overwrite error register	ADCAnOWER	<ADCAn_base> + $34C_H$
	Upper limit/lower limit error VC flag register 0	ADCAnULEVCFR0	<ADCAn_base> + 540_H
	Upper limit/lower limit error VC flag register 1	ADCAnULEVCFR1	<ADCAn_base> + 544_H
	Upper limit/lower limit table register 3	ADCAnULLMTBR3	<ADCAn_base> + 564_H
	Upper limit/lower limit table register 4	ADCAnULLMTBR4	<ADCAn_base> + 568_H
	Upper limit/lower limit table register 5	ADCAnULLMTBR5	<ADCAn_base> + $56C_H$
	Upper limit/lower limit table register 6	ADCAnULLMTBR6	<ADCAn_base> + 570_H
	Upper limit/lower limit table register 7	ADCAnULLMTBR7	<ADCAn_base> + 574_H
Scan group (SG) specific registers			
ADCAn	Scan group x start control register	ADCAnSGSTCRx	<ADCAn_base> + x × 40_H + 400_H
	PWM-Diag scan stop control register	ADCAnPWDSGSTPCR	<ADCAn_base> + 504_H
	PWM-Diag scan group control register	ADCAnPWDSGCR	<ADCAn_base> + 508_H
	Scan group x control register	ADCAnSGCRx	<ADCAn_base> + x × 40_H + 408_H
	Scan group x start virtual channel pointer	ADCAnSGVCSPx	<ADCAn_base> + x × 40_H + $40C_H$
	Scan group x end virtual channel pointer	ADCAnSGVCEPx	<ADCAn_base> + x × 40_H + 410_H
	Scan group x multicycle register	ADCAnSGMCYCRx	<ADCAn_base> + x × 40_H + 414_H
	PWM-Diag scan end flag clear register	ADCAnPWDSGSEFCR	<ADCAn_base> + 518_H
	Scan group x scan end flag clear register	ADCAnSGSEFCRx	<ADCAn_base> + x × 40_H + 418_H
	Scan group status register	ADCAnSGSTR	<ADCAn_base> + 308_H
	Scan group x stop control register	ADCAnSGSTPCRx	<ADCAn_base> + x × 40_H + 404_H
Hardware trigger specific register			
ADCAn	Scan group x start trigger control register	ADCAnSGTSELx	<ADCAn_base> + x × 40_H + $41C_H$
Self-diagnosis specific registers			
ADCAn	Self-diagnosis control register 0	ADCAnDGCTL0	<ADCAn_base> + 350_H
	Self-diagnosis control register 1	ADCAnDGCTL1	<ADCAn_base> + 354_H
	Pull down control register 1	ADCAnPDCTL1	<ADCAn_base> + 358_H
	Pull down control register 2	ADCAnPDCTL2	<ADCAn_base> + $35C_H$
Emulation specific register			
ADCAn	Emulation control register	ADCAnEMU	<ADCAn_base> + 388_H

38.3.2 ADCA Specific Registers

This section describes the registers that are equipped in each of ADCA0 and ADCA1.

38.3.2.1 ADCAnVCRj — Virtual Channel Register j

This register is used to control the virtual channel.

Access: ADCAnVCRj register can be read or written in 32-bit units.
ADCAnVCRjL register can be read or written in 16-bit units.
ADCAnVCRjLL and ADCAnVCRjLH registers can be read or written in 8-bit units.

Address: ADCAnVCRj: <ADCAn_base> + j × 4_H
ADCAnVCRjL: <ADCAn_base> + j × 4_H
ADCAnVCRjLL: <ADCAn_base> + j × 4_H
ADCAnVCRjLH: <ADCAn_base> + j × 4_H + 1_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE*1,*2	MPXV[2:0]*1			ULS[3:2]		CNVCLS*1,*3	ADIE	ULS[1:0]		GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.37 ADCAnVCRj Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	MPXE*1,*2	MPX Enable 0: The use of MPX is prohibited. No wait is inserted before A/D conversion is performed. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0]*1	These bits are used to set the MPX value to be transferred to an external analog multiplexer.
11, 10	ULS[3:2]	Upper Limit/Lower Limit Table Select*4
9	CNVCLS*1,*3	A/D Conversion Type Select for Self-Diagnosis 0: A/D conversion of the hold value is performed during a self-diagnosis. 1: Normal A/D conversion is performed during a self-diagnosis. When normal A/D conversion is performed during a self-diagnosis and MPX is in use (MPXE is set), however, a wait of one A/D conversion time is inserted before A/D conversion is performed. On the other hand, MPX cannot be used when A/D conversion of the hold value is performed during a self-diagnosis.
8	ADIE	A/D Conversion End Interrupt Enable 0: A scan group x end interrupt (INT_SGx) is not generated when A/D conversion for virtual channel j ends in SGx. 1: A scan group x end interrupt (INT_SGx) is generated when A/D conversion for virtual channel j ends in SGx.

Table 38.37 ADCAnVCRj Register Contents

Bit Position	Bit Name	Function
7, 6	ULS[1:0]	Upper Limit/Lower Limit Table Select*4
5 to 0	GCTRL[5:0]	Physical Channel Select 0_H to 23_H: Corresponding ANInm is selected. 24_H: Diagnosis channel for A/D converter is selected. Others: Setting prohibit **NOTE:** These bits shall set the physical channel which is supported. See **Table 38.13, Indices of Products (RH850/F1KH-D8)**, **Table 38.14, Indices of Products (RH850/F1KM-S4)**, **Table 38.15, Indices of Products (RH850/F1KM-S1)**, **Table 38.28, ADCA0 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.29, ADCA0 Analog Input Signals (RH850/F1KM-S4)**, **Table 38.30, ADCA0 Analog Input Signals (RH850/F1KM-S1)**, **Table 38.32, ADCA1 Analog Input Signals (RH850/F1KH-D8)** and **Table 38.33, ADCA1 Analog Input Signals (RH850/F1KM-S4)**.

Note 1. These bits are only supported for ADCA0. For ADCA1, when writing, write the value after reset.

Note 2. When GCTRL[5:0] is set to 24_H (Self-Diagnosis channel), set MPXE to 0.

Note 3. This bit is only supported when j = 33 to 35. Otherwise, when writing, write the value after reset.

Note 4. ULS[3:0] =
0000: Upper limit and lower limit are not checked
0001: Upper limit and lower limit are checked for ADCAnULLMTBR0.
0010: Upper limit and lower limit are checked for ADCAnULLMTBR1.
0011: Upper limit and lower limit are checked for ADCAnULLMTBR2.
0100: Upper limit and lower limit are checked for ADCAnULLMTBR3.
0101: Upper limit and lower limit are checked for ADCAnULLMTBR4.
0110: Upper limit and lower limit are checked for ADCAnULLMTBR5.
0111: Upper limit and lower limit are checked for ADCAnULLMTBR6.
1000: Upper limit and lower limit are checked for ADCAnULLMTBR7.
Other than above : Setting prohibited (Upper limit and lower limit are not checked when set)

CAUTION

To prevent malfunction, ADCAnVCRj should be set (except ADIE = 0 setting when stop procedure of scan group) when SGACT of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

Table 38.38 Selection of Physical Channels

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
0	0	0	0	0	0	ADCAnI0 (Physical channel ANIn00)
0	0	0	0	0	1	ADCAnI1 (Physical channel ANIn01)
0	0	0	0	1	0	ADCAnI2 (Physical channel ANIn02)
0	0	0	0	1	1	ADCAnI3 (Physical channel ANIn03)
0	0	0	1	0	0	ADCAnI4 (Physical channel ANIn04)
0	0	0	1	0	1	ADCAnI5 (Physical channel ANIn05)
0	0	0	1	1	0	ADCAnI6 (Physical channel ANIn06)
0	0	0	1	1	1	ADCAnI7 (Physical channel ANIn07)
0	0	1	0	0	0	ADCAnI8 (Physical channel ANIn08)
0	0	1	0	0	1	ADCAnI9 (Physical channel ANIn09)
0	0	1	0	1	0	ADCAnI10 (Physical channel ANIn10)
0	0	1	0	1	1	ADCAnI11 (Physical channel ANIn11)
0	0	1	1	0	0	ADCAnI12 (Physical channel ANIn12)
0	0	1	1	0	1	ADCAnI13 (Physical channel ANIn13)
0	0	1	1	1	0	ADCAnI14 (Physical channel ANIn14)
0	0	1	1	1	1	ADCAnI15 (Physical channel ANIn15)
0	1	0	0	0	0	ADCAnI0S (Physical channel ANIn16)
0	1	0	0	0	1	ADCAnI1S (Physical channel ANIn17)
0	1	0	0	1	0	ADCAnI2S (Physical channel ANIn18)
0	1	0	0	1	1	ADCAnI3S (Physical channel ANIn19)
0	1	0	1	0	0	ADCAnI4S (Physical channel ANIn20)
0	1	0	1	0	1	ADCAnI5S (Physical channel ANIn21)
0	1	0	1	1	0	ADCAnI6S (Physical channel ANIn22)
0	1	0	1	1	1	ADCAnI7S (Physical channel ANIn23)
0	1	1	0	0	0	ADCAnI8S (Physical channel ANIn24)
0	1	1	0	0	1	ADCAnI9S (Physical channel ANIn25)
0	1	1	0	1	0	ADCAnI10S (Physical channel ANIn26)
0	1	1	0	1	1	ADCAnI11S (Physical channel ANIn27)
0	1	1	1	0	0	ADCAnI12S (Physical channel ANIn28)
0	1	1	1	0	1	ADCAnI13S (Physical channel ANIn29)
0	1	1	1	1	0	ADCAnI14S (Physical channel ANIn30)
0	1	1	1	1	1	ADCAnI15S (Physical channel ANIn31)
1	0	0	0	0	0	ADCAnI16S (Physical channel ANIn32)
1	0	0	0	0	1	ADCAnI17S (Physical channel ANIn33)
1	0	0	0	1	0	ADCAnI18S (Physical channel ANIn34)
1	0	0	0	1	1	ADCAnI19S (Physical channel ANIn35)
1	0	0	1	0	0	Diagnosis channel for A/D converter
Other than above						Setting prohibited

38.3.2.2 ADCAnPWDVCR — PWM-Diag Virtual Channel Register

This register is used to indicate virtual channel setting (PWSAnPVCRx_y register setting) of the PWM-Diag (SG4).

Access:	ADCAnPWDVCR register is a read-only register that can be read in 32-bit units. ADCAnPWDVCRL register is a read-only register that can be read in 16-bit units. ADCAnPWDVCRLL register is a read-only register that can be read in 8-bit units. ADCAnPWDVCRLH register is a read-only register that can be read in 8-bit units.
Address:	ADCAnPWDVCR: <ADCAn_base> + 0F4$_{\text{H}}$ ADCAnPWDVCRL: <ADCAn_base> + 0F4$_{\text{H}}$ ADCAnPWDVCRLL: <ADCAn_base> + 0F4$_{\text{H}}$ ADCAnPWDVCRLH: <ADCAn_base> + 0F4$_{\text{H}}$ + 1$_{\text{H}}$
Value after reset:	0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE *1	MPXV[2:0]*1			ULS[3:2]		—	—	ULS[1:0]		GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.39 ADCAnPWDVCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	MPXE*1	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[27] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[11] (even channel) bit. MPX Enable Set this bit to 1 when an external analog multiplexer is used. 0: The use of MPX is prohibited. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0]*1	These bits are used to set the MPX value to be transferred to an external analog multiplexer by using the PWSAnPVCRx_y.PWSAnVRDTy[26:24] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[10:8] (even channel) bit.
11, 10	ULS[3:2]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[29:28] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[13:12] (even channel) bit. Upper Limit/Lower Limit Table Select*2
9, 8	Reserved	When read, the value after reset is returned.
7, 6	ULS[1:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[23:22] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[7:6] (even channel) bit. Upper Limit/Lower Limit Table Select*2
5 to 0	GCTRL[5:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[21:16] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[5:0] (even channel) bit. Physical Channel Select These bits are used to specify a physical channel to be assigned to virtual channel j. For the selection of the channel, see **Table 38.38, Selection of Physical Channels**.

Note 1. These bits are only supported for ADCA0. For ADCA1, when read, the value after reset is returned.

Note 2. ULS[3:0] =
0000: Upper limit and lower limit are not checked.
0001: Upper limit and lower limit are checked for ADCAnULLMTBR0.
0010: Upper limit and lower limit are checked for ADCAnULLMTBR1.
0011: Upper limit and lower limit are checked for ADCAnULLMTBR2.
0100: Upper limit and lower limit are checked for ADCAnULLMTBR3.
0101: Upper limit and lower limit are checked for ADCAnULLMTBR4.
0110: Upper limit and lower limit are checked for ADCAnULLMTBR5.
0111: Upper limit and lower limit are checked for ADCAnULLMTBR6.
1000: Upper limit and lower limit are checked for ADCAnULLMTBR7.
Other than above : Setting prohibited (Upper limit and lower limit are not checked when set)

38.3.2.3 ADCAnDRj — Data Register j

This register is a 32-/16-bit read-only register that stores the A/D conversion results corresponding to ADCAnVCRj and ADCAnVCR(j+1). As the A/D conversion results, the conversion result for ADCAnVCR(j+1) is stored in the upper bits (ADCAnDR(j+1)), and the conversion result for ADCAnVCRj is stored in the lower bits (ADCAnDRj).

Access:	ADCAnDRj register is a read-only register that can be read in 32-bit units. ADCAnDRjL and ADCAnDRjH registers are read-only registers that can be read in 16-bit units.
Address:	ADCAnDRj: <ADCAn_base> + 100_H + j × 2_H ADCAnDRjL: <ADCAn_base> + 100_H + j × 2_H ADCAnDRjH: <ADCAn_base> + 100_H + j × 2_H + 2_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DR(j+1)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRj[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.40 ADCAnDRj Register Contents

Bit Position	Bit Name	Function
31 to 16	DR(j+1)[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCAnVCR(j+1) are transferred.)
15 to 0	DRj[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCAnVCRj are transferred.)

CAUTION

ADCAnDIRj.WFLG and ADCAnDIR(j+1).WFLG are cleared by reading ADCAnDRjL or ADCAnDRjH in 16-bit units at this time. If virtual channels 33, 34, and 35 are used exclusively for self-diagnosis, the lower-order bits (DRj[15:0]) for channel 32 cannot be used.

NOTES

1. j = 00, 02, …, 46, 48 (for ADCA0)
   j = 00, 02, …, 32, 34 (for ADCA1)
2. By controlling ADCAnADCR.CRAC and ADCAnADCR.CTYP, the data format of this register becomes as follows:
   - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 0 → Right alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 27 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 11 to 0.
   - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 1 → Left alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 20, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 4.

- ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 0 → Right alignment is used.
  →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 25 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 9 to 0.
- ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 1 → Left alignment is used.
  →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 22, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 6.

38.3.2.4 ADCAnDIRj — Data Supplementary Information Register j

This register is a 32-bit read-only register that stores the A/D conversion result for ADCAnDRj and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnDRj value is transferred. As information incidental to the A/D converted value, information about the write flag (WFLG), the MPX value (MPXV[2:0]), and the physical channel (ID[5:0]) is transferred. The data format of the A/D conversion result stored in ADCAnDIRj is the same as that for the ADCAnDRj register.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	<ADCAn_base> + 200_H + j × 4_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE *1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.41 ADCAnDIRj Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: MPX function is not used. 1: MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnDRj or ADCAnDIRj is read (cleared when read). 1: A/D converted value is stored in ADCAnDRj (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the most recent conversion result.
15 to 0	DR[15:0]	These bits are used to store the A/D conversion result.

Note 1. These bits are only supported by ADCA0.
For ADCA1, when read, the value after reset is returned.

38.3.2.5 ADCAnPWDTSNDR — PWM-Diag Data Register

This register is a 32-/16-bit read-only register that stores the A/D conversion results corresponding to the PWM-Diag. As the A/D conversion results, the conversion result for the PWM-Diag (PWDDR) is stored in the upper bits.

Access:	ADCAnPWDTSNDR register is a read-only register that can be read in 32-bit units. ADCAnPWDTSNDRH register is a read-only register that can be read in 16-bit units.
Address:	ADCAnPWDTSNDR: <ADCAn_base> + 178_H ADCAnPWDTSNDRH: <ADCAn_base> + $17A_H$
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.42 ADCAnPWDTSNDR Register Contents

Bit Position	Bit Name	Function
31 to 16	PWDDR[15:0]	These bits are used to store the A/D conversion result data for the PWM-Diag.
15 to 0	Reserved	When read, the value after reset is returned.

NOTE

The data format of this register is controlled by ADCAnADCR.CRAC and ADCAnADCR.CTYP, as shown below.

- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 0 → Right alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 27 to 16.
- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 1 → Left alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 20.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 0 → Right alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 25 to 16.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 1 → Left alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 22.

38.3.2.6 ADCAnPWDDIR — PWM-Diag Data Supplementary Information Register

This register is a 32-bit read-only register that stores the A/D conversion result when PWM-Diag is used, and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnPWDSNDR.PWDDR[15:0] value is transferred. As supplementary information to the A/D converted value, the write flag (WFLG), MPX value (MPXV[2:0]), and physical channel (ID[5:0]) are transferred. The data format of the A/D conversion result stored in ADCAnPWDDIR is the same as that for the ADCAnPWDTSNDR register. In addition, PWDDR[15:0] can be read with corresponded PWGA channel by PWSAnPWDDIRz register. For details, see **Section 37, PWM Output/Diagnostic (PWM-Diag)**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCAn_base> + 2F4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE*1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.43 ADCAnPWDDIR Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: The MPX function is not used. 1: The MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnPWDTSNDR or ADCAnPWDDIR is read (cleared when read). 1: The A/D converted value is stored in ADCAnPWDTSNDR (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits are used to store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the physical channel number corresponding to the most recent conversion result.
15 to 0	PWDDR[15:0]	These bits are used to store the A/D conversion result for PWM-Diag.

Note 1. These bits are only supported for ADCA0.
For ADCA1, when read, the value after reset is returned.

38.3.2.7 ADCAnADHALTR — A/D Force Halt Register

This register is used to halt conversion for all SGs of ADCAn. The read value is always 0.

Access:	ADCAnADHALTR register is a write-only register that can be written in 32-bit units. ADCAnADHALTRL register is a write-only register that can be written in 16-bit units. ADCAnADHALTRLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnADHALTR:<ADCAn_base> + 300_H ADCAnADHALTRL:<ADCAn_base> + 300_H ADCAnADHALTRLL:<ADCAn_base> + 300_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HALT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.44 ADCAnADHALTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HALT	ADCA Force Halt Trigger All scan groups are halted and initialized, and ADCA becomes idle state. Writing of 0: No effect Writing of 1: Scan groups are halted.

38.3.2.8 ADCAnADCR — A/D Control Register

This register is used for ADCAn common control.

Access: ADCAnADCR register can be read or written in 32-bit units.
ADCAnADCRL register can be read or written in 16-bit units.
ADCAnADCRLL register can be read or written in 8-bit units.

Address: ADCAnADCR: <ADCAn_base> + 304_H
ADCAnADCRL: <ADCAn_base> + 304_H
ADCAnADCRLL: <ADCAn_base> + 304_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DGON	—	CRAC	CTYP	—	—	SUSMTD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R/W	R/W

Table 38.45 ADCAnADCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DGON	Self-Diagnostic Voltage Standby Control 0: The self-diagnostic voltage circuit is turned off. 1: The self-diagnostic voltage circuit is turned on, or the reference voltage is updated.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CRAC	Alignment Control 0: The results of conversion to PWDDR and ADCAnDRj are stored right-aligned. 1: The results of conversion to PWDDR and ADCAnDRj are stored left-aligned.
4	CTYP	12/10 Bit Select Mode 0: 12-bit mode 1: 10-bit mode
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SUSMTD [1:0]	Suspend Mode Select These bits are used to select the suspend method when a higher-priority scan group interrupts a lower-priority scan group. 00: Synchronous suspend when a higher-priority SG or SVSTOP interrupts. 01: Asynchronous suspend when a higher-priority SG (SG2, SG3, SG4) and SVSTOP interrupt SG1, and synchronous suspend when a higher-priority SG (SG3, SG4) and SVSTOP interrupt SG2, or when a higher-priority SG (SG4) and SVSTOP interrupt SG3. 10: Asynchronous suspend when a higher-priority SG or SVSTOP interrupts. 11: Setting prohibited

CAUTION

To prevent malfunction, ADCAnADCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

- Synchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the A/D conversion for the higher-priority SG is performed after the on-going A/D conversion of a channel is completed. After processing for the higher-priority SG is completed, the suspended A/D channel processing for the lower-priority SG is resumed.
- Asynchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the on-going channel processing is suspended, and then the A/D conversion for the higher-priority SG is performed. After processing for the higher-priority SG is completed, the suspended A/D channel conversion for the lower-priority SG is resumed.

For details, see **Figure 38.25, Example of Synchronous Suspend and Resume Operation** and **Figure 38.26, Example of Asynchronous Suspend and Resume Operation**.

38.3.2.9 ADCAnMPXCURR — MPX Current Register

This register is used to store the MPX value for an external analog multiplexer.

Access:	ADCAnMPXCURR register is a read-only register that can be read in 32-bit units. ADCAnMPXCURRL register is a read-only register that can be read in 16-bit units. ADCAnMPXCURRLL register is a read-only register that can be read in 8-bit units.
Address:	ADCAnMPXCURR: <ADCAn_base> + 30C$_{H}$ ADCAnMPXCURRL: <ADCAn_base> + 30C$_{H}$ ADCAnMPXCURRLL: <ADCAn_base> + 30C$_{H}$
Value after reset:	0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MPXCUR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.46 ADCAnMPXCURR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	MPXCUR[2:0]	These bits are used to store the current MPX value. If conversion of a virtual channel starts after setting ADCAnVCRj.MPXE to 1, the setting of ADCAnVCRj.MPXV[2:0] is stored. If conversion of a virtual channel starts after setting ADCAnPWDVCR.MPXE to 1, the setting of ADCAnPWDVCR.MPXV[2:0] is stored.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.10 ADCAnTHSMPSTCR — T&H Sampling Start Control Register

This register is used to control the start of sampling for all T&Hk (k = 0 to 5). The bits are always read as 0.

Access:	ADCAnTHSMPSTCR register is a write-only register that can be written in 32-bit units. ADCAnTHSMPSTCRL register is a write-only register that can be written in 16-bit units. ADCAnTHSMPSTCRLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnTHSMPSTCR: <ADCAn_base> + 314_H ADCAnTHSMPSTCRL: <ADCAn_base> + 314_H ADCAnTHSMPSTCRLL: <ADCAn_base> + 314_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SMPST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.47 ADCAnTHSMPSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SMPST	T&H Sampling Start Control Trigger 0: No effect 1: Sampling for all T&H is started. The conditions to place the T&H circuit in the sampling state are as follows: ● Condition to start sampling while T&H is stopped: 1 being written to ADCAnTHSMPSTCR.SMPST while ADCAnTHER.THkE = 1 (k = 0 to 5). ● Condition to start continuous sampling in automatic sampling: A/D conversion of the hold value for T&Hk being completed while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHCR.ASMPMSK = 1.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.11 ADCAnTHCR — T&H Control Register

This register controls the sampling transition after A/D conversion of the hold value for T&H is completed.

Automatic start of sampling on the T&H circuit after A/D conversion of the hold value for T&H is completed shortens the time required for the generation of succeeding hold completion triggers.

Access:	ADCAnTHCR register can be read or written in 32-bit units. ADCAnTHCRL register can be read or written in 16-bit units. ADCAnTHCRLL register can be read or written in 8-bit units.
Address:	ADCAnTHCR: <ADCAn_base> + 318$_H$ ADCAnTHCRL: <ADCAn_base> + 318$_H$ ADCAnTHCRLL: <ADCAn_base> + 318$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASMPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.48 ADCAnTHCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ASMPMSK	Automatic Sampling Mask Control 0: Automatic sampling is not performed. 1: Automatic sampling is performed.

CAUTION

To prevent malfunction, ADCAnTHCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.12 ADCAnTHAHLDSTCR — T&H Group A Hold Start Control Register

This register is used to control the start of the hold for T&H group A. The bits are always read as 0.

Access: ADCAnTHAHLDSTCR register is a write-only register that can be written in 32-bit units.
ADCAnTHAHLDSTCRL register is a write-only register that can be written in 16-bit units.
ADCAnTHAHLDSTCRLL register is a write-only register that can be written in 8-bit units.

Address: ADCAnTHAHLDSTCR: <ADCAn_base> + $31C_H$
ADCAnTHAHLDSTCRL: <ADCAn_base> + $31C_H$
ADCAnTHAHLDSTCRLL: <ADCAn_base> + $31C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.49 ADCAnTHAHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group A Hold Start Control Trigger 0: No effect 1: Hold for T&H group A is started. The condition to place T&H group A in the hold state is as follows: 1 being written to ADCAnTHAHLDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 0 (k = 0 to 5).

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.13 ADCAnTHBHLDSTCR — T&H Group B Hold Start Control Register

This register is used to control the start of the hold for T&H group B. The bits are always read as 0.

Access:	ADCAnTHBHLDSTCR register is a write-only register that can be written in 32-bit units. ADCAnTHBHLDSTCRL register is a write-only register that can be written in 16-bit units. ADCAnTHBHLDSTCRLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnTHBHLDSTCR: <ADCAn_base> + 320_H ADCAnTHBHLDSTCRL: <ADCAn_base> + 320_H ADCAnTHBHLDSTCRLL: <ADCAn_base> + 320_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.50 ADCAnTHBHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group B Hold Start Control Trigger 0: No effect 1: Hold for T&H group B is started. The condition to place T&H group B in the hold state is as follows: 1 being written to ADCAnTHBHLDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 1 (k = 0 to 5).

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.14 ADCAnTHACR — T&H Group A Control Register

This register is used to control T&H group A.

Access:	ADCAnTHACR register can be read or written in 32-bit units. ADCAnTHACRL register can be read or written in 16-bit units. ADCAnTHACRLL register can be read or written in 8-bit units.
Address:	ADCAnTHACR: <ADCAn_base> + 324_H ADCAnTHACRL: <ADCAn_base> + 324_H ADCAnTHACRLL: <ADCAn_base> + 324_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.51 ADCAnTHACR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group A Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds. **NOTE:** The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group A Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group A. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group A. **NOTE:** ADCAnTHAHLDSTCR.HLDST becomes a hold start trigger regardless of the ADCAnTHACR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SGS[1:0]	T&H Group A Scan Group Select 00: No scan group is selected for T&H group A. 01: SG1 is selected for T&H group A. 10: SG2 is selected for T&H group A. 11: SG3 is selected for T&H group A. **NOTES:** 1. If ADCAnTHACR.SGS[1:0] is set to 0_H, T&H does not operate. When you enable T&Hk in ADCAnTHER.THkE, make sure that a scan group is specified in SGS[1:0]. 2. Selecting the same scan group as T&H group B is prohibited.

CAUTION

To prevent malfunction, ADCAnTHACR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.15 ADCAnTHBCR — T&H Group B Control Register

This register is used to control T&H group B.

Access: ADCAnTHBCR register can be read or written in 32-bit units.
ADCAnTHBCRL register can be read or written in 16-bit units.
ADCAnTHBCRLL register can be read or written in 8-bit units.

Address: ADCAnTHBCR: <ADCAn_base> + 328_H
ADCAnTHBCRL: <ADCAn_base> + 328_H
ADCAnTHBCRLL: <ADCAn_base> + 328_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.52 ADCAnTHBCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group B Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds. **NOTE:** The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group B Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group B. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group B. **NOTE:** ADCAnTHBHLDSTCR.HLDST becomes a hold start trigger regardless of the ADCAnTHBCR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SGS[1:0]	T&H Group B Scan Group Select 00: No scan group is selected for T&H group B. 01: SG1 is selected for T&H group B. 10: SG2 is selected for T&H group B. 11: SG3 is selected for T&H group B. **NOTES:** 1. If ADCAnTHBCR.SGS[1:0] is set to 0_H, T&H does not operate. When you enable T&Hk in ADCAnTHER.THkE, make sure that a scan group is specified in SGS[1:0]. 2. Selecting the same scan group as T&H group A is prohibited.

CAUTION

To prevent malfunction, ADCAnTHBCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.16 ADCAnTHER — T&H Enable Register

This register controls enabling and disabling of each T&H.

Access: ADCAnTHER register can be read or written in 32-bit units.
ADCAnTHERL register can be read or written in 16-bit units.
ADCAnTHERLL register can be read or written in 8-bit units.

Address: ADCAnTHER: <ADCAn_base> + 32C$_H$
ADCAnTHERL: <ADCAn_base> + 32C$_H$
ADCAnTHERLL: <ADCAn_base> + 32C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TH5E	TH4E	TH3E	TH2E	TH1E	TH0E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.53 ADCAnTHER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5E	T&H5 Enable 0: T&H5 is disabled. 1: T&H5 is enabled. **NOTE:** If TH5E is set to 0, T&H5 is always stopped.
4	TH4E	T&H4 Enable 0: T&H4 is disabled. 1: T&H4 is enabled. **NOTE:** If TH4E is set to 0, T&H4 is always stopped.
3	TH3E	T&H3 Enable 0: T&H3 is disabled. 1: T&H3 is enabled. **NOTE:** If TH3E is set to 0, T&H3 is always stopped.
2	TH2E	T&H2 Enable 0: T&H2 is disabled. 1: T&H2 is enabled **NOTE:** If TH2E is set to 0, T&H2 is always stopped.
1	TH1E	T&H1 Enable 0: T&H1 is disabled. 1: T&H1 is enabled. **NOTE:** If TH1E is set to 0, T&H1 is always stopped.
0	TH0E	T&H0 Enable 0: T&H0 is disabled. 1: T&H0 is enabled **NOTE:** If TH0E is set to 0, T&H0 is always stopped.

CAUTION

To prevent malfunction, ADCAnTHER should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.17 ADCAnTHGSR — T&H Group Select Register

This register is used to select a T&H group for each T&H.

Access:	ADCAnTHGSR register can be read or written in 32-bit units. ADCAnTHGSRL register can be read or written in 16-bit units. ADCAnTHGSRLL register can be read or written in 8-bit units.
Address:	ADCAnTHGSR: <ADCAn_base> + 330$_H$ ADCAnTHGSRL: <ADCAn_base> + 330$_H$ ADCAnTHGSRLL: <ADCAn_base> + 330$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TH5GS	TH4GS	TH3GS	TH2GS	TH1GS	TH0GS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.54 ADCAnTHGSR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5GS	T&H5 Group Select 0: T&H5 is selected to group A. 1: T&H5 is selected to group B.
4	TH4GS	T&H4 Group Select 0: T&H4 is selected to group A. 1: T&H4 is selected to group B.
3	TH3GS	T&H3 Group Select 0: T&H3 is selected to group A. 1: T&H3 is selected to group B.
2	TH2GS	T&H2 Group Select 0: T&H2 is selected to group A. 1: T&H2 is selected to group B.
1	TH1GS	T&H1 Group Select 0: T&H1 is selected to group A. 1: T&H1 is selected to group B.
0	TH0GS	T&H0 Group Select 0: T&H0 is selected to group A. 1: T&H0 is selected to group B.

CAUTION

- Do not set T&H0 to T&H2 to the same group as T&H3 to T&H5.
  Example
  - Group A: 0ch, 1ch, 2ch
    Group B: 3ch, 4ch, 5ch → Setting allowed
  - Group A: 0ch
    Group B: 1ch, 2ch → Setting allowed
  - Group A: 0ch, 1ch, 3ch
    Group B: 2ch, 4ch → Setting prohibited
- To prevent malfunction, ADCAnTHGSR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.18 ADCAnSMPCR — Sampling Control Register

This register controls sampling.

ADCAnSMPCR controls the sampling time for SG4 (PWM-Diag) and SG1 to SG3.

Access:	ADCAnSMPCR register can be read or written in 32-bit units. ADCAnSMPCRL register can be read or written in 16-bit units. ADCAnSMPCRLL register can be read or written in 8-bit units.
Address:	ADCAnSMPCR: <ADCAn_base> + 380_H ADCAnSMPCRL: <ADCAn_base> + 380_H ADCAnSMPCRLL: <ADCAn_base> + 380_H
Value after reset:	0000 0018_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMPT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.55 ADCAnSMPCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	SMPT[7:0]	These bits are used to set the sampling time (the number of cycles). 12_H: 18 cycles (ADCLK = 8 MHz to 32 MHz) 18_H: 24 cycles (ADCLK = 8 MHz to 40 MHz) Settings other than above are prohibited.

CAUTION

- To prevent malfunction, ADCAnSMPCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- When SMPT is changed, the A/D conversion wait time is also changed when MPX is used by virtual channel register j (ADCAnVCRj) or PWM-Diag virtual channel register (ADCAnPWDVCR).

38.3.2.19 ADCAnMPXSTBTSELR0 to 4 — MPX Stabilization Time Selection Register 0 to 4

This register is used to select the stabilization time for respective physical channels when external multiplexer is used. ADCAnMPXSTBTSELR0 corresponds to physical channels ANIn00 to ANIn07, ADCAnMPXSTBTSELR1 corresponds to physical channels of ANIn08 to ANIn15, ADCAnMPXSTBTSELR2 corresponds to physical channels of ANIn16 to ANIn23, and ADCAnMPXSTBTSELR3 corresponds to physical channels of ANIn24 to ANIn31, and ADCAnMPXSTBTSELR4 corresponds to physical channels of ANIn32 to ANIn35.

Access: ADCAnMPXSTBTSELR0 to 4 registers can be read or written in 32-bit units.
ADCAnMPXSTBTSELR0L to 4L registers can be read or written in 16-bit units.
ADCAnMPXSTBTSELR0H to 3H registers can be read or written in 16-bit units.

Address: ADCAnMPXSTBTSELR0: <ADCAn_base> + 5F0$_{H}$
ADCAnMPXSTBTSELR1: <ADCAn_base> + 5F4$_{H}$
ADCAnMPXSTBTSELR2: <ADCAn_base> + 5F8$_{H}$
ADCAnMPXSTBTSELR3: <ADCAn_base> + 5FC$_{H}$
ADCAnMPXSTBTSELR4: <ADCAn_base> + 600$_{H}$

ADCAnMPXSTBTSELR0L: <ADCAn_base> + 5F0$_{H}$
ADCAnMPXSTBTSELR1L: <ADCAn_base> + 5F4$_{H}$
ADCAnMPXSTBTSELR2L: <ADCAn_base> + 5F8$_{H}$
ADCAnMPXSTBTSELR3L: <ADCAn_base> + 5FC$_{H}$
ADCAnMPXSTBTSELR4L: <ADCAn_base> + 600$_{H}$

ADCAnMPXSTBTSELR0H: <ADCAn_base> + 5F0$_{H}$ + 2$_{H}$
ADCAnMPXSTBTSELR1H: <ADCAn_base> + 5F4$_{H}$ + 2$_{H}$
ADCAnMPXSTBTSELR2H: <ADCAn_base> + 5F8$_{H}$ + 2$_{H}$
ADCAnMPXSTBTSELR3H: <ADCAn_base> + 5FC$_{H}$ + 2$_{H}$

Value after reset: 0000 0000$_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	STBTCHhh[2:0]			—	STBTCHgg[2:0]			—	STBTCHff[2:0]			—	STBTCHee[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	STBTCHdd[2:0]			—	STBTCHcc[2:0]			—	STBTCHbb[2:0]			—	STBTCHaa[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 38.56 ADCAnMPXSTBTSELR0 to 4 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write value after reset.
30 to 28	STBTCHhh[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
27	Reserved	When read, the value after reset is returned. When writing, write value after reset.
26 to 24	STBTCHgg[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
23	Reserved	When read, the value after reset is returned. When writing, write value after reset.
22 to 20	STBTCHff[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
19	Reserved	When read, the value after reset is returned. When writing, write value after reset.
18 to 16	STBTCHee[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
15	Reserved	When read, the value after reset is returned. When writing, write value after reset.
14 to 12	STBTCHdd[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
11	Reserved	When read, the value after reset is returned. When writing, write value after reset.
10 to 8	STBTCHcc[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
7	Reserved	When read, the value after reset is returned. When writing, write value after reset.
6 to 4	STBTCHbb[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].
3	Reserved	When read, the value after reset is returned. When writing, write value after reset.
2 to 0	STBTCHaa[2:0]*[1]	Select the value from ADCAnMPXSTBTR0 to 7 for MPX stabilization time*[2].

Note 1. The relationship between physical channel and register symbol is shown below table. ADCAn012S and ADCA0I13S are not supported on RH850/F1KH-D8, RH850/F1KM-S4. When writing corresponding bits (STBTCH28[2:0] and STBTCH29[2:0] of ADCA0MPXSTBTSELR3 register), write value after reset.

Note 2. STBTCH00[2:0] to STBTCH35[2:0]=
000: The set value of ADCAnMPXSTBTR0 is used for MPX stabilization time
001: The set value of ADCAnMPXSTBTR1 is used for MPX stabilization time
010: The set value of ADCAnMPXSTBTR2 is used for MPX stabilization time
011: The set value of ADCAnMPXSTBTR3 is used for MPX stabilization time
100: The set value of ADCAnMPXSTBTR4 is used for MPX stabilization time
101: The set value of ADCAnMPXSTBTR5 is used for MPX stabilization time
110: The set value of ADCAnMPXSTBTR6 is used for MPX stabilization time
111: The set value of ADCAnMPXSTBTR7 is used for MPX stabilization time

Physical Channel		Selection Register	Symbol	
ANIn00	ADCAnI0	ADCAnMPXSTBTSELR0	aa	00
ANIn08	ADCAnI8	ADCAnMPXSTBTSELR1		08
ANIn16	ADCAnI0S	ADCAnMPXSTBTSELR2		16
ANIn24	ADCAnI8S	ADCAnMPXSTBTSELR3		24
ANIn32	ADCAnI16S	ADCAnMPXSTBTSELR4		32
ANIn01	ADCAnI1	ADCAnMPXSTBTSELR0	bb	01
ANIn09	ADCAnI9	ADCAnMPXSTBTSELR1		09
ANIn17	ADCAnI1S	ADCAnMPXSTBTSELR2		17
ANIn25	ADCAnI9S	ADCAnMPXSTBTSELR3		25
ANIn33	ADCAnI17S	ADCAnMPXSTBTSELR4		33
ANIn02	ADCAnI2	ADCAnMPXSTBTSELR0	cc	02
ANIn10	ADCAnI10	ADCAnMPXSTBTSELR1		10
ANIn18	ADCAnI2S	ADCAnMPXSTBTSELR2		18
ANIn26	ADCAnI10S	ADCAnMPXSTBTSELR3		26
ANIn34	ADCAnI18S	ADCAnMPXSTBTSELR4		34
ANIn03	ADCAnI3	ADCAnMPXSTBTSELR0	dd	03
ANIn11	ADCAnI11	ADCAnMPXSTBTSELR1		11
ANIn19	ADCAnI3S	ADCAnMPXSTBTSELR2		19
ANIn27	ADCAnI11S	ADCAnMPXSTBTSELR3		27
ANIn35	ADCAnI19S	ADCAnMPXSTBTSELR4		35
ANIn04	ADCAnI4	ADCAnMPXSTBTSELR0	ee	04
ANIn12	ADCAnI12	ADCAnMPXSTBTSELR1		12
ANIn20	ADCAnI4S	ADCAnMPXSTBTSELR2		20
ANIn28	ADCAnI12S	ADCAnMPXSTBTSELR3		28
ANIn05	ADCAnI5	ADCAnMPXSTBTSELR0	ff	05
ANIn13	ADCAnI13	ADCAnMPXSTBTSELR1		13
ANIn21	ADCAnI5S	ADCAnMPXSTBTSELR2		21
ANIn29	ADCAnI13S	ADCAnMPXSTBTSELR3		29
ANIn06	ADCAnI6	ADCAnMPXSTBTSELR0	gg	06
ANIn14	ADCAnI14	ADCAnMPXSTBTSELR1		14
ANIn22	ADCAnI6S	ADCAnMPXSTBTSELR2		22
ANIn30	ADCAnI14S	ADCAnMPXSTBTSELR3		30
ANIn07	ADCAnI7	ADCAnMPXSTBTSELR0	hh	07
ANIn15	ADCAnI15	ADCAnMPXSTBTSELR1		15
ANIn23	ADCAnI7S	ADCAnMPXSTBTSELR2		23
ANIn31	ADCAnI15S	ADCAnMPXSTBTSELR3		31

NOTE

- STBTCH00[2:0] to STBTCH35[2:0] are enabled when the MPXE bit of the ADCAnVCRj or ADCAnPWDVCR register is set to 1.
- In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

CAUTION

To prevent malfunction, ADCAnMPXSTBTSELR0 to ADCAnMPXSTBTSELR4 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

38.3.2.20 ADCAnMPXSTBTR0 to 7 — MPX Stabilization Time Setting Register 0 to 7

This register is used to set the stabilization time when external multiplexer is used.

Stabilization time of corresponded physical channels which is selected by ADCAnMPXSTBTSEL0 to 4 register is set by setting this register.

Access: ADCAnMPXSTBTR0 to 7 registers can be read or written in 32-bit units.
ADCAnMPXSTBTR0L to 7L registers can be read or written in 16-bit units.

Address:
ADCAnMPXSTBTR0: <ADCAn_base> + 610_H
ADCAnMPXSTBTR1: <ADCAn_base> + 614_H
ADCAnMPXSTBTR2: <ADCAn_base> + 618_H
ADCAnMPXSTBTR3: <ADCAn_base> + $61C_H$
ADCAnMPXSTBTR4: <ADCAn_base> + 620_H
ADCAnMPXSTBTR5: <ADCAn_base> + 624_H
ADCAnMPXSTBTR6: <ADCAn_base> + 628_H
ADCAnMPXSTBTR7: <ADCAn_base> + $62C_H$

ADCAnMPXSTBTR0L: <ADCAn_base> + 610_H
ADCAnMPXSTBTR1L: <ADCAn_base> + 614_H
ADCAnMPXSTBTR2L: <ADCAn_base> + 618_H
ADCAnMPXSTBTR3L: <ADCAn_base> + $61C_H$
ADCAnMPXSTBTR4L: <ADCAn_base> + 620_H
ADCAnMPXSTBTR5L: <ADCAn_base> + 624_H
ADCAnMPXSTBTR6L: <ADCAn_base> + 628_H
ADCAnMPXSTBTR7L: <ADCAn_base> + $62C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	STBTIME[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.57 ADCAnMPXSTBR0 to 7 Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write value after reset.
13 to 0	STBTIME[13:0]	Specify the analog input stabilization time when an external analog multiplexer is used only. When these bits are all “0” (reset value), wait time of one A/D conversion is inserted. Other than all “0”, wait time is inserted before A/D conversion according to the calculation below. Calculation formula: STBTIME[13:0] / ADCLK Refer to **Table 38.58, Setting Example of ADCAnMPXSTBR0 to 7 Register** below for as example. Stabilization time has to be set within from 1μs to 250μs.

Table 38.58 Setting Example of ADCAnMPXSTBR0 to 7 Register

Stabilization Time (µs)	ADCLK (MHz)						
	8	10	12	16	20	24	40
1	0008_H	$000A_H$	$000C_H$	0010_H	0014_H	0018_H	0028_H
100	0320_H	$03E8_H$	$04B0_H$	0640_H	$07D0_H$	0960_H	$0FA0_H$
250	$07D0_H$	$09C4_H$	$0BB8_H$	$0FA0_H$	1388_H	1770_H	2710_H

Note 1. This setting should be changed in accordance with A/D conversion clock selection.

CAUTION

To prevent malfunction, ADCAnMPXSTBTR0 to ADCAnMPXSTBTR7 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1KH, RH850/F1KM, only ADCA0 supports this function.

38.3.2.21 ADCAnSFTCR — Safety Control Register

This is a register for safety control.

Access:	ADCAnSFTCR register can be read or written in 32-bit units. ADCAnSFTCRL register can be read or written in 16-bit units. ADCAnSFTCRLL register can be read or written in 8-bit units.
Address:	ADCAnSFTCR: <ADCAn_base> + 334_{H} ADCAnSFTCRL: <ADCAn_base> + 334_{H} ADCAnSFTCRLL: <ADCAn_base> + 334_{H}
Value after reset:	0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RDCLRE	ULEIE	OWEIE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 38.59 ADCAnSFTCR Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	RDCLRE	Read & Clear Enable When the A/D conversion result is read, this bit selects whether the A/D conversion result is cleared by hardware. 0: ADCAnPWDTSNDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are not cleared when ADCAnPWDTSNDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. 1: ADCAnPWDTSNDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are cleared when ADCAnPWDTSNDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. WFLG of ADCAnDIRj is cleared regardless of the RDCLRE setting when ADCAnDRj or ADCAnDIRj is read. Even when reading with 16-bit access to ADCAnDRj or ADCAnPWDTSNDR is performed with the setting of RDCLRE = 1, all 32-bit data will be cleared.
3	ULEIE	A/D Error Interrupt (INT_ADE) Enable on Upper/Lower Limit Error Detection 0: Disabled 1: Enabled
2	OWEIE	A/D Error Interrupt (INT_ADE) Enable on Overwrite Error Detection 0: Disabled 1: Enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

To prevent malfunction, ADCAnSFTCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

38.3.2.22 ADCAnULLMTBR0 to 7 — Upper Limit/Lower Limit Table Register 0 to 7

These registers are used to set the threshold for detection of an upper limit or lower limit error in the A/D converted value. Any of ADCAnULLMTBR0 to ADCAnULLMTBR7 is specified by setting ADCAnPWDVCR.ULS[3:0] and ADCAnVCRj.ULS[3:0] and compared with ADCAnPWDTSNDR and ADCAnDRj.

Access: ADCAnULLMTBR0 to 7 registers can be read or written in 32-bit units.
ADCAnULLMTBR0L to 7L and ADCAnULLMTBR0H to 7H registers can be read or written in 16-bit units.

Address: ADCAnULLMTBR0: <ADCAn_base> + 338_{H}
ADCAnULLMTBR1: <ADCAn_base> + $33C_{H}$
ADCAnULLMTBR2: <ADCAn_base> + 340_{H}
ADCAnULLMTBR3: <ADCAn_base> + 564_{H}
ADCAnULLMTBR4: <ADCAn_base> + 568_{H}
ADCAnULLMTBR5: <ADCAn_base> + $56C_{H}$
ADCAnULLMTBR6: <ADCAn_base> + 570_{H}
ADCAnULLMTBR7: <ADCAn_base> + 574_{H}

ADCAnULLMTBR0L: <ADCAn_base> + 338_{H}
ADCAnULLMTBR1L: <ADCAn_base> + $33C_{H}$
ADCAnULLMTBR2L: <ADCAn_base> + 340_{H}
ADCAnULLMTBR3L: <ADCAn_base> + 564_{H}
ADCAnULLMTBR4L: <ADCAn_base> + 568_{H}
ADCAnULLMTBR5L: <ADCAn_base> + $56C_{H}$
ADCAnULLMTBR6L: <ADCAn_base> + 570_{H}
ADCAnULLMTBR7L: <ADCAn_base> + 574_{H}

ADCAnULLMTBR0H: <ADCAn_base> + 338_{H} + 2_{H}
ADCAnULLMTBR1H: <ADCAn_base> + $33C_{H}$+ 2_{H}
ADCAnULLMTBR2H: <ADCAn_base> + 340_{H} + 2_{H}
ADCAnULLMTBR3H: <ADCAn_base> + 564_{H} + 2_{H}
ADCAnULLMTBR4H: <ADCAn_base> + 568_{H} + 2_{H}
ADCAnULLMTBR5H: <ADCAn_base> + $56C_{H}$ + 2_{H}
ADCAnULLMTBR6H: <ADCAn_base> + 570_{H} + 2_{H}
ADCAnULLMTBR7H: <ADCAn_base> + 574_{H} + 2_{H}

Value after reset: FFF0 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULMTB[11:0]												—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLMTB[11:0]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 38.60 ADCAnULLMTBR0 to 7 Registers Contents

Bit Position	Bit Name	Function
31 to 20	ULMTB[11:0]	Upper Limit Table Specify the threshold for detection of an upper limit error in the A/D converted value. The upper limit error (ADCAnULER.UE) is set when the following condition is met: ULMTB[11:0] < A/D converted value
19 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 4	LLMTB[11:0]	Lower Limit Table Specify the threshold for detection of a lower limit error in the A/D converted value. The lower limit error (ADCAnULER.LE) is set when the following condition is met: LLMTB[11:0] > A/D converted value
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

- When A/D conversion is executed in 10-bit mode (ADCAnADCR.CTYP = 1), ULMTB[1:0] and LLMTB[1:0] should be set to 11B and 00B, respectively.
- To prevent malfunction, ADCAnULLMTBR0 to ADCAnULLMTBR7 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- The upper-limit table (ULMTB[11:0]) must be greater than the lower-limit table (LLMTB[11:0]).

38.3.2.23 ADCAnECR — Error Clear Register

This register is used to control clearing of an error. The read value is always 0.

Access: ADCAnECR register is a write-only register that can be written in 32-bit units.
ADCAnECRL register is a write-only register that can be written in 16-bit units.
ADCAnECRLL register is a write-only register that can be written in 8-bit units.

Address: ADCAnECR: <ADCAn_base> + 344_H
ADCAnECRL: <ADCAn_base> + 344_H
ADCAnECRLL: <ADCAn_base> + 344_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ULEC	OWEC	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	R	R

Table 38.61 ADCAnECR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	ULEC	Upper Limit Error Flag (ADCAnULER.UE) / Lower Limit Error Flag (ADCAnULER.LE) Clear, Upper/Lower Limit Error Capture (ADCAnULER.ULECAP[5:0]), Scan Group Bit (ULSG[1:0]) When Upper/Lower Limit Error Occurs, MPX Usage Bit (MPXE), and the MPX Value Storing Bit (MPXV[2:0]) Clear When Upper/Lower Limit Error Occurs Upper Limit/Lower Limit Error VC Flag (ADCAnULEVCFR0-1.ULEVC[49:0], ADCAnULEVCFR1.ULEPWDVC) Clear 0: No effect. 1: Clears the flag.
2	OWEC	Overwrite Error Flag (ADCAnOWER.OWE) and Overwrite Error Capture (ADCAnOWER.OWECAP[5:0]) Clear 0: No effect. 1: Clears the flag.
1, 0	Reserved	When writing, write the value after reset.

38.3.2.24 ADCAnULER — Upper Limit/Lower Limit Error Register

This register is a read-only register that indicates information regarding the upper limit/lower limit errors.

Access:	ADCAnULER register is a read-only register that can be read in 32-bit units. ADCAnULERL register is a read-only register that can be read in 16-bit units. ADCAnULERLH register is a read-only register that can be read in 8-bit units. ADCAnULERLL register is a read-only register that can be read in 8-bit units.
Address:	ADCAnULER: <ADCAn_base> + 348_H ADCAnULERL: <ADCAn_base> + 348_H ADCAnULERLL: <ADCAn_base> + 348_H ADCAnULERLH: <ADCAn_base> + 348_H + 1_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UE	LE	ULSG[1:0]		MPXE	MPXV[2:0]			—	—	ULECAP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.62 ADCAnULER Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	UE	Upper Limit Error Flag 0: An upper limit error is not detected. 1: An upper limit error is detected. Setting condition: The A/D converted value exceeds the upper limit threshold specified by the upper limit/lower limit table registers 0 to 7 (ADCAnULLMTBR0 to 7). If a subsequent upper limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated. Clearing condition: 1 is written to ADCAnECR.ULEC.
14	LE	Lower Limit Error Flag 0: A lower limit error is not detected. 1: A lower limit error is detected. Setting condition: The A/D converted value is lower than the lower limit threshold specified by the upper limit/lower limit table registers 0 to 7 (ADCAnULLMTBR0 to 7). If a subsequent lower limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated. Clearing condition: 1 is written to ADCAnECR.ULEC.

Table 38.62 ADCAnULER Register Contents

Bit Position	Bit Name	Function
13, 12	ULSG[1:0]	Scan Group where an Upper Limit/Lower Limit Error Occurs 00: No upper limit/lower limit error occurred. 01: The scan group where an upper limit/lower limit error occurred is SG1 to SG3. 10: The scan group where an upper limit/lower limit error occurred is PWM-Diag. Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: When 1 is written to ADCAnECR.ULEC.
11	MPXE*[1]	MPX Usage 0: The MPX function was not used when an upper limit/lower limit error occurred. 1: The MPX function was used when an upper limit/lower limit error occurred. Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0 Clearing condition: When 1 is written to ADCAnECR.ULEC.
10 to 8	MPXV[2:0]*[1]	The value of MPX is stored when the errors of the upper and lower limit occurred Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: When 1 is written to ADCAnECR.ULEC.
7	Reserved	When read, an undefined value is read.
6	Reserved	When read, the value after reset is returned.
5 to 0	ULECAP[5:0]	Upper Limit/Lower Limit Error Capture The physical channel is captured when an upper limit/lower limit error occurred. Capturing condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: 1 is written to ADCAnECR.ULEC.

Note 1. These bits are only supported for ADCA0.
For ADCA1, when read, the value after reset is returned.

NOTE

ADCAnULER is updated when the A/D converted value is set in ADCAnDRj or ADCAnPWDTSNDR.

38.3.2.25 ADCAnOWER — Overwrite Error Register

This register is a 32/16/8-bit read-only register that indicates an overwrite error. The target for overwrite errors is SG1 to SG3, and not PWM-Diag.

Access: ADCAnOWER register is a read-only register that can be read in 32-bit units.
ADCAnOWERL register is a read-only register that can be read in 16-bit units.
ADCAnOWERLL register is a read-only register that can be read in 8-bit units.

Address: ADCAnOWER: <ADCAn_base> + 34C$_H$
ADCAnOWERL: <ADCAn_base> + 34C$_H$
ADCAnOWERLL: <ADCAn_base> + 34C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OWE	—	OWECAP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.63 ADCAnOWER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	OWE	Overwrite Error Flag 0: An overwrite error is not detected. 1: An overwrite error is detected. Setting condition: ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj. If a subsequent overwrite error is detected in A/D conversion while this bit is set to 1, the ADCAnOWER register is not updated. Clearing condition: 1 is written to ADCAnECR.OWEC.
6	Reserved	When read, the value after reset is returned.
5 to 0	OWECAP[5:0]	Overwrite Error Capture The virtual channel number is captured when an overwrite error occurs. Capturing condition: OWE = 0 and ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj Clearing condition: 1 is written to ADCAnECR.OWEC.

NOTE

ADCAnOWER is updated when the A/D converted value is set in ADCAnDRj.

38.3.2.26 ADCAnULEVCFR0 — Upper Limit/Lower Limit Error VC Flag Register 0

This is a flag register for each virtual channel indicating an upper or lower limit error. ADCAnULEVCFR0 corresponds to ADCAnVCRj (j = 0 to 31).

Access: ADCAnULEVCFR0 register is a read-only register that can be read in 32-bit units.

Address: ADCAnULEVCFR0: <ADCAn_base> + 540_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULEVC[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ULEVC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.64 ADCAnULEVCFR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ULEVC[31:0]	ADCAnVCRj (j = 0 to 31) Upper Limit/Lower Limit Error Flag 0: An Upper Limit/lower Limit error is not detected. 1: An Upper Limit/lower Limit error is detected. Setting condition: The A/D converted value exceeds or lowers the upper/lower limit threshold specified by the upper limit/lower limit table registers 0 to 7 (ADCAnULLMTBR0 to ADCAnULLMTBR7). Clearing condition: 1 is written to ADCAnECR.ULEC.

38.3.2.27 ADCAnULEVCFR1 — Upper Limit/Lower Limit Error VC Flag Register 1

This is a flag register for each virtual channel indicating an upper or lower limit error. ADCAnULEVCFR1 corresponds to ADCAnVCRj (j = 32 to 49) and ADCAnPWDVCR.

Access: ADCAnULEVCFR1 register is a read-only register that can be read in 32-bit units.

Address: ADCAnULEVCFR1: <ADCAn_base> + 544_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULEPWDVC	—	—	—	—	—	—	—	—	—	—	—	—	—	ULEVC[49:48]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ULEVC[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.65 ADCAnULEVCFR1 Register Contents

Bit Position	Bit Name	Function
31	ULEPWDVC	ADCAnPWDVCR Upper Limit/Lower Limit Error Flag 0: An Upper Limit/lower Limit error is not detected. 1: An Upper Limit/lower Limit error is detected. Setting condition: The A/D converted value exceeds or lowers the upper/lower limit threshold specified by the upper limit/lower limit table registers 0 to 7 (ADCAnULLMTBR0 to ADCAnULLMTBR7). Clearing condition: 1 is written to ADCAnECR.ULEC.
30 to 18	Reserved	When read, the value after reset is returned.
17 to 0	ULEVC[49:32]	ADCAnVCRj (j=32-49) Upper Limit/Lower Limit Error Flag 0: An Upper Limit/lower Limit error is not detected. 1: An Upper Limit/lower Limit error is detected. Setting condition: The A/D converted value exceeds or lowers the upper/lower limit threshold specified by the upper limit/lower limit table registers 0 to 7 (ADCAnULLMTBR0 to ADCAnULLMTBR7). Clearing condition: 1 is written to ADCAnECR.ULEC.

38.3.3 Scan Group (SG) Specific Registers

This section describes the registers provided for each scan group.

38.3.3.1 ADCAnSGSTCRx — Scan Group x Start Control Register

This register is used to control the start of scan group x. The read value is always 0.

Access:	ADCAnSGSTCRx is a write-only register that can be written in 32-bit units. ADCAnSGSTCRxL is a write-only register that can be written in 16-bit units. ADCAnSGSTCRxLL is a write-only register that can be written in 8-bit units.
Address:	ADCAnSGSTCRx: <ADCAn_base> + x × 40_H + 400_H ADCAnSGSTCRxL: <ADCAn_base> + x × 40_H + 400_H ADCAnSGSTCRxLL: <ADCAn_base> + x × 40_H + 400_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.66 ADCAnSGSTCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SGST	Scan Group Start Trigger Writing 1 to SGST while ADCAnSGSTR.SGACT[3:1] = 0 starts the target SGx.

38.3.3.2 ADCAnPWDSGSTPCR — PWM-Diag Scan Stop Control Register

This is a control register to stop scan group for PWM-Diag. The read value is always 0.

Access:	ADCAnPWDSGSTPCR register is a write-only register that can be written in 32-bit units. ADCAnPWDSGSTPCRL register is a write-only register that can be written in 16-bit units. ADCAnPWDSGSTPCRLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnPWDSGSTPCR: <ADCAn_base> + 504_H ADCAnPWDSGSTPCRL: <ADCAn_base> + 504_H ADCAnPWDSGSTPCRLL: <ADCAn_base> + 504_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDS GSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.67 ADCAnPWDSGSTPCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	PWDSGSTP	Scan Group for PWM-Diag Stop Trigger Writing 1 to PWDSGSTP while ADCAnSGSTR.SGACT[4] = 1.

38.3.3.3 ADCAnPWDSGCR — PWM-Diag Scan Group Control Register

This register is used to control PWM-Diag.

Access: ADCAnPWDSGCR register can be read or written in 32-bit units.
ADCAnPWDSGCRL register can be read or written in 16-bit units.
ADCAnPWDSGCRLL register can be read or written in 8-bit units.

Address: ADCAnPWDSGCR: <ADCAn_base> + 508$_H$
ADCAnPWDSGCRL: <ADCAn_base> + 508$_H$
ADCAnPWDSGCRLL: <ADCAn_base> + 508$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDTRGMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.68 ADCAnPWDSGCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWDTRGMD	PWM-Diag Trigger Mode Select 0: PWSA_ADTRG trigger input is disabled. 1: PWSA_ADTRG is selected for the trigger input to the PWM-Diag scan group.

CAUTION

To prevent malfunction, ADCAnPWDSGCR should be set (except clearing PWDTRGMD upon completion of A/D conversion) when SGACT of the PWM-Diag scan group (SG4) is 0 (before the scan group is started).

38.3.3.4 ADCAnSGCRx — Scan Group x Control Register

This register controls scan group x.

Access:	ADCAnSGCRx register can be read or written in 32-bit units. ADCAnSGCRxL register can be read or written in 16-bit units. ADCAnSGCRxLL register can be read or written in 8-bit units.
Address:	ADCAnSGCRx: <ADCAn_base> + x × 40_H + 408_H ADCAnSGCRxL: <ADCAn_base> + x × 40_H + 408_H ADCAnSGCRxLL: <ADCAn_base> + x × 40_H + 408_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SCANMD	ADIE	SCT[1:0]		—	TRGMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W

Table 38.69 ADCAnSGCRx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode Write 0 to this bit for SG2 and SG3.
4	ADIE	Scan End Interrupt Enable 0: INT_SGx is not output when the scan for SGx ends. 1: INT_SGx is output when the scan for SGx ends.
3, 2	SCT[1:0]	Channel Repeat Times Select 00: The selected number of channel repeat times is one. 01: The selected number of channel repeat times is two. 10: The selected number of channel repeat times is four. 11: Setting prohibited
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD	Trigger Mode 0: Trigger input to SGx_TRG is disabled (Hardware trigger disabled). 1: SGx_TRG start trigger or hold complete trigger A/B is selected for the trigger input to SGx. The software trigger is valid regardless of the TRGMD bit setting.

CAUTION

To prevent malfunction, ADCAnSGCRx should be set (except clearing TRGMD upon completion of A/D conversion and ADIE = 0 setting when stop procedure of scan group) when SGACT of all scan groups is 0 (before the scan group is started) and TRGMD of all scan groups is 0.

38.3.3.5 ADCAnSGVCSPx — Scan Group x Start Virtual Channel Pointer

This register specifies the start pointer of a virtual channel.

Access: ADCAnSGVCSPx register can be read or written in 32-bit units.
ADCAnSGVCSPxL register can be read or written in 16-bit units.
ADCAnSGVCSPxLL register can be read or written in 8-bit units.

Address: ADCAnSGVCSPx: <ADCAn_base> + x × 40_H + $40C_H$
ADCAnSGVCSPxL: <ADCAn_base> + x × 40_H + $40C_H$
ADCAnSGVCSPxLL: <ADCAn_base> + x × 40_H + $40C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCSP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.70 ADCAnSGVCSPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0]	Start Virtual Channel Pointer These bits are used to specify the virtual channel from which the SGx scan is to be started.

CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When writing to the channel pointers, be sure to write in the following order: ADCAnSGVCSPx → ADCAnSGVCEPx. When SGx is started, the A/D conversion for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
- Though ADCAnSGVCSPx can be written during the A/D conversion, the register is updated at the time when ADCAnSGVCEPx is written. The new setting is applied when SGx is started next time.
- When the hardware trigger is used, writing to this register during operation is prohibited.

38.3.3.6 ADCAnSGVCEPx — Scan Group x End Virtual Channel Pointer

This register specifies the end pointer of a virtual channel.

Access:	ADCAnSGVCEPx register can be read or written in 32-bit units. ADCAnSGVCEPxL register can be read or written in 16-bit units. ADCAnSGVCEPxLL register can be read or written in 8-bit units.
Address:	ADCAnSGVCEPx: <ADCAn_base> + x × 40_H + 410_H ADCAnSGVCEPxL: <ADCAn_base> + x × 40_H + 410_H ADCAnSGVCEPxLL: <ADCAn_base> + x × 40_H + 410_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCEP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.71 ADCAnSGVCEPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCEP[5:0]	End Virtual Channel Pointer These bits are used to specify the virtual channel at which the SGx scan is to be ended.

CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When SGx is started, processing for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
  ADCAnSGVCEPx can be rewritten even when SGx is being processed. The new setting is applied when SGx is started next time.

38.3.3.7 ADCAnSGMCYCRx — Scan Group x Multicycle Register

This register is a 32/16/8-bit read/write register that indicates the number of scan times in multicycle scan mode.

Access:	ADCAnSGMCYCRx register can be read or written in 32-bit units. ADCAnSGMCYCRxL register can be read or written in 16-bit units. ADCAnSGMCYCRxLL register can be read or written in 8-bit units.
Address:	ADCAnSGMCYCRx: <ADCAn_base> + x × 40_H + 414_H ADCAnSGMCYCRxL: <ADCAn_base> + x × 40_H + 414_H ADCAnSGMCYCRxLL: <ADCAn_base> + x × 40_H + 414_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCYC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.72 ADCAnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MCYC[1:0]	Multicycle Number Specification These bits are used to specify the number of scan times in multicycle scan mode. 00_B: Number of scans = 1 01_B: Number of scans = 2 10_B: Setting prohibited 11_B: Number of scans = 4

CAUTION

- To prevent malfunction, ADCAnSGMCYCRx should be set when SGACT of scan group x is 0 (before the scan group is started) and TRGMD is 0.
- When SGx is started, the scan for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is repeatedly executed as many times as specified in ADCAnSGMCYCRx.

38.3.3.8 ADCAnPWDSGSEFCR — PWM-Diag Scan End Flag Clear Register

This register is used to control the clearing of PWM-Diag scan end flag (SEF). The bits are always read as 0.

Access:	ADCAnPWDSGSEFCR register is a write-only register that can be written in 32-bit units. ADCAnPWDSGSEFCRL register is a write-only register that can be written in 16-bit units. ADCAnPWDSGSEFCRLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnPWDSGSEFCR: <ADCAn_base> + 518_H ADCAnPWDSGSEFCRL: <ADCAn_base> + 518_H ADCAnPWDSGSEFCRLL: <ADCAn_base> + 518_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDSEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.73 ADCAnPWDSGSEFCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	PWDSEFC	PWM-Diag Scan End Flag Clear Trigger 0: No effect. 1: Clears the PWM-Diag scan end flag (ADCAnSGSTR.SEF[4]).

38.3.3.9 ADCAnSGSEFCRx — Scan Group x Scan End Flag Clear Register

This register is a write-only register that clears the scan end flag (ADCAnSGSTR.SEFx). The read value is always 0.

Access:	ADCAnSGSEFCRx register is a write-only register that can be written in 32-bit units. ADCAnSGSEFCRxL register is a write-only register that can be written in 16-bit units. ADCAnSGSEFCRxLL register is a write-only register that can be written in 8-bit units.
Address:	ADCAnSGSEFCRx: <ADCAn_base> + x × 40_H + 418_H ADCAnSGSEFCRxL: <ADCAn_base> + x × 40_H + 418_H ADCAnSGSEFCRxLL: <ADCAn_base> + x × 40_H + 418_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.74 ADCAnSGSEFCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SEFC	Scan End Flag Clear Trigger 0: No effect. 1: Clears the target SG scan end flag (ADCAnSGSTR.SEFx).

38.3.3.10 ADCAnSGSTR — Scan Group Status Register

This register indicates the state of T&H, SVSTOP, scan group x, and PWM-Diag scan group. The SHACT and SGACT bits are cleared when HALT is executed.

Access:	ADCAnSGSTR register is a read-only register that can be read in 32-bit units. ADCAnSGSTRL register is a read-only register that can be read in 16-bit units.
Address:	ADCAnSGSTR: <ADCAn_base> + 308_H ADCAnSGSTRL: <ADCAn_base> + 308_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SHACT	SGACT[5:1]					—	—	—	—	SEF[4:1]				—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.75 ADCAnSGSTR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	SHACT	T&H Status Flag 0: T&H is stopped. 1: T&H conversion or sampling is in progress.
13	SGACT[5]	SVSTOP Status Flag 0: SVSTOP is canceled. 1: SVSTOP is accepted.
12	SGACT[4]	PWM-Diag Scan Group (SG4) Status Flag 0: A/D conversion for PWM-Diag (SG4) is completed. 1: A/D conversion for PWM-Diag (SG4) is in processing or suspension.
11	SGACT[3]	Scan Group 3 (SG3) Status Flag 0: A/D conversion for SG3 is completed. 1: A/D conversion for SG3 is in processing or suspension.
10	SGACT[2]	Scan Group 2 (SG2) Status Flag 0: A/D conversion for SG2 is completed. 1: A/D conversion for SG2 is in processing or suspension.
9	SGACT[1]	Scan Group 1 (SG1) Status Flag 0: A/D conversion for SG1 is completed. 1: A/D conversion for SG1 is in processing or suspension.
8 to 5	Reserved	When read, the value after reset is returned.
4	SEF[4]	PWM-Diag Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: ● ADCAnPWDTSNDR for PWM-Diag is read. ● ADCAnPWDDIR for PWM-Diag is read. ● ADCAnPWDSGSEFCR.PWDSEFC is written as 1. 1: The A/D conversion result is written to ADCAnPWDTSNDR for PWM-Diag.

Table 38.75 ADCAnSGSTR Register Contents

Bit Position	Bit Name	Function
3	SEF[3]	SG3 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: ● ADCAnDRj for the virtual channel which ADCAnSGVCEP3 indicates is read. ● ADCAnDIRj for the virtual channel which ADCAnSGVCEP3 indicates is read. ● ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP3 indicates.
2	SEF[2]	SG2 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: ● ADCAnDRj for the virtual channel which ADCAnSGVCEP2 indicates is read. ● ADCAnDIRj for the virtual channel which ADCAnSGVCEP2 indicates is read. ● ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP2 indicates.
1	SEF[1]	SG1 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: ● ADCAnDRj for the virtual channel which ADCAnSGVCEP1 indicates is read. ● ADCAnDIRj for the virtual channel which ADCAnSGVCEP1 indicates is read. ● ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP1 indicates.
0	Reserved	When read, the value after reset is returned.

38.3.3.11 ADCAnSGSTPCRx — Scan Group x Stop Control Register

This is a control register to stop scan group x. The read value is always 0.

Access: ADCAnSGSTPCRx register is a write-only register that can be written in 32-bit units.
ADCAnSGSTPCRxL register is a write-only register that can be written in 16-bit units.
ADCAnSGSTPCRxLL register is a write-only register that can be written in 8-bit units.

Address: ADCAnSGSTPCRx: <ADCAn_base> + x × 40_H + 404_H
ADCAnSGSTPCRxL: <ADCAn_base> + x × 40_H + 404_H
ADCAnSGSTPCRxLL: <ADCAn_base> + x × 40_H + 404_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 38.76 ADCAnSGSTPCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SGSTP	Scan Group x Stop Trigger Writing 1 to SGSTP while ADCAnSGSTR.SGACT[3:1] = 1 stops the target SGx.

38.3.4 Hardware Trigger Specific Register

38.3.4.1 ADCAnSGTSELx — Scan Group x Start Trigger Control Register

This register is used to select the A/D conversion trigger (hardware trigger) for SGx.

Access:	ADCAnSGTSELx register can be read or written in 32-bit units. ADCAnSGTSELxL register can be read or written in 16-bit units.
Address:	ADCAnSGTSELx: <ADCAn_base> + x × 40_H + $41C_H$ ADCAnSGTSELxL: <ADCAn_base> + x × 40_H + $41C_H$
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TxSEL9 *1	TxSEL8 *1	TxSEL7 *1	TxSEL6 *1	TxSEL5 *1	TxSEL4 *1	TxSEL3 *1	TxSEL2 *1	TxSEL1 *1	TxSEL0 *1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ADCA1 supports only TxSEL0 to TxSEL4. When writing to the other bits, write the value after reset.

Table 38.77 ADCAnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	TxSELp (p = 0 to 9)	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled. **CAUTION:** When setting TxSELp to 1, set only one of the bits to 1.

The list below shows the hardware triggers to be selected.

Table 38.78 List of A/D Conversion Hardware Triggers

	Control Register/Bit		Trigger Input Signal	
Unit	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCA0	ADCA0SGTSEL1	T1SEL0	ADCA0TRG0	External trigger pin
		T1SEL1	INTTAUJ0I3	TAUJ0
		T1SEL2	INTTAUD0I7	TAUD0
		T1SEL3	INTTAUD0I15	TAUD0
		T1SEL4	SEQADTRG	LPS
		T1SEL5	INTENCA0I1	ENCA0
		T1SEL6	TAPATADOUT0	Motor control (TAPA0)
		T1SEL7	TAPATADOUT1	Motor control (TAPA0)
		T1SEL8	ADOPA0ADCATTIN00	Motor control (PIC0)
		T1SEL9	INTTAUJ2I3	TAUJ2
	ADCA0SGTSEL2	T2SEL0	ADCA0TRG1	External trigger pin
		T2SEL1	INTTAUJ0I3	TAUJ0
		T2SEL2	INTTAUD0I7	TAUD0
		T2SEL3	INTTAUD0I15	TAUD0
		T2SEL4	SEQADTRG	LPS
		T2SEL5	INTENCA0I1	ENCA0
		T2SEL6	TAPATADOUT0	Motor control (TAPA0)
		T2SEL7	TAPATADOUT1	Motor control (TAPA0)
		T2SEL8	ADOPA1ADCATTIN00	Motor control (PIC0)
		T2SEL9	INTTAUJ2I3	TAUJ2
	ADCA0SGTSEL3	T3SEL0	ADCA0TRG2	External trigger pin
		T3SEL1	INTTAUJ0I3	TAUJ0
		T3SEL2	INTTAUD0I7	TAUD0
		T3SEL3	INTTAUD0I15	TAUD0
		T3SEL4	SEQADTRG	LPS
		T3SEL5	INTENCA0I1	ENCA0
		T3SEL6	TAPATADOUT0	Motor control (TAPA0)
		T3SEL7	TAPATADOUT1	Motor control (TAPA0)
		T3SEL8	ADOPA2ADCATTIN00	Motor control (PIC0)
		T3SEL9	INTTAUJ2I3	TAUJ2
ADCA1	ADCA1SGTSEL1	T1SEL0	ADCA1TRG0	External trigger pin
		T1SEL1	INTTAUJ1I3	TAUJ1
		T1SEL2	INTTAUB0I7	TAUB0
		T1SEL3	INTTAUB0I15	TAUB0
		T1SEL4	INTTAUJ3I3	TAUJ3
	ADCA1SGTSEL2	T2SEL0	ADCA1TRG1	External trigger pin
		T2SEL1	INTTAUJ1I3	TAUJ1
		T2SEL2	INTTAUB0I7	TAUB0
		T2SEL3	INTTAUB0I15	TAUB0
		T2SEL4	INTTAUJ3I3	TAUJ3
	ADCA1SGTSEL3	T3SEL0	ADCA1TRG2	External trigger pin
		T3SEL1	INTTAUJ1I3	TAUJ1
		T3SEL2	INTTAUB0I7	TAUB0
		T3SEL3	INTTAUB0I15	TAUB0
		T3SEL4	INTTAUJ3I3	TAUJ3

CAUTIONS

1. When enabling the LPS trigger factor (SEQADTRG), select and enable only one of ADCA0SGTSEL1.T1SEL4, ADCA0SGTSEL2.T2SEL4, and ADCA0SGTSEL3.T3SEL4.
2. To prevent malfunction, ADCAnSGTSELx should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

38.3.5 Self-Diagnosis Specific Registers

38.3.5.1 ADCAnDGCTL0 — Self-Diagnosis Control Register 0

This register controls the self-diagnostic voltage level.

Access:	ADCAnDGCTL0 register can be read or written in 32-bit units. ADCAnDGCTL0L register can be read or written in 16-bit units. ADCAnDGCTL0LL register can be read or written in 8-bit units.
Address:	ADCAnDGCTL0: <ADCAn_base> + 350_{H} ADCAnDGCTL0L: <ADCAn_base> + 350_{H} ADCAnDGCTL0LL: <ADCAn_base> + 350_{H}
Value after reset:	$0000\ 0000_{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PSEL2	PSEL1	PSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.79 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PSEL[2:0]	Self-Diagnostic Voltage Level Select

ADCAnDGCTL0			Output Signal			
PSEL2	PSEL1	PSEL0	ADDIAGOUT	DIAGOUT2	DIAGOUT1	DIAGOUT0
0	0	0	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	AnVSS	2/3AnV$_{REF}$	1/2AnV$_{REF}$	1/3AnV$_{REF}$
0	1	0	1/3AnV$_{REF}$	1/3AnV$_{REF}$	2/3AnV$_{REF}$	1/2AnV$_{REF}$
0	1	1	1/2AnV$_{REF}$	1/2AnV$_{REF}$	1/3AnV$_{REF}$	2/3AnV$_{REF}$
1	0	0	2/3AnV$_{REF}$	Hi-z	Hi-z	Hi-z
1	0	1	AnV$_{REF}$	1/3AnV$_{REF}$	1/3AnV$_{REF}$	1/3AnV$_{REF}$
1	1	0	AnV$_{REF}$	1/2AnV$_{REF}$	1/2AnV$_{REF}$	1/2AnV$_{REF}$
1	1	1	AnV$_{REF}$	2/3AnV$_{REF}$	2/3AnV$_{REF}$	2/3AnV$_{REF}$

NOTE

The value of the ADCAnDGCTL0.PSEL[2:0] bits are updated after the following condition occurs.

- ADCAnADCR.DGON = 1
- Sampling completion

38.3.5.2 ADCAnDGCTL1 — Self-Diagnosis Control Register 1

This register controls the self-diagnostic channel.

Access: ADCAnDGCTL1 register can be read or written in 32-bit units.
ADCAnDGCTL1L register can be read or written in 16-bit units.

Address: ADCAnDGCTL1: <ADCAn_base> + 354_H
ADCAnDGCTL1L: <ADCAn_base> + 354_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.80 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15, 12, 9, 6, 3, 0	CDG [15, 12, 9, 6, 3, 0]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT0 is selected.
13, 10, 7, 4, 1	CDG [13, 10, 7, 4, 1]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT1 is selected.
14, 11, 8, 5, 2	CDG [14, 11, 8, 5, 2]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT2 is selected.

CAUTION

To prevent malfunction, ADCAnDGCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

38.3.5.3 ADCAnPDCTL1 — Pull Down Control Register 1

This register specifies the channel to which the pull down resistor is connected.

For details, see **Section 38.5.3, Diagnosis of Open Pins**.

Access:	ADCAnPDCTL1 register can be read or written in 32-bit units. ADCAnPDCTL1L register can be read or written in 16-bit units.
Address:	ADCAnPDCTL1: <ADCAn_base> + 358_H ADCAnPDCTL1L: <ADCAn_base> + 358_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNA[15:0]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ADCAnPDCTL1.PDNA [15:0] corresponds to ANInI15 to ANInI0. For details, see **Table 38.28, ADCA0 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.29, ADCA0 Analog Input Signals (RH850/F1KM-S4)**, **Table 38.30, ADCA0 Analog Input Signals (RH850/F1KM-S1)**, **Table 38.32, ADCA1 Analog Input Signals (RH850/F1KH-D8)**, and **Table 38.33, ADCA1 Analog Input Signals (RH850/F1KM-S4)**. When writing to the bits of unused ANIn signals, write the value after reset.

Table 38.81 ADCAnPDCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDNA[15:0]	Pull Down Enable Control These bits set whether an on-chip pull-down resistor is to be connected to the corresponding physical channel (ANIn[00:15]). 0: An on-chip pull-down resistor is not connected. 1: An on-chip pull-down resistor is connected.

CAUTION

To prevent malfunction, ADCAnPDCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

For on-chip pull-down resistor values, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.

38.3.5.4 ADCAnPDCTL2 — Pull Down Control Register 2

This register specifies the channel to which the pull down resistor is connected.

For details, see **Section 38.5.3, Diagnosis of Open Pins**.

Access:	ADCAnPDCTL2 register can be read or written in 32-bit units. ADCAnPDCTL2H and ADCAnPDCTL2L registers can be read or written in 16-bit units. ADCAnPDCTL2HL, ADCAnPDCTL2LH, and ADCAnPDCTL2LL registers can be read or written in 8-bit units.
Address:	ADCAnPDCTL2: <ADCAn_base> + 35C$_H$ ADCAnPDCTL2L: <ADCAn_base> + 35C$_H$ ADCAnPDCTL2H: <ADCAn_base> + 35C$_H$ + 2$_H$ ADCAnPDCTL2HL: <ADCAn_base> + 35C$_H$ +2$_H$ ADCAnPDCTL2LL: <ADCAn_base> + 35C$_H$ ADCAnPDCTL2LH: <ADCAn_base> + 35C$_H$ + 1$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PDNB[19:16]*1			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNB[15:0]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ADCAnPDCTL2.PDNB [19:0] corresponds to ANInI19S to ANInI0S. For details, see **Table 38.28, ADCA0 Analog Input Signals (RH850/F1KH-D8)**, **Table 38.29, ADCA0 Analog Input Signals (RH850/F1KM-S4)**, **Table 38.30, ADCA0 Analog Input Signals (RH850/F1KM-S1)**, **Table 38.32, ADCA1 Analog Input Signals (RH850/F1KH-D8)**, and **Table 38.33, ADCA1 Analog Input Signals (RH850/F1KM-S4)**. When writing to the bits of unused ANIn signals, write the value after reset.

Table 38.82 ADCAnPDCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	PDNB[19:0]	Pull Down Enable Control These bits set whether the on-chip pull-down resistor is to be connected with the corresponding physical channel (ANIn[16:35]). 0: The on-chip pull-down resistor is not connected. 1: The on-chip pull-down resistor is connected.

CAUTION

To prevent malfunction, ADCAnPDCTL2 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

For on-chip pull-down resistor values, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.

38.3.6 Emulation Specific Register

38.3.6.1 ADCAnEMU — Emulation Control Register

This register controls the SVSTOP disable signal.

Access: This register can be read or written in 8-bit units.

Address: <ADCAn_base> + 388_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 38.83 ADCAnEMU Register Contents

Bit Position	Bit Name	Function
7	SVSDIS	SVSTOP Disable 0: SVSTOP is enabled 1: SVSTOP is disabled For the A/D conversion when SVSTOP is enabled, see **Section 38.4.10.3, SVSTOP Operation**.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

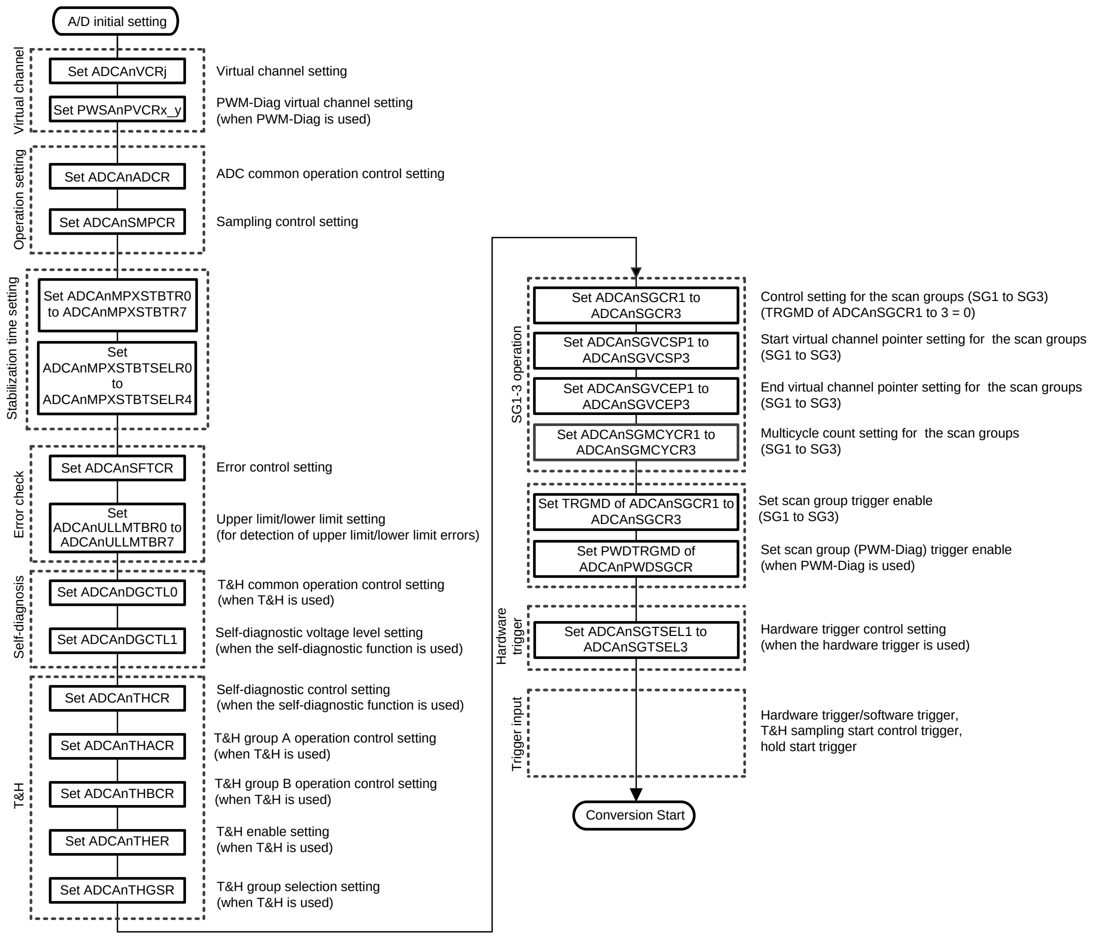
CAUTION

To prevent malfunction, SVSDIS should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

38.4 Operation

38.4.1 Initial Setting

Figure 38.6, Flowchart for Initial Setting shows an initial setting example of the A/D conversion. For trigger input, see **Figure 38.7, Flowchart for Trigger Input** in the next section. For interrupt request signals, see **Section 38.4.12, Scan End Interrupt Request**.



Note: Set these registers as necessary.

Figure 38.6 Flowchart for Initial Setting

38.4.2 Trigger Input

The following figure shows the flowchart for trigger input.

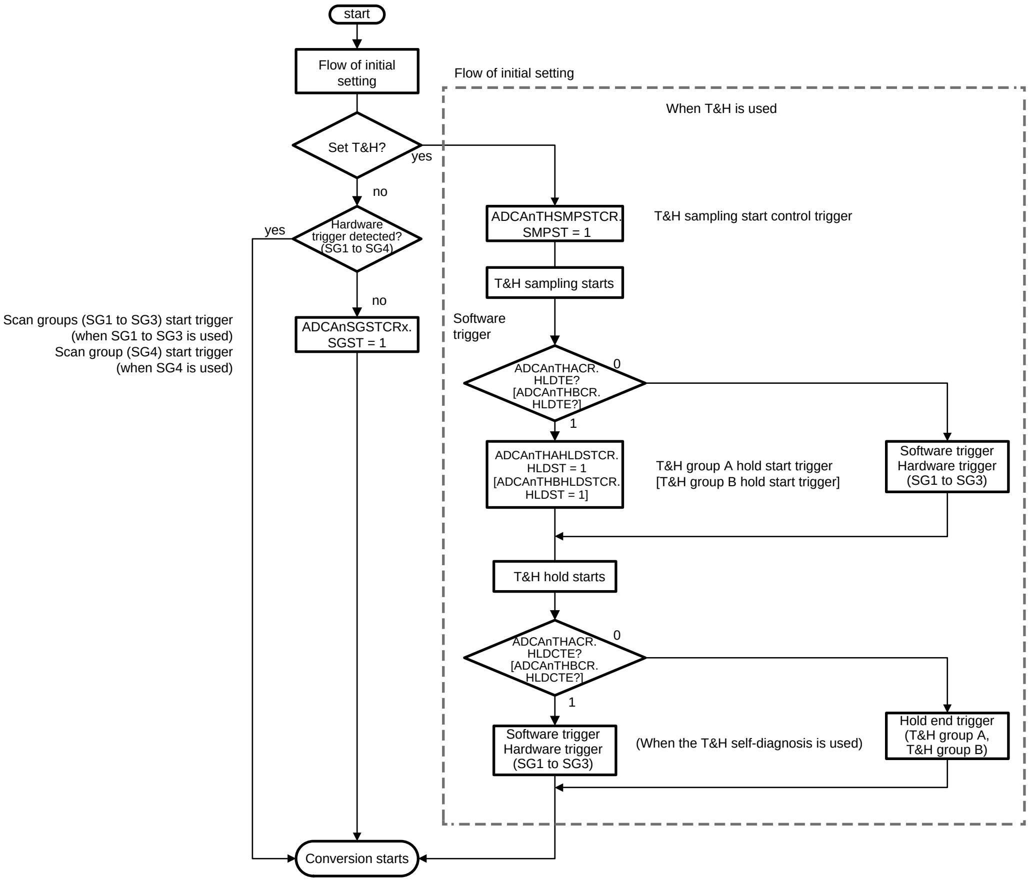


Figure 38.7 Flowchart for Trigger Input

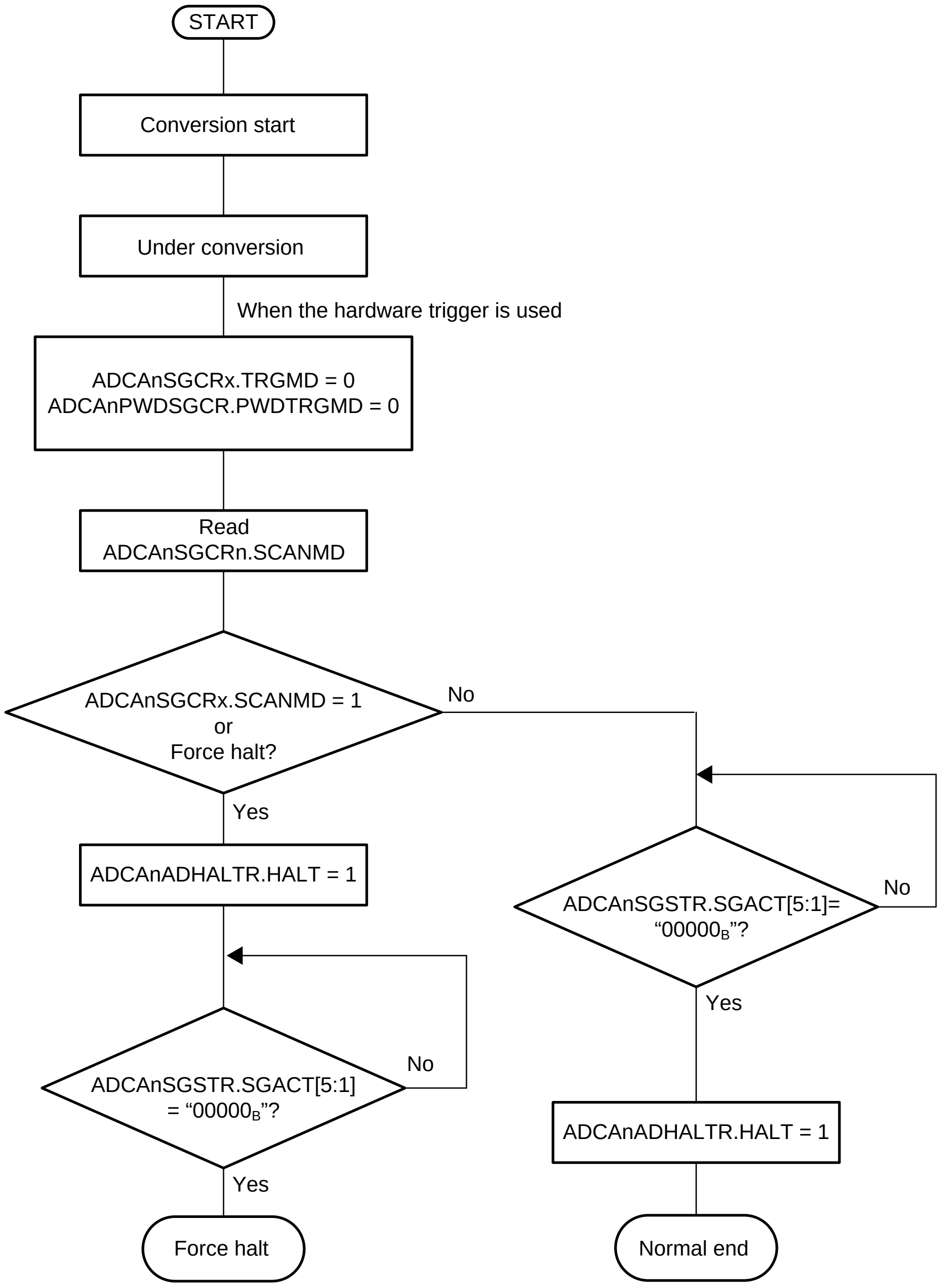
NOTE

When an SG start trigger is generated during scanning, the SG start trigger is ignored.

38.4.3 Ending/Stopping A/D Conversion

38.4.3.1 Ending by ADHALT

The flow for ending A/D conversion is shown below.



Note: The final setting of HALT when A/D conversion ends normally is done to clarify whether the status is "Conversion" or "IDLE". Whether HALT is set or not when A/D conversion ends normally does not affect the A/D conversion accuracy.

Figure 38.8 Flowchart for Ending A/D Conversion by ADHALT

38.4.3.2 Stopping by SGSTP (for Each Scan Group)

The flow for stopping A/D conversion for each scan group is shown below with SG1 as an example. Scan group which is selected in T&H group (Group A or Group B) has to be stopped after stopping of T&H (Group A or Group B).

Other scan group can be also controlled by its corresponding bit (ADCAnSGSTPCR2.SGSTP, ADCAnSGSTPCR3.SGSTP and ADCAnPWDSGSTPCR.PWDSGSTP). In case of SGSTP = 1 is set, it is necessary to confirm the clearing of corresponded scan group of SGACT.

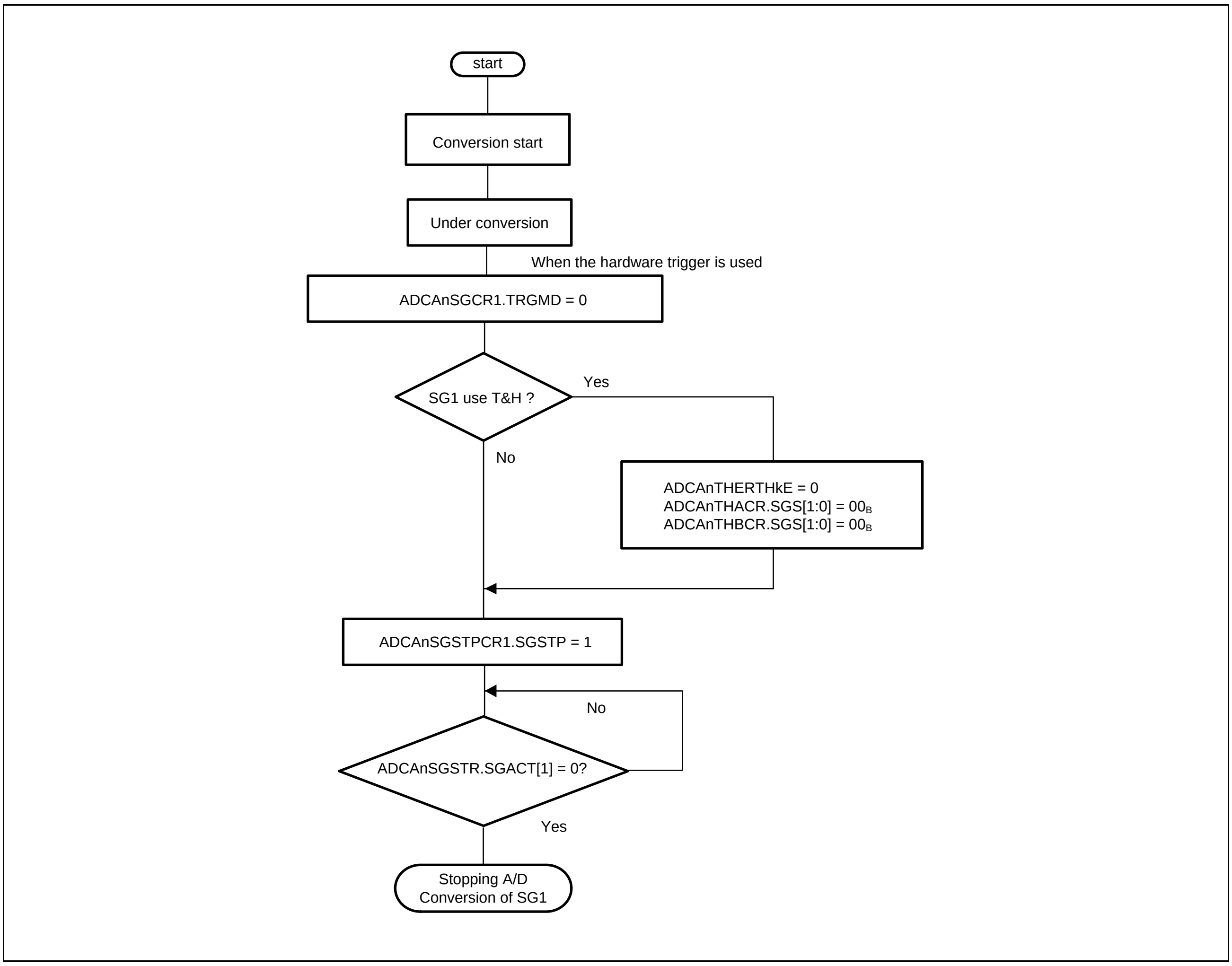


Figure 38.9 Flowchart for Stopping A/D Conversion for Each Scan Group (SG1)

CAUTION

In order to treat the A/D conversion results and re-set registers correctly after scan group stop setting, more than 14-bus clock (7-times of dummy read) has to be waited. In case of using the interrupt, this procedure is not necessary.

NOTE

When the scan group stop bit is set right before A/D conversion is completed, the corresponding scan group makes a transition to the IDLE state, but an A/D conversion end interrupt or scan end interrupt may be output. If it has any system inconvenience, disable the interrupt and then set the stop bit.

When the scan group stop bit (ADCAnSGSTPCRx.SGSTP, ADCAnPWDSGSTPCR.PWDSGST) is set, the corresponding scan group enters the IDLE state and SGACT is cleared to 0 after finishing of the virtual channel in progress. An example of the operation is shown below.

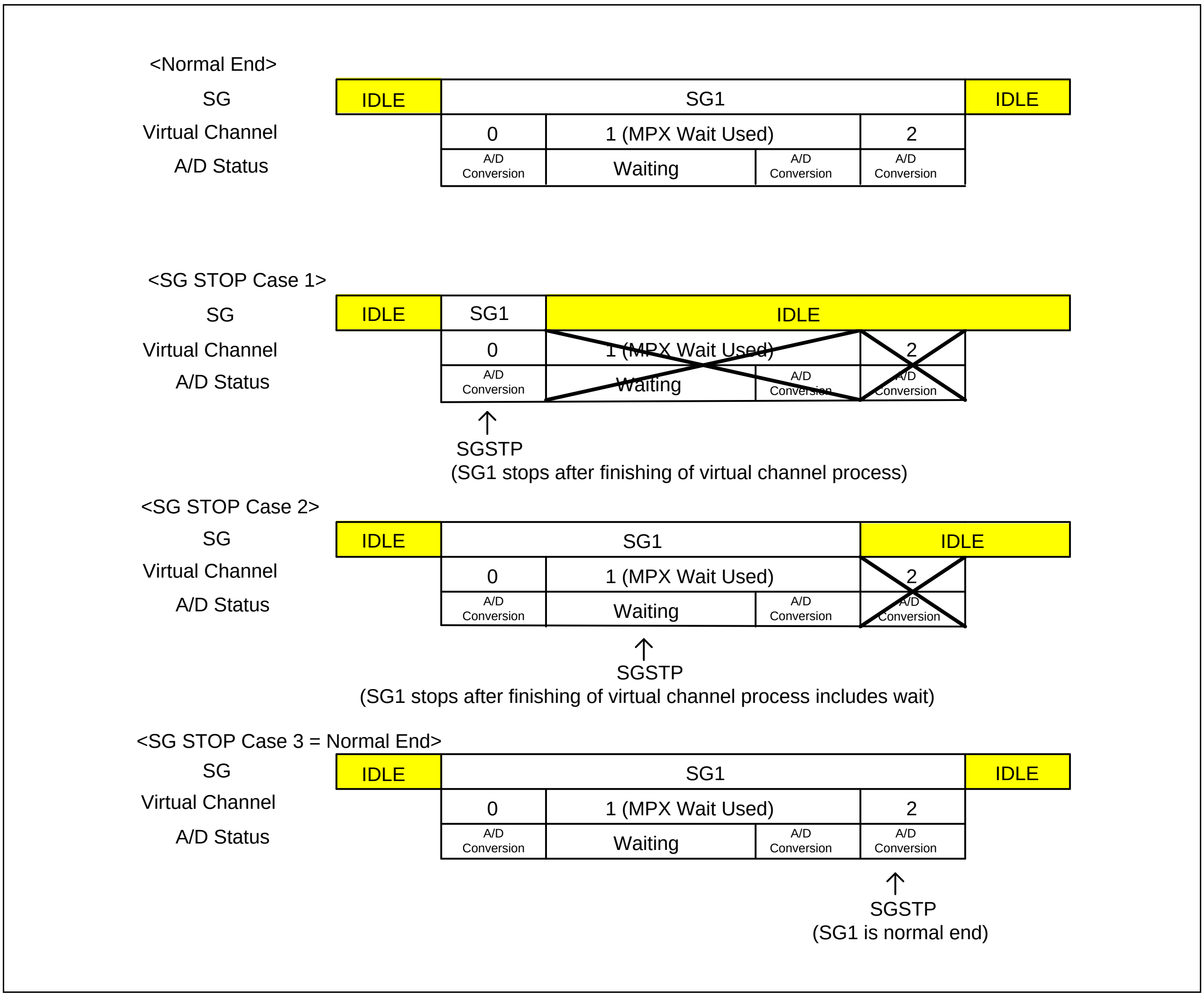


Figure 38.10 Example of Scan Group Stop Operation (SG1)

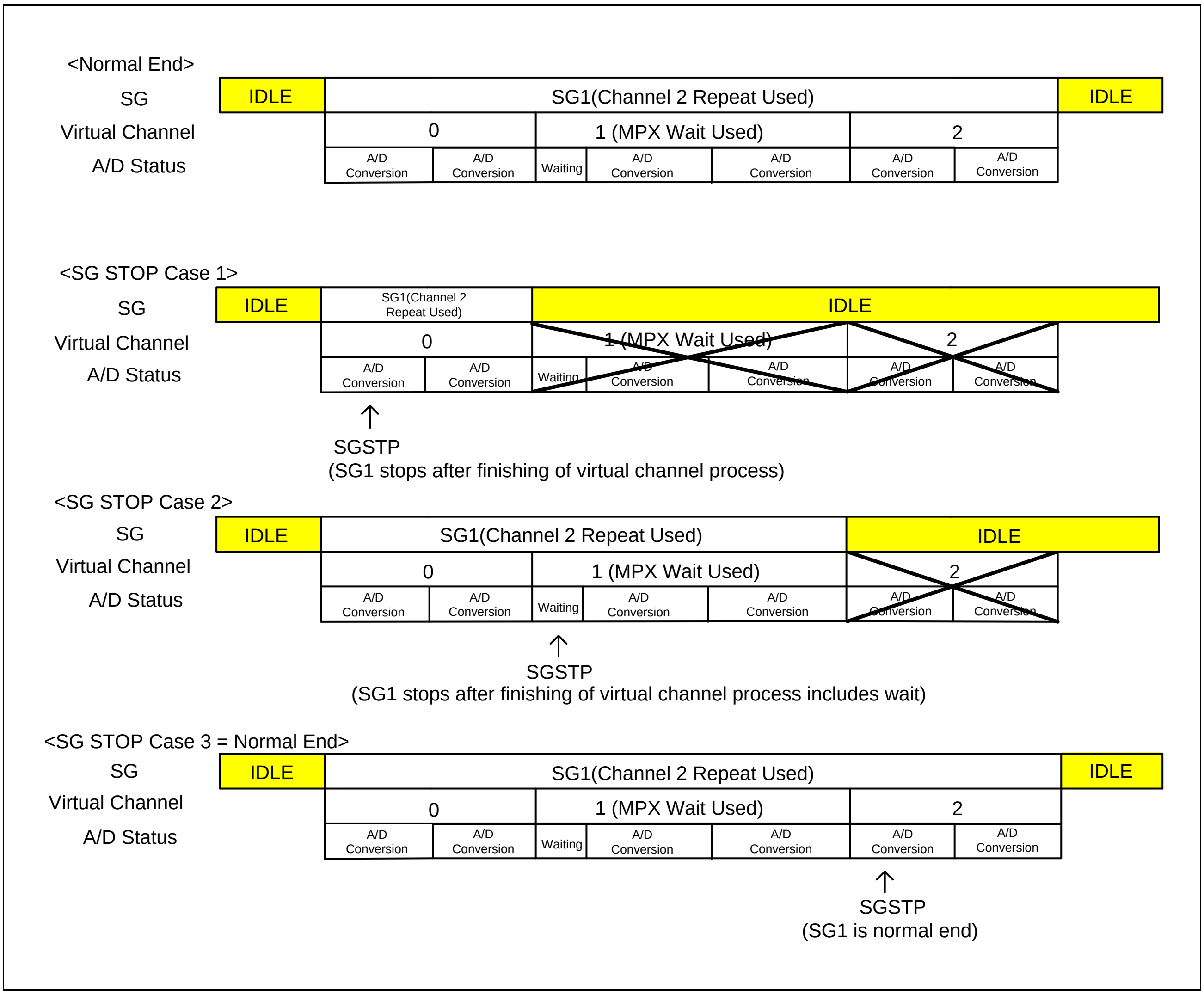


Figure 38.11 Example of Scan Group Stop Operation (SG1 channel repeat)

When the stop bit of the scan group that has been suspended due to synchronous/asynchronous suspend is set, the corresponding scan group enters the IDLE state and SGACT is cleared to 0 without resuming the processing.

When the scan group stop bit is set and asynchronous suspend occurs due to interruption by a scan group with a higher priority during processing by a virtual channel, the processing of the virtual channel is stopped as shown in **Figure 38.12, Example of Scan Group Stop Operation (Asynchronous suspend of SG1)**, the virtual channel enters the IDLE state and SGACT is cleared to 0.

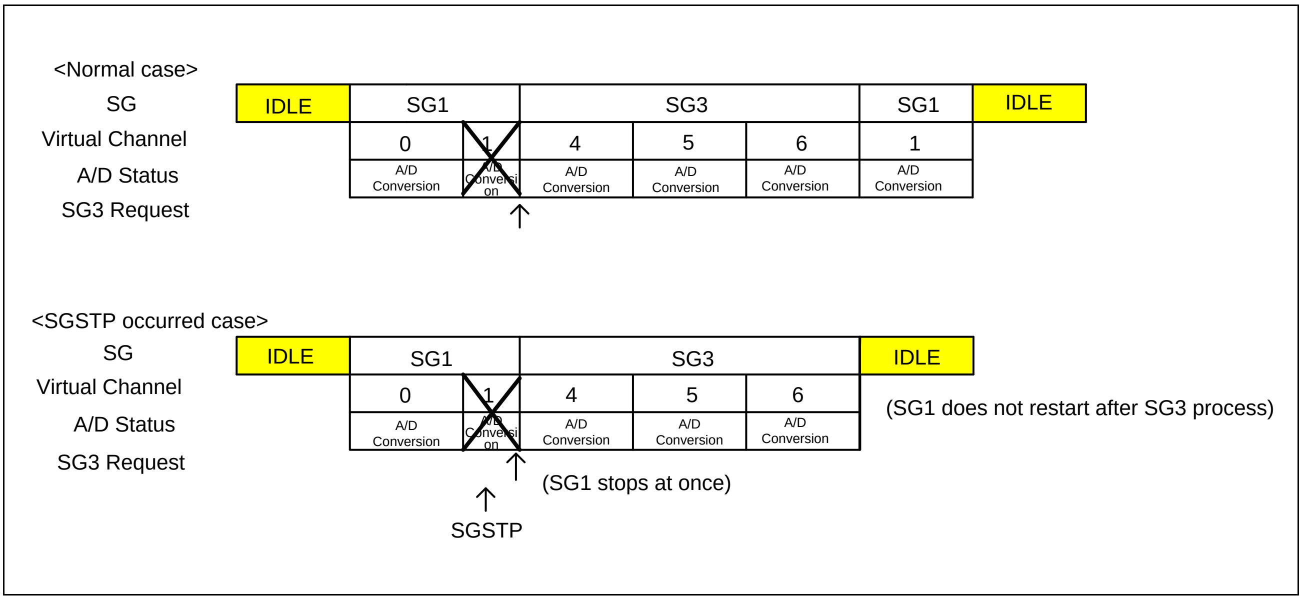
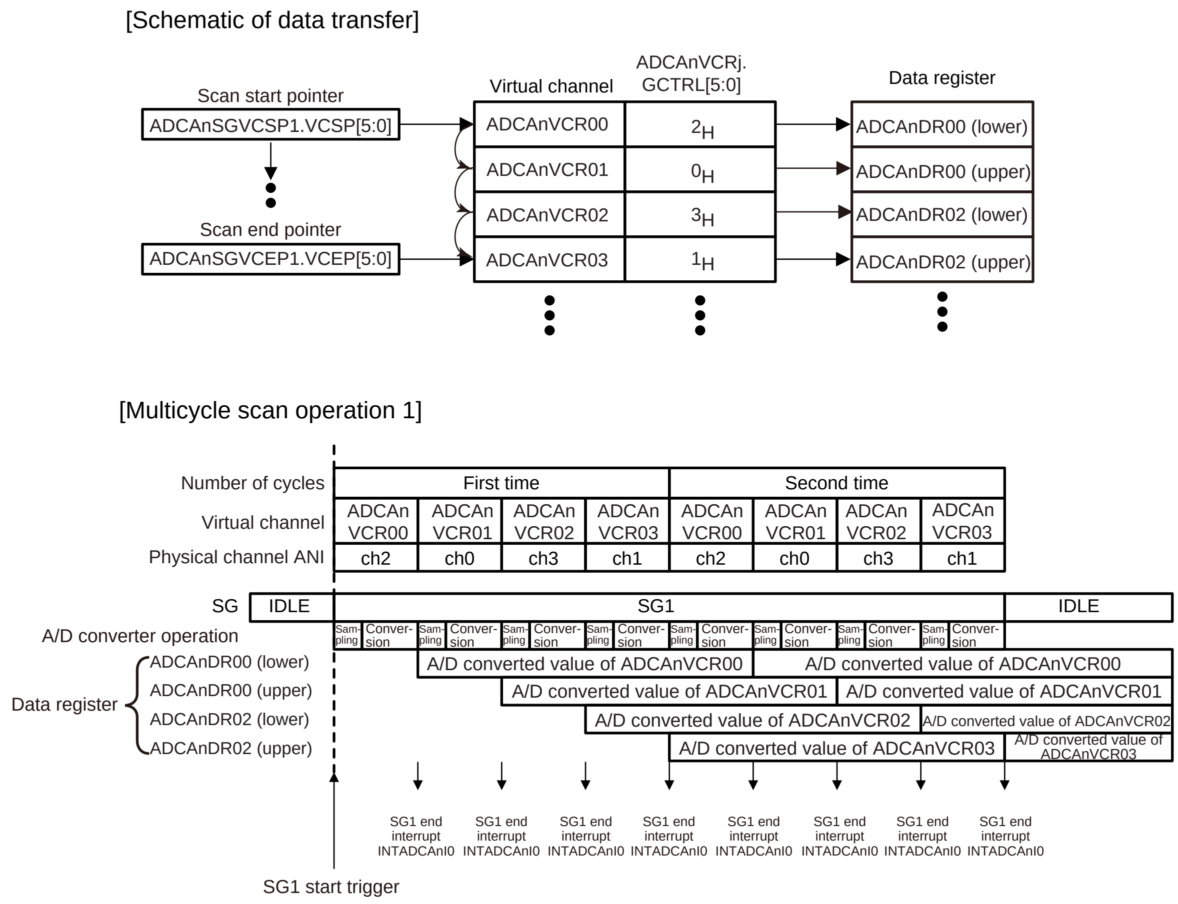


Figure 38.12 Example of Scan Group Stop Operation (Asynchronous suspend of SG1)

38.4.4 Example of Scan Group Operation

(1) Multicycle Scan Mode

The following figure illustrates an operation example where four virtual channels of scan group 1 are converted using the two-cycle scan in multicycle scan mode.

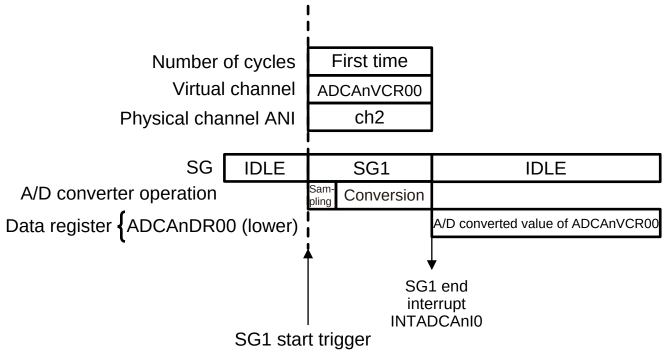


Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 using the two-cycle scan in multicycle scan mode

Figure 38.13 Example of Multicycle Scan Operation 1

The following figure illustrates an operation example where a pin is scanned once in multicycle scan mode.



Conditions

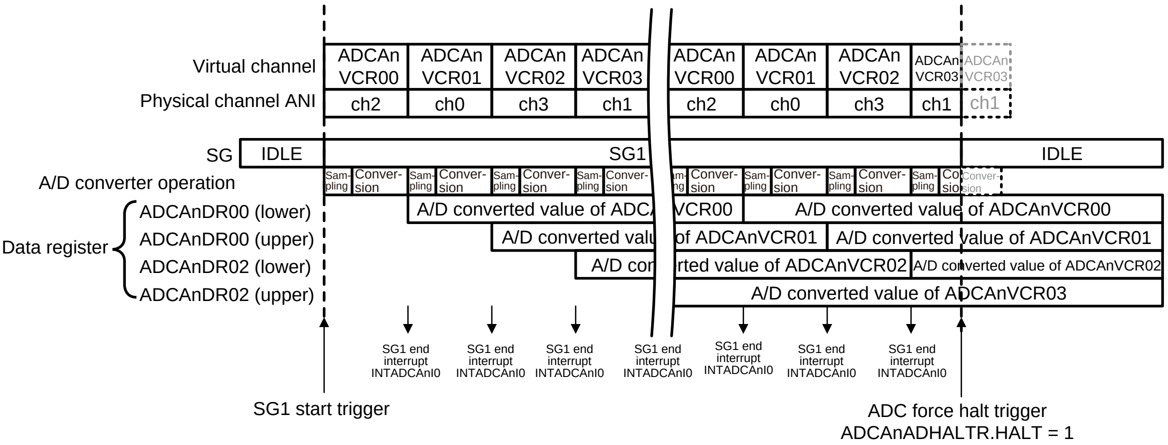
Scan group 1: A/D conversion of virtual channel 0 in which a pin is scanned once in multicycle scan mode

Figure 38.14 Example of Multicycle Scan Operation 2

(2) Continuous Scan Mode

Continuous scan mode allows A/D conversion of the SG channels indicated by the pointers specified by ADCAnSGVCSPx.VCSP[5:0] to ADCAnSGVCEPx.VCEP[5:0] to continue until ADCAnADHALTR.HALT or ADCAnSGSTPCRx.SGSTP is asserted. This mode can be used only with SG1.

The following figure shows an example of operation in continuous scan mode.



Conditions

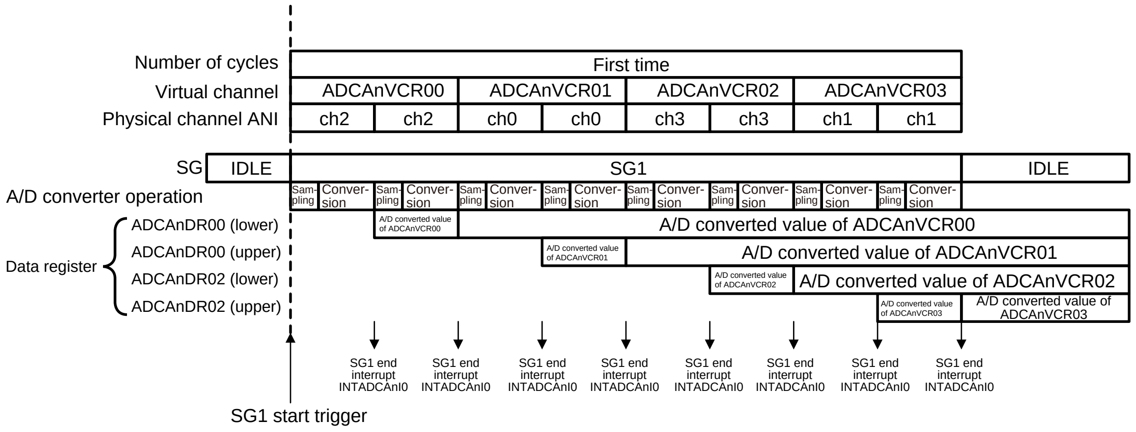
Scan group 1: A/D conversion of virtual channels 0 to 3 in continuous scan mode

Figure 38.15 Example of Continuous Scan Operation

38.4.5 Channel Repeat Mode

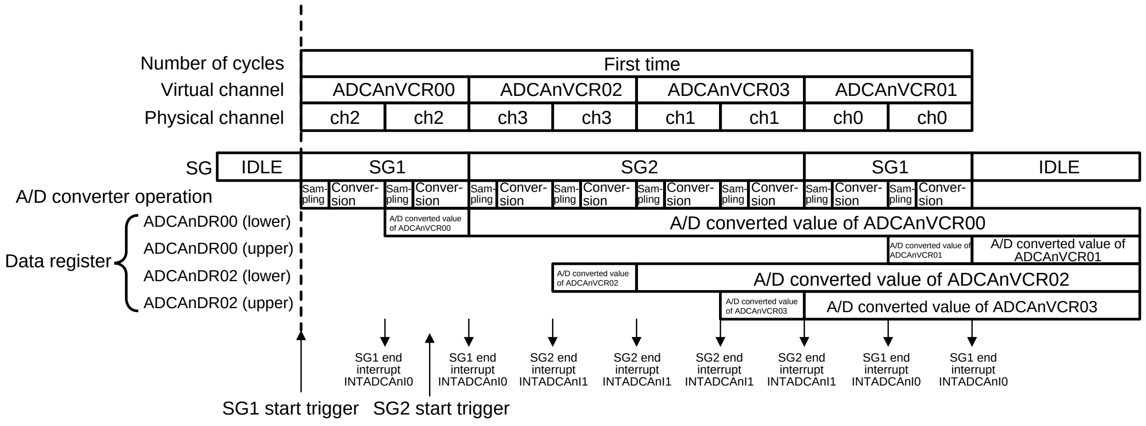
Channel repeat mode allows A/D conversion of the SG channel indicated by the pointer specified by ADCAnSGVCSPx.VCSP[5:0] to ADCAnSGVCEPx.VCEP[5:0] to repeat number of channel repeat times specified by ADCAnSGCRx.SCT[1:0]. This mode operates exclusively in each SG. The number of channel repeat times is selectable from 1, 2, and 4.

The following figures show examples of operation under respective conditions.



Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

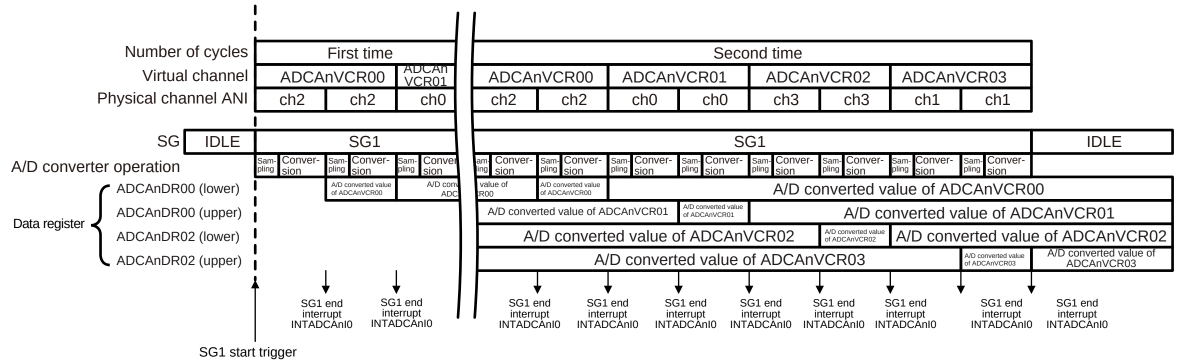


Conditions

Scan group 1: A/D conversion of virtual channels 0 and 1 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Scan group 2: A/D conversion of virtual channels 2 and 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

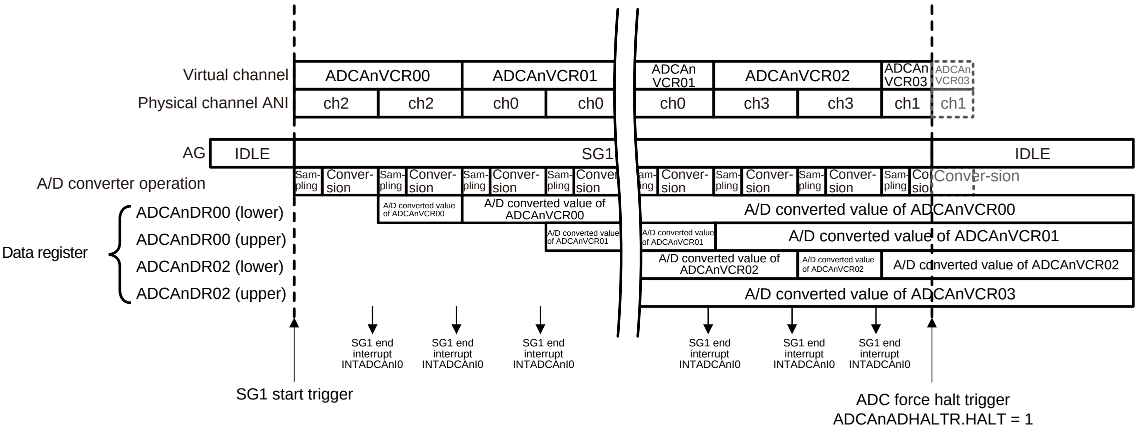
Figure 38.16 Example of Channel Repeat Operation 1



Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice and the two-cycle scan is used in multicycle scan mode

Figure 38.17 Example of Channel Repeat Operation 2



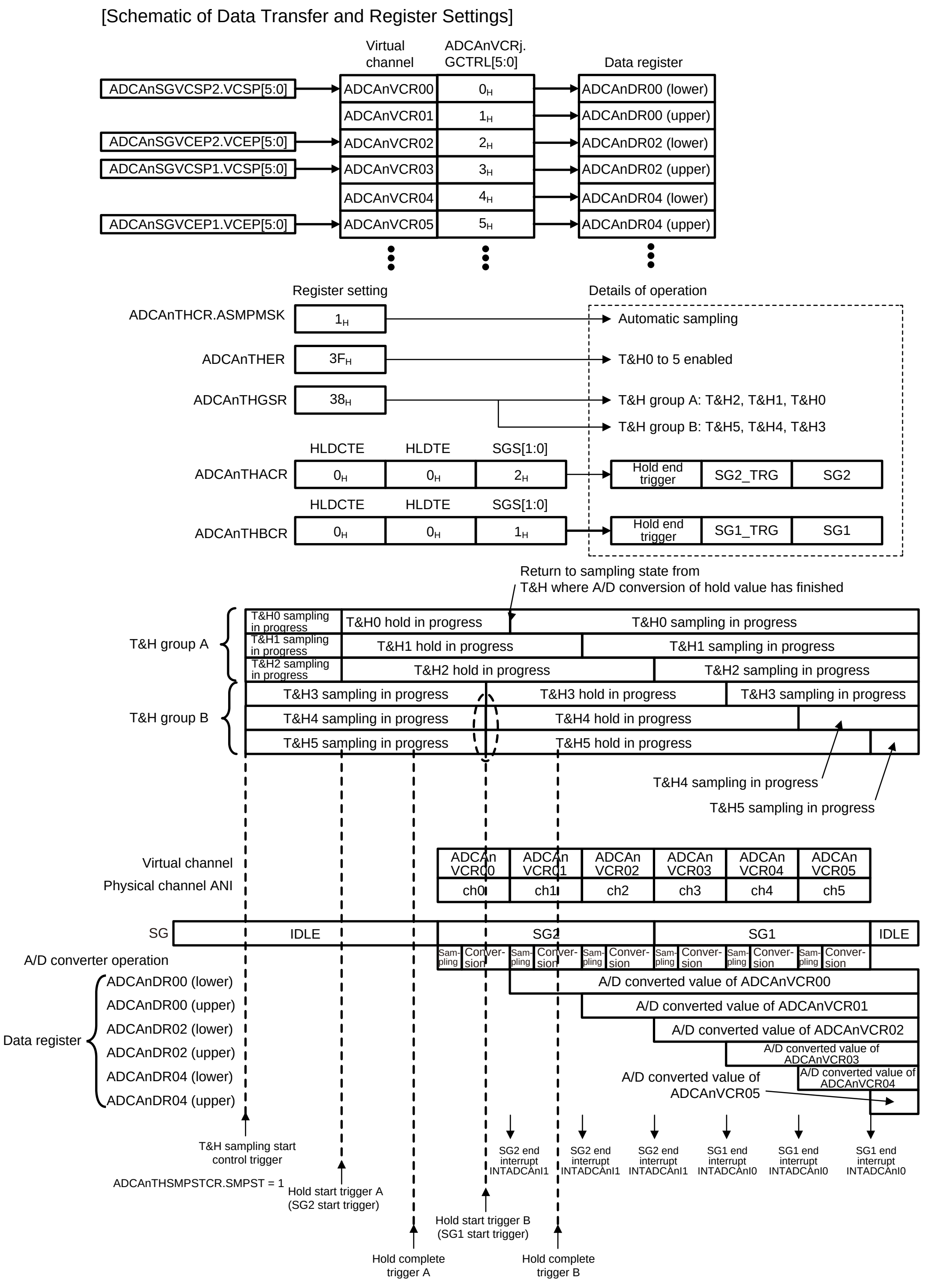
Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice in continuous scan mode

Figure 38.18 Example of Channel Repeat Operation 3

38.4.6 Example of Simultaneous Track and Hold Operation

Figure 38.19, Example of Simultaneous Track & Hold Operation 1 shows an operation example of simultaneous track and hold.



Conditions

Scan group 1: A/D conversion of virtual channels 3 to 5 in which a pin is scanned once in multicycle scan mode

Scan group 2: A/D conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

Figure 38.19 Example of Simultaneous Track & Hold Operation 1

CAUTION

- Do not specify the same physical channel in different groups.
- Two-cycle (or more) scan in multicycle scan mode and track & hold operation using continuous scan mode are prohibited.
- Because ADCAnTHSMPSTCR.SMPST is common to group A and group B, set SMPST after T&H operation for both group A and group B has been completed.
- If the hardware trigger is asserted before HLDCTE = 1 and HLDTE = 1 of ADCAnTHACR register or ADCAnTHBCR register are set and HLDST of ADCAnTHAHLDSTCR register or ADCAnTHBHLDSTCR register is written to hold T&H, scan operation starts. In that case, the channel switch opens with T&H staying in the sampling state. Therefore, all scan results are undefined. Do not assert the hardware trigger by setting HLDCTE = 1 and HLDTE = 1 before writing HLDST.
- Setting any channel from among 0 to 2 and any channel from among 3 to 5 in the same scan group is prohibited.
- Set the interval between T&H sampling start control trigger and hold start trigger to be 450 ns or more.
- Set the interval between hold start trigger and completion of the group A/D conversion to be 10 μs or less.
  In suspend mode, do not use T&H for the channel of a scan group with low priority setting, if suspend time exceeds 10 μs.

38.4.7 A/D Conversion with External Analog Multiplexer

The following figures show examples of A/D conversion in each case.

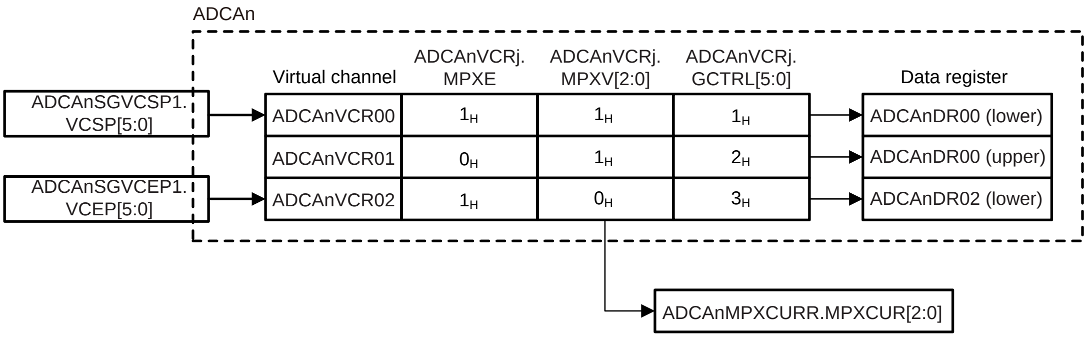
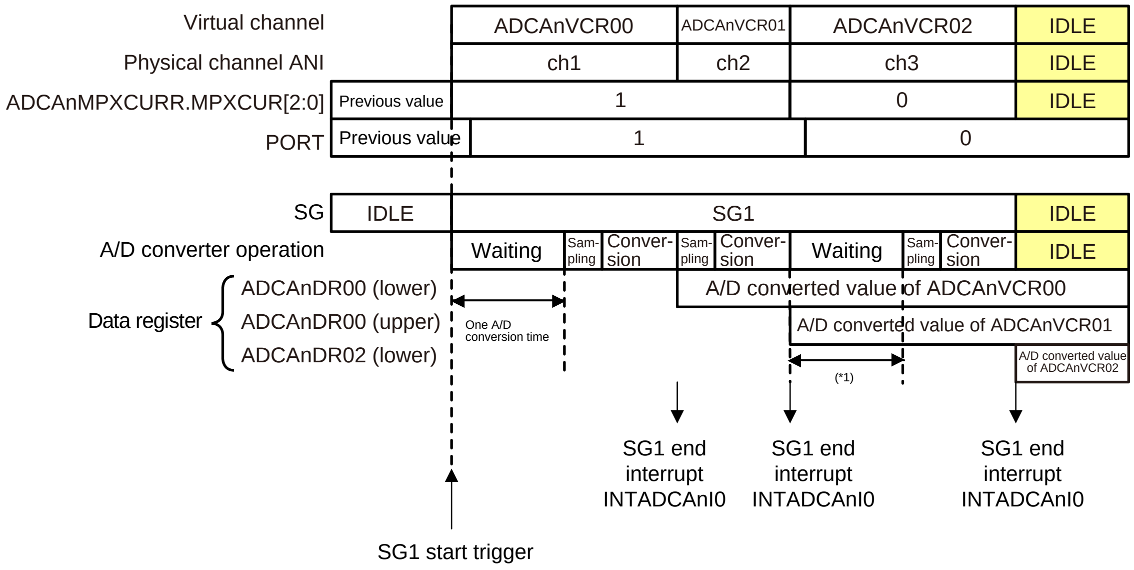


Figure 38.20 Schematic of Data Transfer and Register Settings



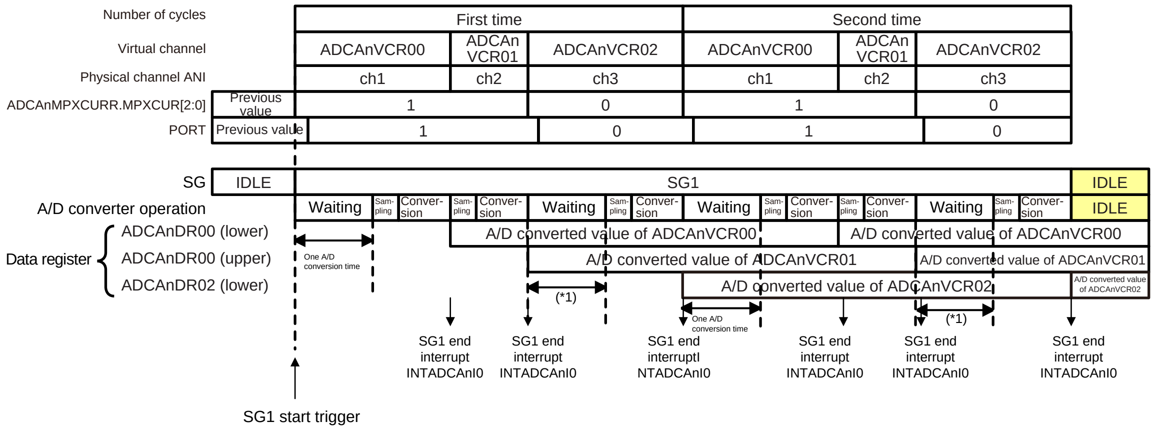
Conditions

Scan group 1: A/D conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

Configurable stabilization time for MPX analog input is selected for physical channel ANI 3. (ADCAnMPXSTBTSELR.STBTCH03=7_H)

Note 1. Waiting time is depended on the setting of ADCAnMPXSTBTR7 register.

Figure 38.21 A/D Conversion 1 at an External Analog Multiplexer



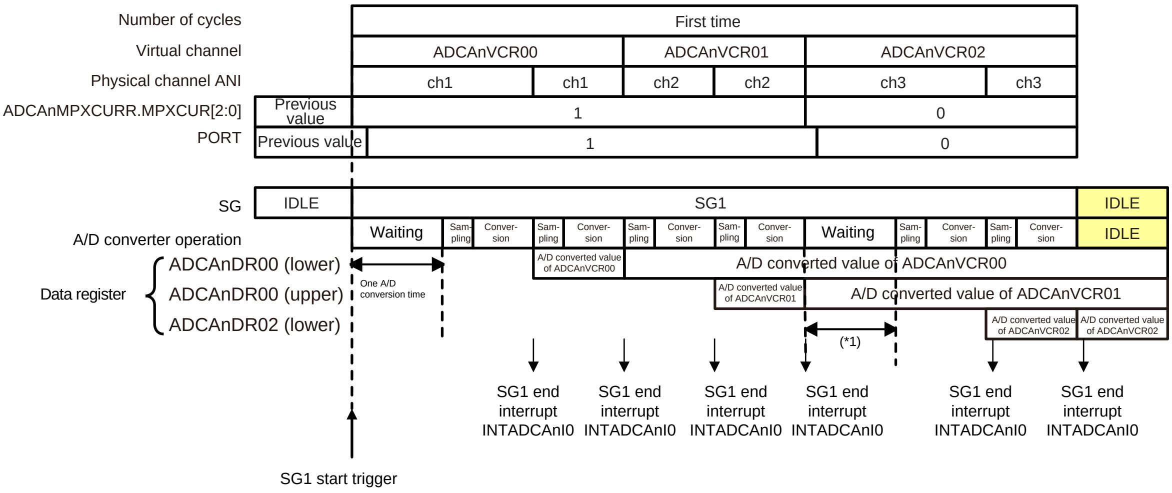
Conditions

Scan group 1: A/D conversion of virtual channels 0 to 2 using the two-cycle scan in multicycle scan mode

Configurable stabilization time for MPX analog input is selected for physical channel ANI 3. (ADCAnMPXSTBTSELR.STBTCH03=7_H)

Note 1. Waiting time is depended on the setting of ADCAnMPXSTBTR7 register.

Figure 38.22 A/D Conversion 2 at an External Analog Multiplexer



Conditions

Scan group 1: A/D conversion of virtual channels 0 to 2 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Configurable stabilization time for MPX analog input is selected for physical channel ANI 3. (ADCAnMPXSTBTSELR.STBTCH03=7_H)

Note 1. Waiting time is depended on the setting of ADCAnMPXSTBTR7 register.

Figure 38.23 A/D Conversion 3 at an External Analog Multiplexer

38.4.7.1 A/D Conversion with PWM-Diag Enabled

With the PWM-Diag function enabled, A/D conversion is performed by the signal from the PWM-Diag.

For details on the PWM-Diag function, see **Section 37, PWM Output/Diagnostic (PWM-Diag)**.

To control the A/D conversion, the A/D converter receives the setting information on the MPX by the A/D conversion trigger select (PWSA) signal. The flow of A/D conversion with PWM-Diag is as follows.

(1) Set the channel MPX value of the MPX to ADCAnPWDVCR.MPXV[2:0]. Up to 8 channels can be specified to the MPX.

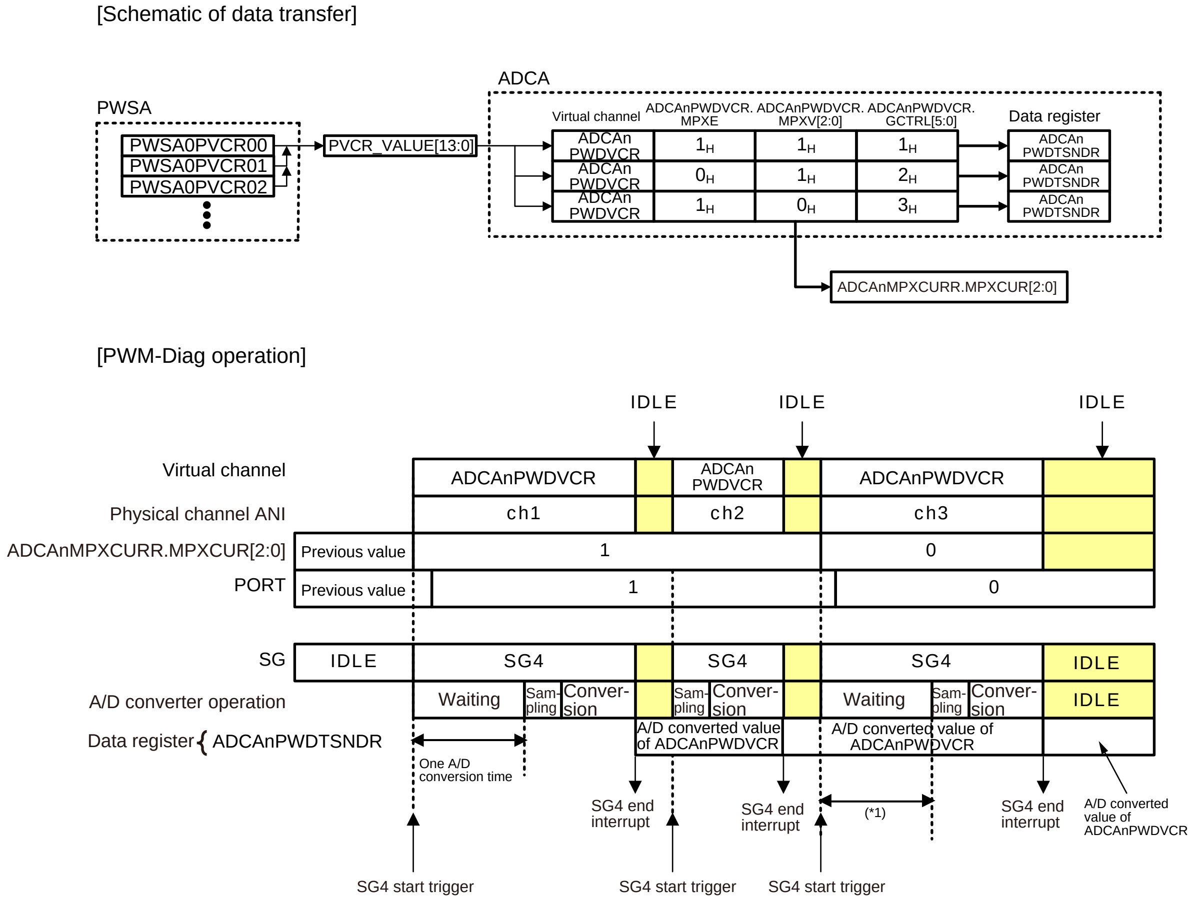
(2) The A/D conversion is started by the trigger signal PWSA_ADTRG from the PWM-Diag. In addition, when the MPX enable bit (ADCAnPWDVCR.MPXE) is 1, a wait of one A/D-conversion time or configured wait time which is set by ADCAnSTBTR0 to ADCAnSTBTR7 and ADCAnSTBTSELR0 to ADCAnSTBTSELR4 is inserted before A/D conversion is performed.

(3) At the end of A/D conversion, the scan end is notified to the PWM-Diag.

CAUTION

As the trigger signal PWSA_ADTRG of PWM-Diag function has a higher-priority than SGx_TRG (x = 1 to 3), the operation of other scan groups may be kept waiting until the PWM-Diag function is ended.

Figure 38.24, PWM-Diag Operation shows an example of PWM-Diag operation using an MPX.



Configurable stabilization time for MPX analog input is selected for physical channel ANI 3. (ADCAnMPXSTBTSELR.STBTCH03 = 7_H)

Note 1. Waiting time is depended on the setting of ADCAnMPXSTBTR7 register.

Figure 38.24 PWM-Diag Operation

38.4.8 Example of Synchronous Suspend and Resume Operation

Figure 38.25, Example of Synchronous Suspend and Resume Operation shows an example of synchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

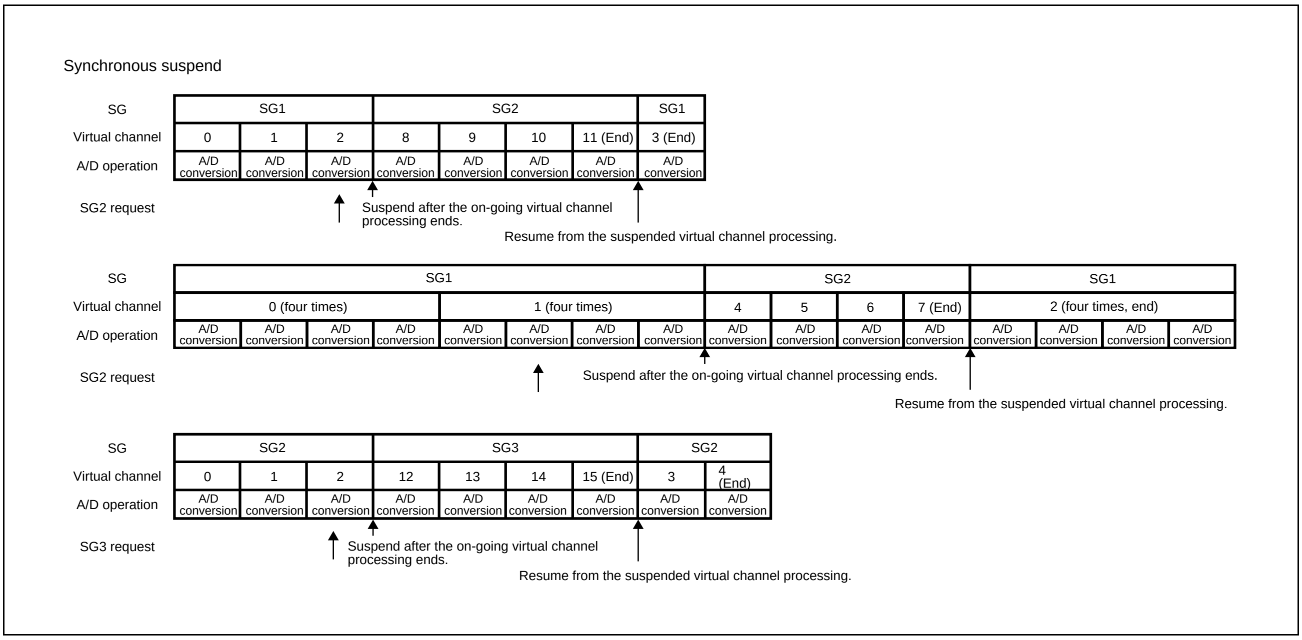


Figure 38.25 Example of Synchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

38.4.9 Example of Asynchronous Suspend and Resume Operation

Figure 38.26, Example of Asynchronous Suspend and Resume Operation shows an example of asynchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

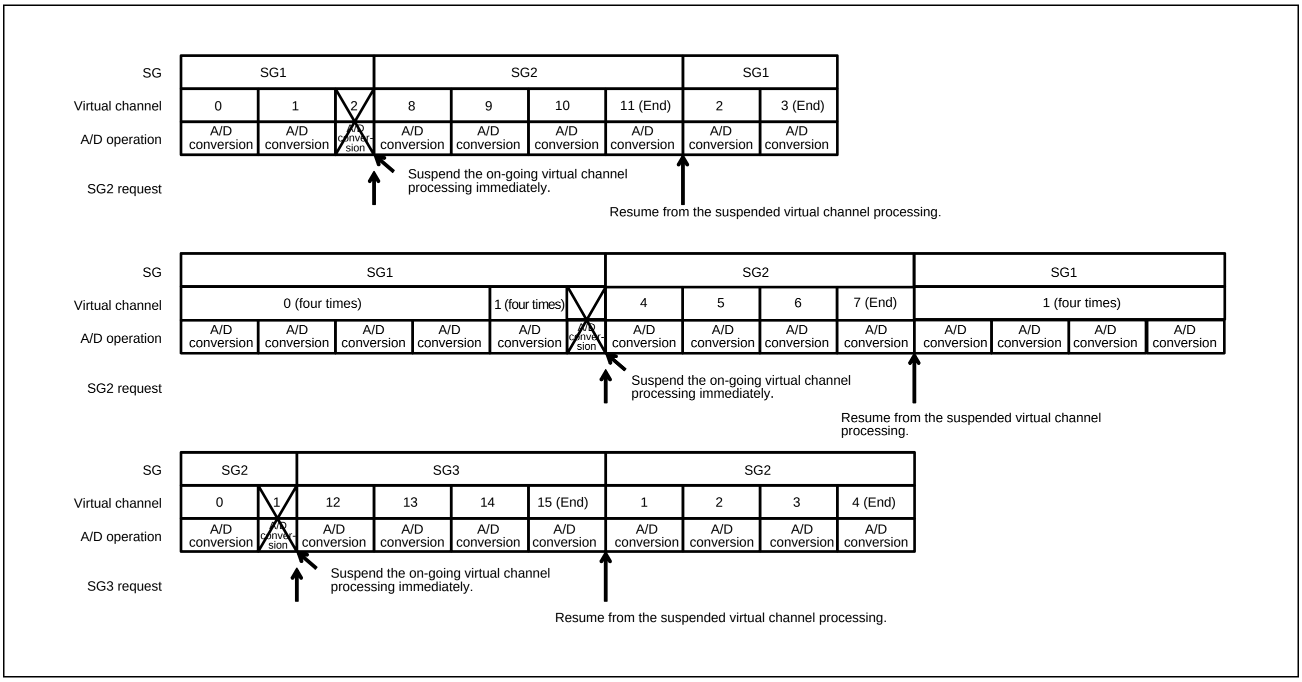


Figure 38.26 Example of Asynchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

38.4.10 Error Detecting Functions

ADCAn covers upper-limit error, lower-limit error, and overwrite error.

38.4.10.1 Upper-Limit/Lower-Limit Error Detecting Function

The upper-limit/lower-limit error detecting function determines whether the A/D converted data is larger than the upper-limit table ADCAnULLMTBR0.ULMTB[11:0] to ADCAnULLMTBR7. ULMTB[11:0] or smaller than the lower-limit table ADCAnULLMTBR0.LLMTB[11:0] to ADCAnULLMTBR7.LLMTB[11:0] at the end of A/D conversion.

38.4.10.2 Overwrite Error Detecting Function

If the ADCAnDIRj register or ADCAnDRj register of a virtual channel is not read while ADCAnDIRj.WFLG = 1 (A/D converted value is stored) and the next A/D converted value is written in the ADCAnDRj register, an overwrite error is detected.

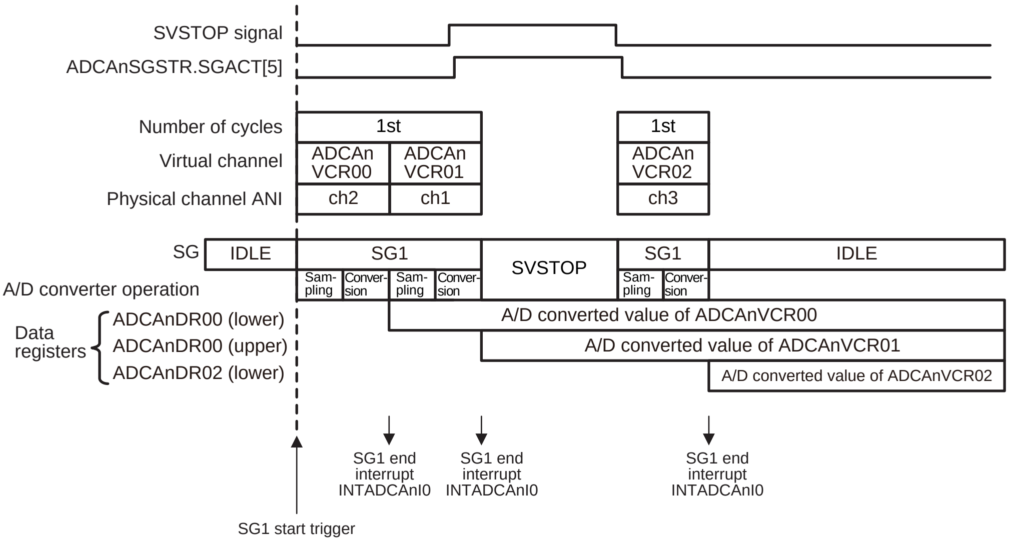
38.4.10.3 SVSTOP Operation

The SVSTOP function is supported by the SVSTOP signal sent from the on-chip debugger control unit. The SVSTOP function stops conversion of the A/D converter when the SVSTOP signal is input during an emulation break. While the SVSTOP signal is high, reading registers ADCAnDRj, ADCAnDIRj, ADCAnSGSTR, ADCAnULER, ADCAnOWER, ADCAnPWDTSNDR, ADCAnPWDDIR, ADCAnULEVCFR0, and ADCAnULEVCFR1 by the external access does not affect these registers.

When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 0, ADCAnSGSTR.SGACT[5] is set to 1 to make a transition to the SVSTOP state. Hardware triggers and software triggers are valid in the SVSTOP state. When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 1, the ADCA does not make a transition to the SVSTOP state. ADHALT (forced termination of A/D conversion) and scan group stop by ADCAnSGSTPCRx should not be performed in the SVSTOP state.

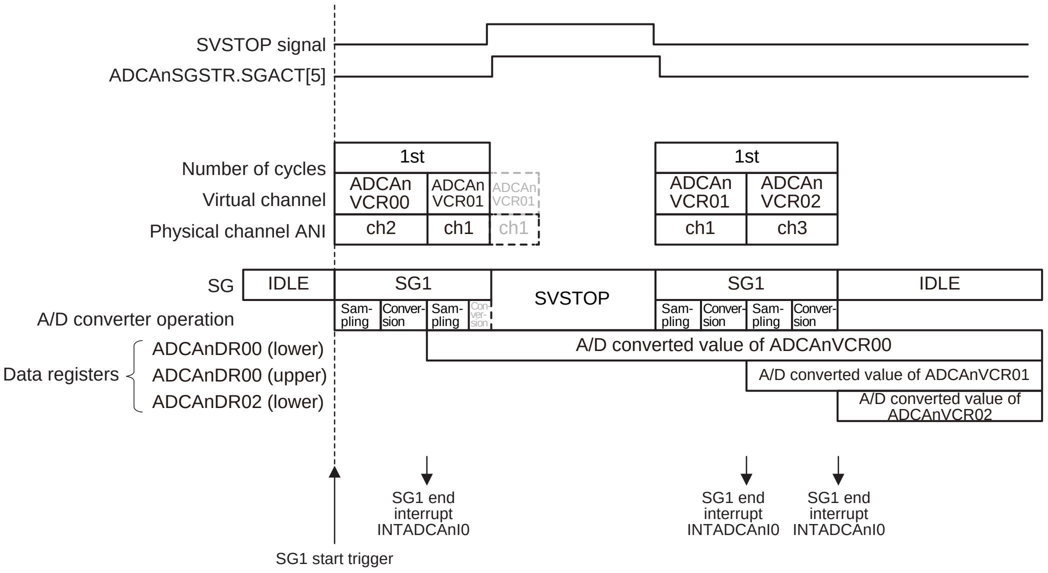
In operations for synchronous suspension, a new start trigger cannot be accepted over the time from when the high level is input to SVSTOP to the completion of conversion on the channel where conversion is currently proceeding. This time can be up to the time taken for one A/D conversion.

The following example illustrates a SVSTOP operation example.



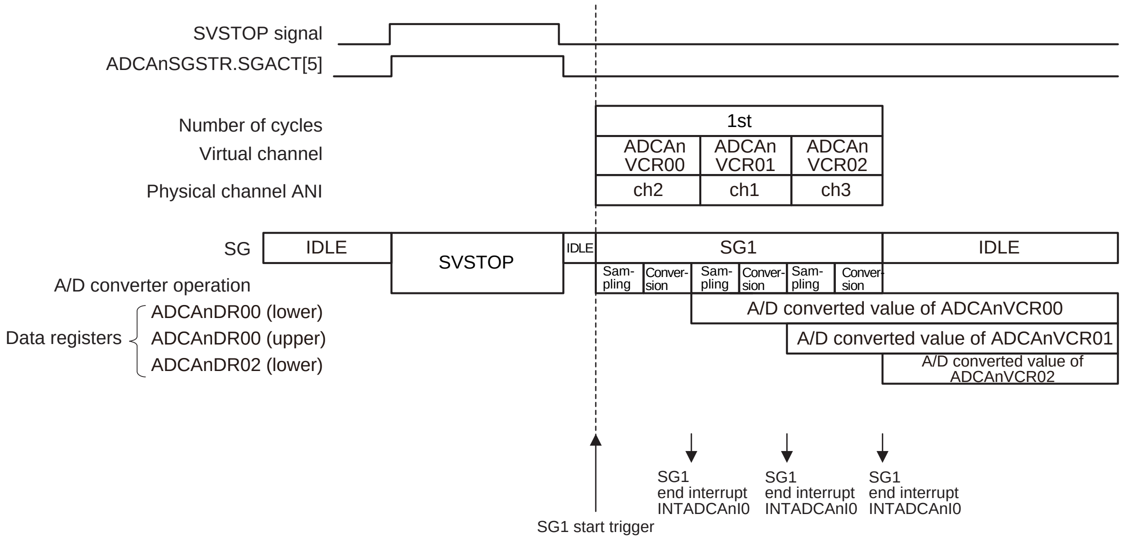
Note: An SG start trigger is accepted even during SVSTOP after the completion of conversion.

Figure 38.27 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



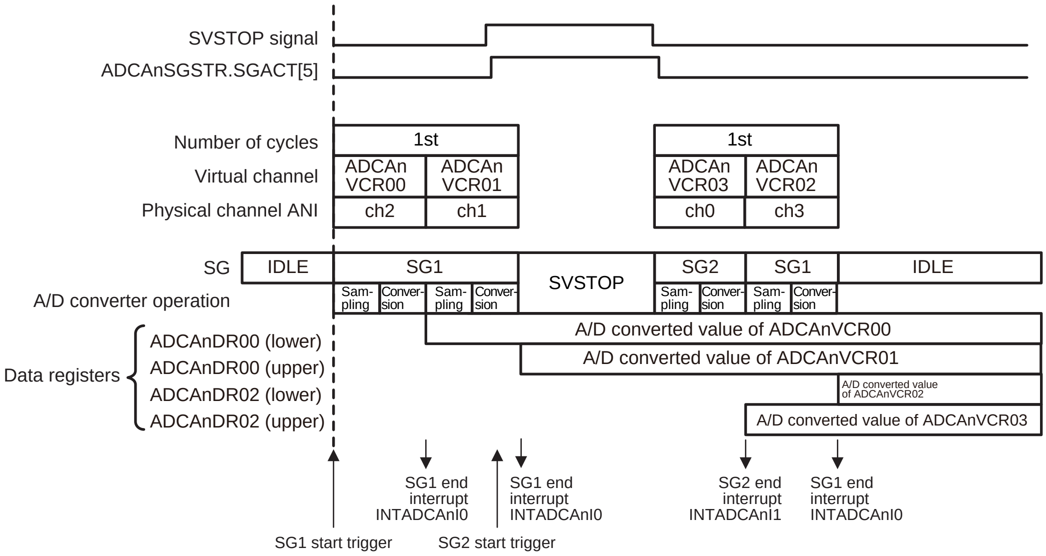
Note: An SG start trigger is accepted even during SVSTOP.

Figure 38.28 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 10 and ADCAnEMU.SVSDIS = 0)



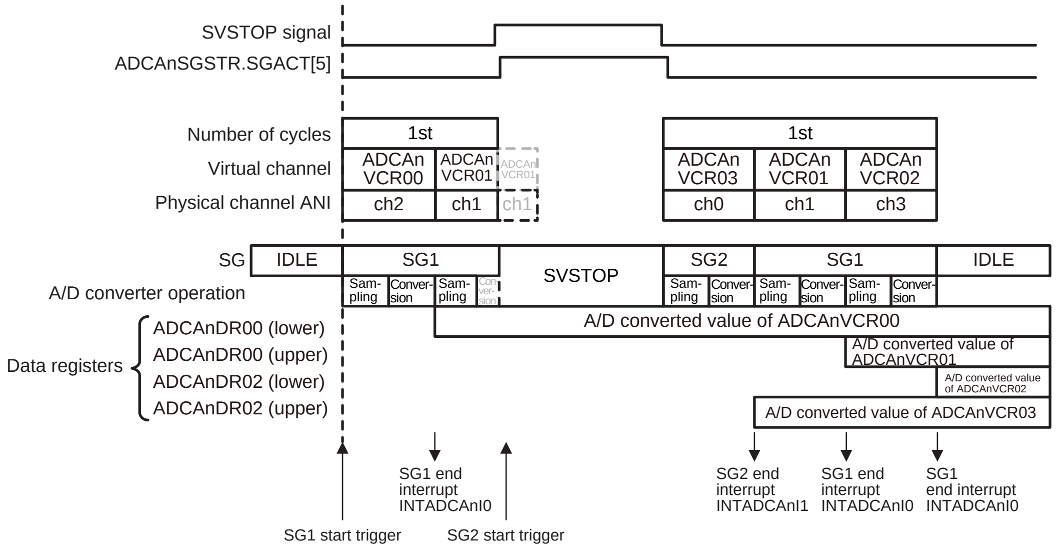
Note: An SG start trigger is accepted even during SVSTOP.

Figure 38.29 Example of SVSTOP Operation in the IDLE State (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



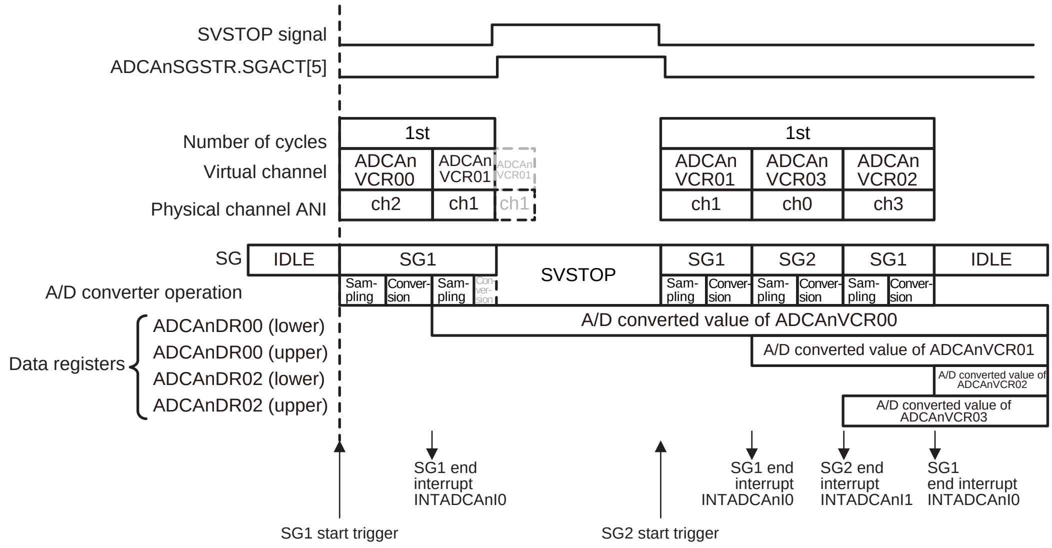
Note: After the completion of conversion, the SG start trigger is accepted even during SVSTOP.

Figure 38.30 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 00, ADCAnEMU.SVSDIS = 0)



Note: An SG start trigger is accepted even during SVSTOP.

Figure 38.31 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)



Note: An SG start trigger is accepted even during SVSTOP.
In cases of conflict between clearing of SVSTOP and setting of the scan group start trigger bit, conversion of the suspended scan group is resumed regardless of its priority.

Figure 38.32 Conflict of SVSTOP Clear and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)

38.4.11 Activating Scan Group by a Hardware Trigger

Scan group x can be activated by the hardware trigger input to SGx_TRG. As for the hardware trigger sources to be used, see **Table 38.78, List of A/D Conversion Hardware Triggers**. When activating SGx_TRG by the hardware trigger, set the peripheral function to be used by the trigger and set the start trigger in the A/D conversion trigger select control register (ADCAnSGTSELx).
A hardware trigger from external trigger input pin requires digital filter setting. For details, see **Section 2A.12, Noise Filter & Edge/Level Detector**, **Section 2B.12, Noise Filter & Edge/Level Detector** and **Section 2C.12, Noise Filter & Edge/Level Detector**. More than one start trigger can be specified.

38.4.11.1 Stopping Scan Group by ADHALT

Setting ADCAnADHALTR.HALT (A/D force halt trigger) to 1 forcibly halts the A/D conversion and clears the scan group status register (ADCAnSGSTR). The error flag of ADCAnULER, ADCAnULEVCFR0, and ADCAnULEVCFR1 (upper limit/lower limit error register) is not cleared. When ADCAnADHALTR.HALT is set, make sure that ADCAnSGSTR.SGACT has been cleared.

38.4.12 Scan End Interrupt Request

Scan group x can issue a scan end interrupt request (INT_SGx) to INTC. If ADIE of ADCAnSGCRx is set to 1, INT_SGx can be output after the SGx scan ends. If ADIE of ADCAnSGCRx is set to 0, the INT_SGx output when the SGx scan ends can be disabled. If ADIE of ADCAnVCRj is set to 1, INT_SGx can be output when A/D conversion for virtual channel j in SGx ends. If ADIE of ADCAnVCRj is set to 0, the INT_SGx output when A/D conversion for virtual channel j in SGx ends can be disabled. Since SGx scan ending is simultaneous with A/D conversion ending for virtual channel j in SGx when ADIEs of both ADCAnSGCRx and ADCAnVCRj are set to 1, INT_SGx occurs only once.

Example 1: A scan is executed for virtual channel 0 or 1 in SG1 when ADIE of ADCAnSGCR1 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 0.
INT_SG1 is output when A/D conversion ends for virtual channel 0.

Example 2: A scan is executed for virtual channel 0 or 1 in SG2 when ADIE of ADCAnSGCR2 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 1.
INT_SG2 is output when A/D conversion ends for virtual channel 0 and virtual channel 1.

Example 3: A scan is executed for virtual channel 0 or 1 in SG3 when ADIE of ADCAnSGCR3 is 1, ADIE of VCR0 is 0, and ADIE of VCR1 is 0.
INT_SG3 is output when a scan ends (when A/D conversion for virtual channel 1 ends).

Furthermore, the DMA can be started when scan ends.

For the setting of DMA, see **Section 8, DMA Controller**.

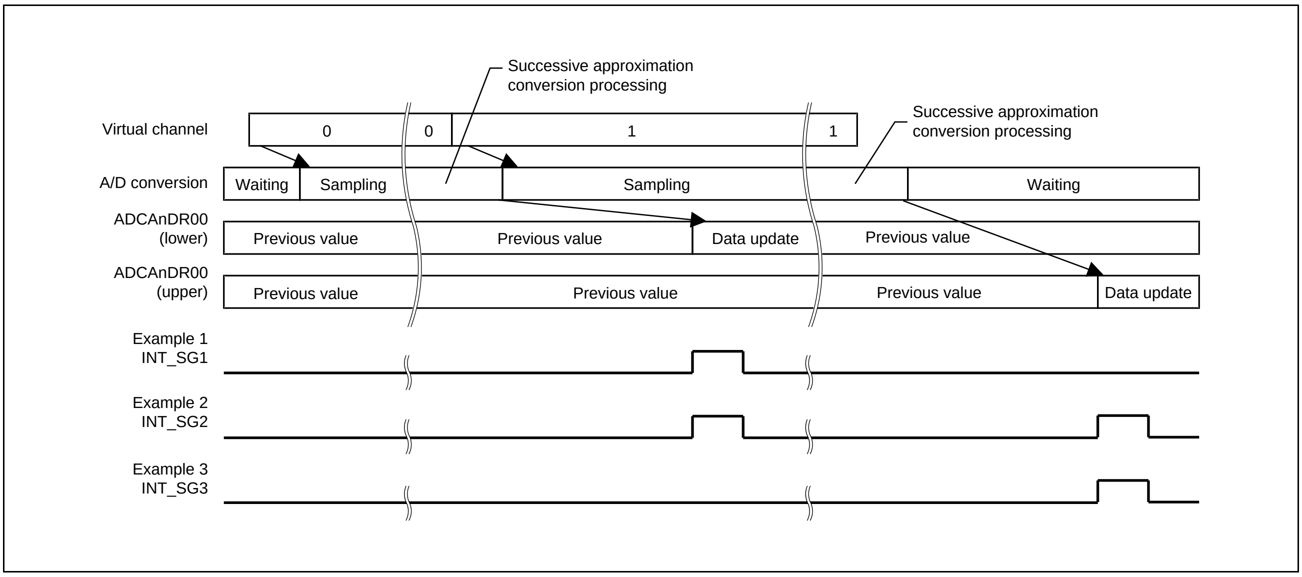


Figure 38.33 Example of a Scan Conversion End Interrupt Occurrence

38.4.13 A/D Error Interrupt Request

ADCA can issue an A/D error interrupt request (INT_ADE) to INTC. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 1, the OR condition of the error source is issued as INT_ADE. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 0, INT_ADE does not output an interrupt.

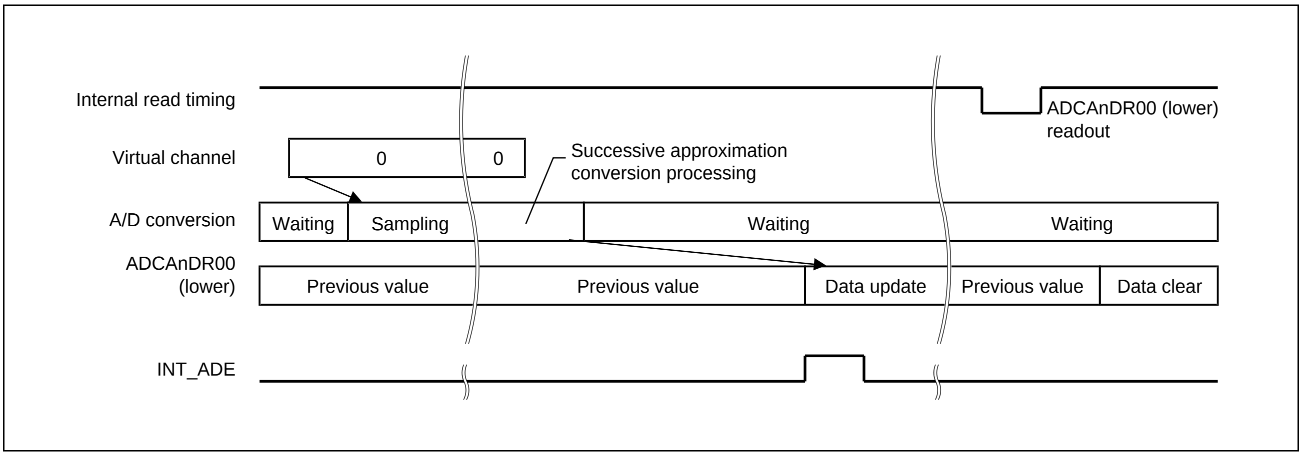


Figure 38.34 A/D Error Interrupt (Example: Overwrite Error)

38.5 Self-Diagnostic Function

To check the ADCAn function, the following self-diagnostic functions are available.

Section 38.5.1, Diagnosis of A/D Conversion Circuit

Section 38.5.2, Diagnosis of Channel Multiplexer

Section 38.5.3, Diagnosis of Open Pins

Section 38.5.4, Diagnosis of T&H Circuit

The overview of the self-diagnostic functions is shown in the figure below. A detailed description is given in the following sections.

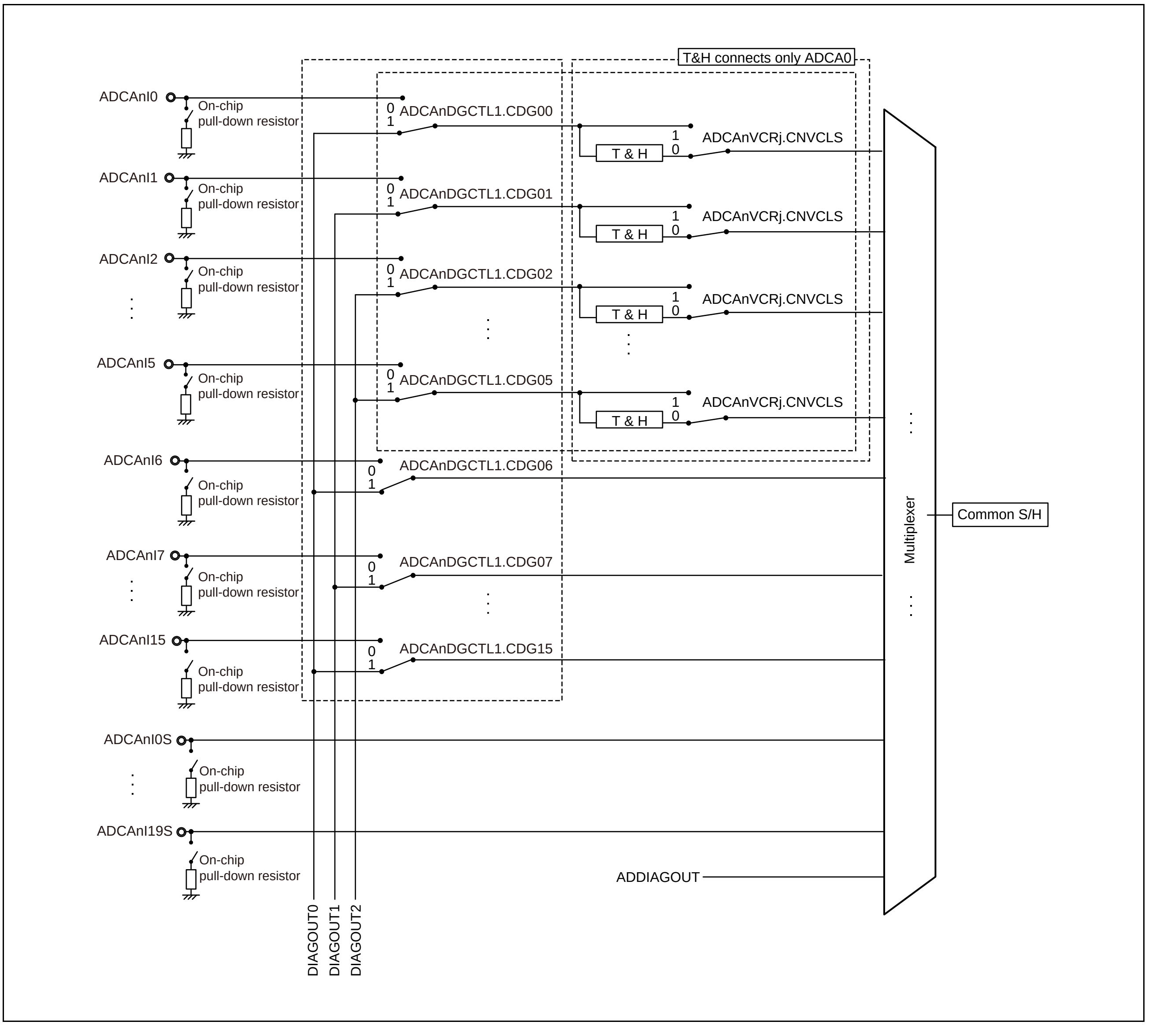


Figure 38.35 Overview of Self-Diagnostic Functions

NOTE

The functions in the dashed-line frames depend on the product.

38.5.1 Diagnosis of A/D Conversion Circuit

This function checks whether the A/D converter is operating normally by verifying the A/D conversion for self-diagnostic voltage (ADDIAGOUT) and the result of conversion. If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the A/D converter are as follows:

- As the self-diagnostic voltage (ADDIAGOUT) level, AnV_{REF}, $2/3AnV_{REF}$, $1/3AnV_{REF}$, $1/2AnV_{REF}$, and AnVSS are selectable by the PSEL[2:0] bits in the ADCAnDGCTL0 register.
- Self-diagnosis of the A/D converter is enabled by performing A/D conversion on one of SG1 to SG3.

38.5.1.1 Diagnostic Procedure

The diagnostic procedures are shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set an arbitrary bit of ADCAnVCRj.GCTRL[5:0] to 100100_B to select the diagnosis channel.
7. Set ADCAnVCRj.ADIE = 1 to enable the A/D conversion end interrupt.
8. Set ADCAnSGVCSPx to specify the start pointer of virtual channel.
9. Set ADCAnSGVCEPx to specify the end pointer of virtual channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
    1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
    2. Clear ADCAnDGCTL0.PSEL[2:0].
    3. Clear ADCAnADCR.DGON.

38.5.2 Diagnosis of Channel Multiplexer

This function checks whether the path from the analog input to the A/D converter is normal.
Set the A/D conversion reference voltage (DIAGOUT0, DIAGOUT1, DIAGOUT2) by ADCAnDGCTL0.PSEL[2:0] and the channels to be connected by the ADCAnDGCTL1 register to perform A/D conversion using multiple analog channels.
If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the channel multiplexer are as follows:

- Channels for diagnosis can be arbitrarily selected from among ADCA0I0 to ADCA0I15 and ADCA1I0 to ADCA1I15.
- As the self-diagnostic voltage level, 2/3AnV$_{REF}$, 1/3AnV$_{REF}$, and 1/2AnV$_{REF}$ are selectable and one of the three reference voltage levels can be allocated to each channel.

Table 38.84 Selection of Channel to be Diagnosed

Connection	Select Channel
DIAGOUT0	Channels 0, 3, 6, 9, 12, and 15
DIAGOUT1	Channels 1, 4, 7, 10, and 13
DIAGOUT2	Channels 2, 5, 8, 11, and 14

- Self-diagnosis of the channel multiplexer is enabled by performing A/D conversion on one of SG1 to SG3 by using multiple channels.

38.5.2.1 Diagnostic Procedure

The diagnostic procedure is shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Use two or more ADCAnVCRj registers.
   Set ADCAnVCRj.GCTRL[5:0] bits to select physical channels.
   Set ADCAnVCRj.ADIE bit to enable the A/D conversion end interrupt.
7. Set ADCAnSGVCSPx register to specify the start pointer of virtual channel.
8. Set ADCAnSGVCEPx register to specify the end pointer of virtual channel.
9. Set ADCAnDGCTL1 register to specify the physical channel to the self-diagnostic channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
  1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
  2. Clear ADCAnDGCTL0.PSEL[2:0].
  3. Clear ADCAnADCR.DGON.

38.5.3 Diagnosis of Open Pins

This function detects whether the analog input pin (ADCAnIm, ADCAnImS) is open due to disconnection, etc. An internal pull-down resistor can be connected to diagnose the analog input pin.
Connect the analog input pin (ADCAnIm, ADCAnImS) with the pull-down resistor for self-diagnosis for A/D conversion of the target channels.

Figure 38.36 Setting of On-chip Pull Down Resistor

When there is a disconnection, the conversion result is almost 0 V and it indicates an open detection.

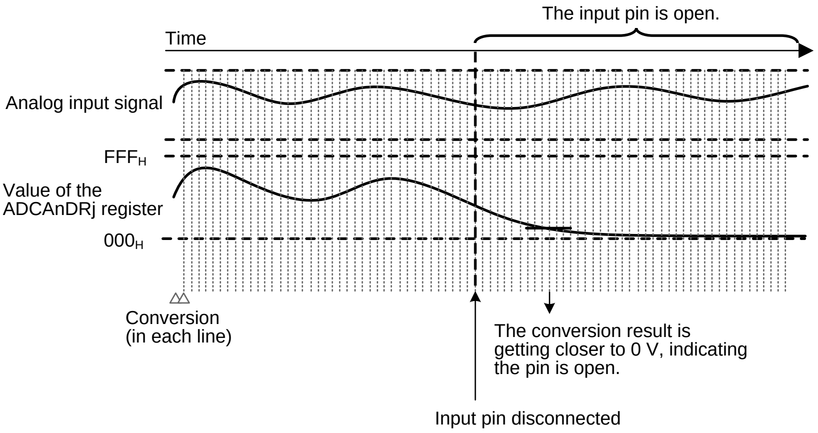


Figure 38.37 Analog Input Signal Disconnection Detection

CAUTIONS

1. The pull-down resistors must not be connected during normal A/D conversion operation. Connected pull-down resistors may lead to a drop in the input voltage and result in erroneous A/D conversion results.
2. When the analog input voltage is nearly equal to the voltage level which is pulled down, a disconnection cannot be detected by this function.

38.5.3.1 Diagnostic Procedure

1. Set the ADCAnPDCTL1.PDNA or ADCAnPDCTL2.PDNB bit which correspond to analog input pins (ADCAnIm, ADCAnImS) to be diagnosed to enable the pull down resistor.
2. Generate the start trigger of scan group to perform the A/D conversion.
3. Perform the A/D conversion multiple times on the same analog input.
4. Monitor the channel's A/D conversion results and check if any result declines to almost 0 V.

38.5.4 Diagnosis of T&H Circuit

This function is used to diagnose proper operation of the T&H0 to T&H5 circuits for ADCA0I0 to ADCA0I5.

Virtual channel registers 33 to 35 (ADCA0VCR33 to ADCA0VCR35) are used exclusively for comparison of the potential conversion result using the T&H circuit and that obtained without using the T&H circuit to detect a failure of the T&H circuit.

For this diagnosis, the ADCA0THACR.HLDCTE is set to 1 (ADCA0THBCR.HLDCTE = 1) and ADCA0THACR.HLDTE to 0 (ADCA0THBCR.HLDTE = 0) and the A/D conversion trigger is used as the hold start/end trigger. Connect the reference voltage signal (DIAGOUT0, DIAGOUT1, or DIAGOUT2) selected by ADCAnDGCTL0.PSEL[2:0] to the target channels for diagnosis by using the ADCAnDGCTL1 register.

38.5.4.1 Diagnostic Procedure (in case of T&H circuit ch0 diagnosis)

1. Set ADCA0ADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCA0DGCTL0.PSEL[2:0] = 001_B to select 1/3AnV$_{REF}$ voltage level.
4. Set ADCA0ADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set ADCA0DGCTL1.CDG0 = 1 to enable DIAGOUT0.
7. Set ADCA0VCR33.GCTRL[5:0] to ADCAVCR35.GCTRL[5:0] to 000000_B to select physical ch0.
8. Set ADCA0VCR33.CNVCLS and ADCA0VCR34.CNVCLS to1 to select normal conversion.
9. Set ADCA0VCR35.CNVCLS = 0 to select hold value conversion.
10. Set ADCA0THACR.SGS[1:0] = 01_B to select SG1 to “T&H group A”.
11. Set ADCA0THER.TH0E = 1 to enable T&H circuit ch0.
12. Set ADCA0THGSR.TH0GS = 0 to select T&H circuit ch0 to “T&H group A”.
13. Set ADCA0SGVCSP1.VCSP[5:0] =100001_B to select SG1 start pointer to VCR33.
14. Set ADCA0SGVCEP1.VCEP[5:0] = 100011_B to select SG1 end pointer to VCR35.
15. Set ADCA0DGCTL0.PSEL[2:0] = 011_B to select 2/3AnV$_{REF}$ voltage level.
16. Set ADCA0THSMPSTCR.SMPST = 1 to execute T&H sampling.
17. Wait for 500 ns.
18. Set ADCA0SGSTCR1.SGST = 1 to execute SG1 A/D conversion.
19. Read ADCA0DIR33 to ADCA0DIR35, and check A/D conversion result to see if SG1 A/D conversion has finished.

NOTES

To clear ADCAnADCR.DGON, follow the procedure below:

1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
2. Use the ADCAnTHER register to disable the diagnosed T&Hk.
3. Clear ADCAnDGCTL0.PSEL[2:0].
4. Clear ADCAnADCR.DGON.

38.5.4.2 Diagnosis Mechanism

(1) A reference voltage “A” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.

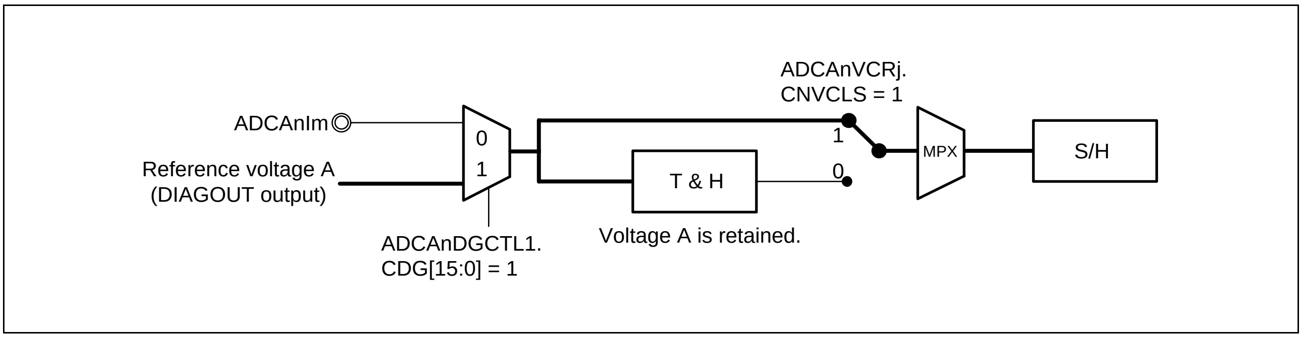


Figure 38.38 T&H Circuit Diagnostic Mechanisms (1)

(2) A reference voltage “B” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit still holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.
Note that since reference voltage “A” is being held, reference voltage “B” is not held.

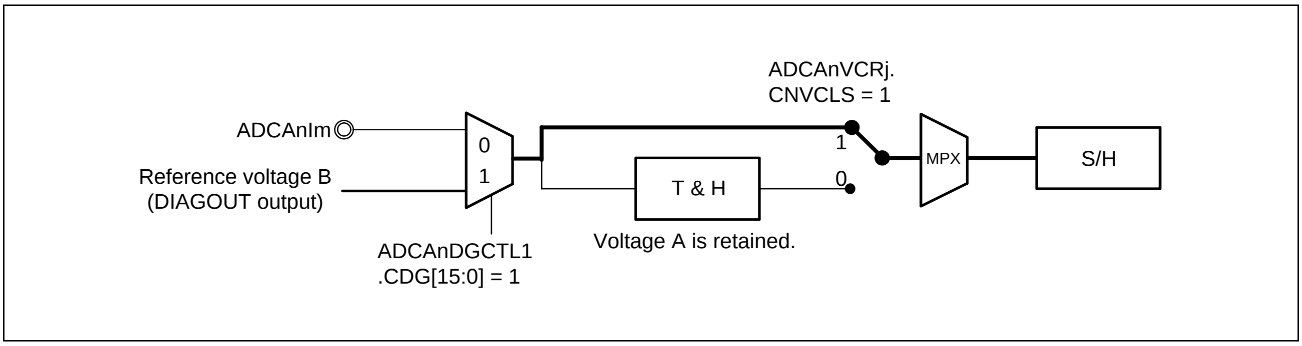


Figure 38.39 T&H Circuit Diagnostic Mechanisms (2)

(3) An A/D conversion is performed using the T&H circuit. The T&H circuit continues to hold the voltage A.

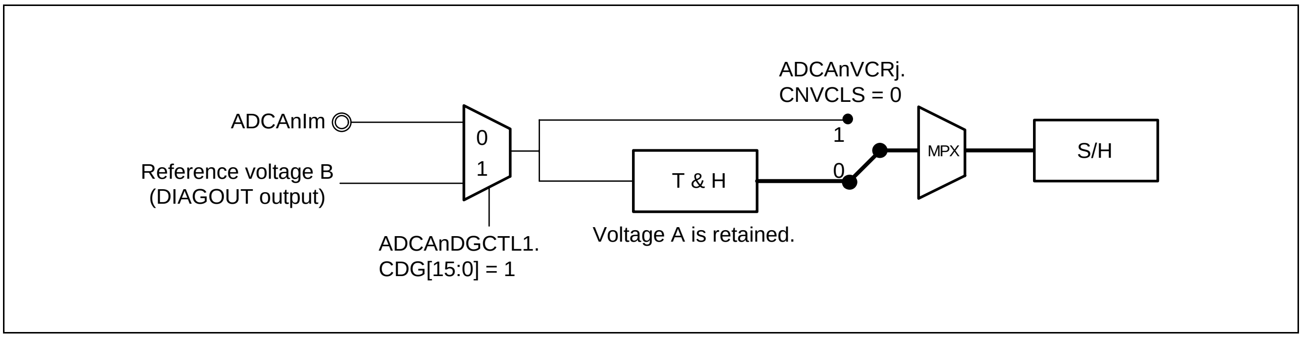


Figure 38.40 T&H Circuit Diagnostic Mechanisms (3)

(4) The diagnosis of T&H circuit is successful if the following results are obtained:

1. The first result (step 1) is voltage “A”.
2. The second result (step 2) is voltage “B”.
3. The last result (step 3) is voltage “A” again.

38.6 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined as follows:

- Resolution
  Digital output code value from the A/D converter
- Quantization error
  An error essentially contained in the A/D converter, which is assumed as 1/2 LSB (**Figure 38.41, Definition of A/D Conversion Accuracy**).
- Offset error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value 000_H to 001_H. However, the quantization error is not included.
- Full scale error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from FFE_H to FFF_H. However, the quantization error is not included.
- DNL (Differential nonlinear error)
  Deviation between the ideal digital output code width (Vq) and the actual digital output code width (Va), which is assumed as (Va – Vq)/Vq. However, the offset error, the full scale error, and the quantization error are not included.
- INL (Integral nonlinear error)
  Deviation of the actual value from the ideal A/D conversion characteristics between the zero voltage and the full scale voltage, which is assumed as an integral of DNL from 000_H to a digital output code. However, the offset error, the full scale error, and the quantization error are not included.
- Absolute accuracy
  Deviation between the digital value and the analog input value. The offset error, the full scale error, the quantization error, DNL, and INL are included.
- Overall error
  Deviation between the digital value and the analog input value. The offset error, the full scale error, DNL, and INL are included, but the quantization error isn't included.

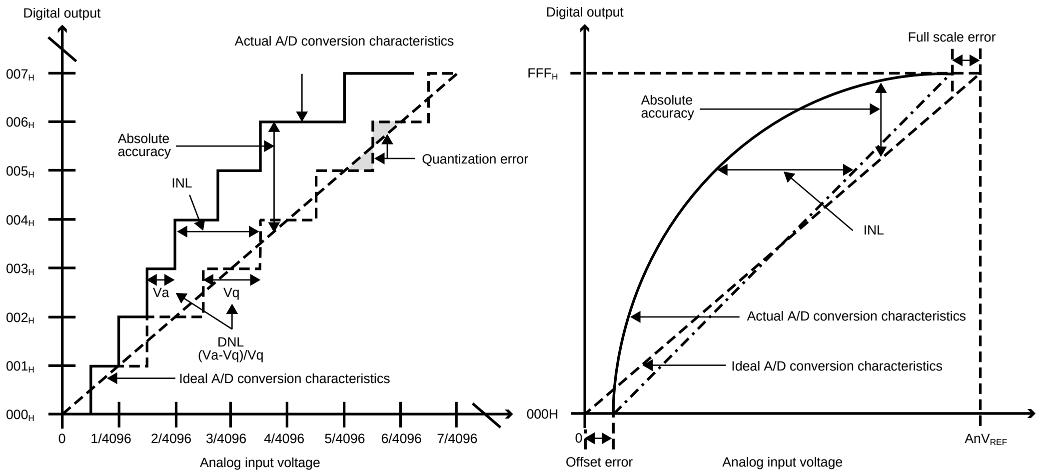


Figure 38.41 Definition of A/D Conversion Accuracy

38.7 Usage Notes

38.7.1 Range of Channel Input Voltage

CAUTION

ADCAnIm and ADCAnImS input voltages should be used within the specification range. When the channel input voltage exceeds AnVREF or falls below AnVSS, an over-voltage/injected current condition applies.

1. When an over-voltage is applied to ADCAnIm pins (within the specified injected current range) and at the same pins the ADC self-diagnosis (diagnosis of channel multiplexer) is executed an offset voltage to the diagnosis voltage will be measured.
2. When an over-voltage is applied to ADCAnIm pins (within the specified injected current range) and at adjacent pins the ADC self-diagnosis (diagnosis of channel multiplexer) is executed an AD conversion result within the range of TESH0SN will be measured.

Case: Injected current is applied at only two pins (ADCA0I0 and ADCA0I1 in the example below) where the ADC self-diagnosis is executed.

- ADCA0I0 and ADCA0I1 with applied injected current, but ADC self-diagnosis executed (CDG0-15 = 1 set to DIAGOUTn selected).
  → Conversion result of TESH0SN with offset voltage
- ADCA0I2-15 without applied injected current, but ADC self-diagnosis executed (CDG0-15 = 1 set to DIAGOUTn selected).
  → Conversion result of TESH0SN without offset voltage

38.7.2 Notes on Application Design

(1) Analog Input Pins (ADCAnIm, ADCAnImS)

- Ensure that the input voltages on the ADCAnIm and ADCAnImS pins are within the specified ranges. We recommend using diodes with VF of 0.3V or below to form a clamp to avoid the input of voltages at or above AnVREF and at or below AnVSS. The results of conversion for input voltages at or above AnVREF and at or below AnVSS are undefined and so are not guaranteed. Input of such voltages can also affect the results of conversion on other channels.
- Reduce noise on the analog input pins (ADCAnIm and ADCAnImS) by connecting a resistor Re between the pins and the external sources of analog input signals for conversion and capacitor Ce to the AnVSS pins.
- Avoid analog signal lines crossing digital signal lines and vice versa, since this can introduce noise and reduce performance in A/D conversion.
- We recommend avoiding the driving of large currents through input and output pins near the ADCAnIm and ADCAnImS pins and toggled signals in particular should be kept away from these pins.
- If you are using the standby functions, confirm the Force halt or the Normal end with the flow for **Section 38.4.3, Ending/Stopping A/D Conversion**. Then, set the ADCAnTHER.THkE bits, which are to be effective on standby, to 0.
- If you are using the LPS on ADCA0 (also when standby function is used), set the ADCA0SGCRx.TRGMD bit to 1 and the ADCA0SGTSELx.TxSEL bits to SEQTRG (LPS). For details, see **Section 15, Low-Power Sampler (LPS)**.
- Do not connect a channel to be used with the T&H function to an external analog multiplexer.

- Changes to physical and virtual channels during operation while the T&H function is in use is prohibited.
- Writing to PWM-Diag-related registers while PWM-Diag is not in use is prohibited.

(2) Power Wiring

The following methods are recommended to minimize the influence of switching noise from digital circuits on A/D converter accuracy.

- Connect markedly thick wiring patterns to the mesh pattern or connect solid patterns to the power-supply lines.
- Insert bypass capacitors between power-supply pins (EVCC, BVCC, and AnVREF) and ground pins (EVSS, BVCC, and AnVSS).
- We recommend separating the analog power supply (AnVREF) from the digital power supplies (EVCC, BVCC) and providing the voltages from a series regulator. If the analog power supply is to come from the same source as that of the digital power supplies, wire the analog and digital power supplies to an electrolytic capacitor, and provide separate wiring patterns on the board.
  We also recommend inserting a chip inductor in the input for the analog power supply. Furthermore, earth the analog and digital grounds to the same point on an electrolytic capacitor, and provide separate wiring patterns for the grounds on the board.
  The analog power supply also serves as the analog reference voltage for this product.

(3) Variation in A/D Converted Data

The effects of noise and variations in the power supply voltages lead to dispersal of the results of A/D conversion. Furthermore, noise on the analog input pins (ADCAnIm and ADCAnImS) or on the reference voltage input pins (AnVREF and AnVSS) can lead to the results of A/D conversion being incorrect.

Apply software processing to avoid ill effects on the system of fluctuations in or incorrectness of the results of A/D conversion.

Examples of software handling are described below.

- Use averaged values from several rounds of A/D conversion
- Execute A/D conversion for several time and omit extreme results
- Repeat the processing for abnormalities to check for repeated abnormalities in the case of results of A/D conversion which will cause malfunctions of the system.

There is a possibility that A/D conversion accuracy of high priority SG become worse when following both conditions are applicable.

(1) During the A/D conversion of low priority SG (e.g. SG1), conversion trigger of the high priority SG (e.g. SG3) occur.

(2) The channel T&H function of high priority SG (e.g. SG3) is enabled.

The above case is one of an example of SG combination. The SG priority is SG4 > SG3 > SG2 > SG1. The fluctuation of conversion error depends on the external circuit and devices mounted on the customer board.

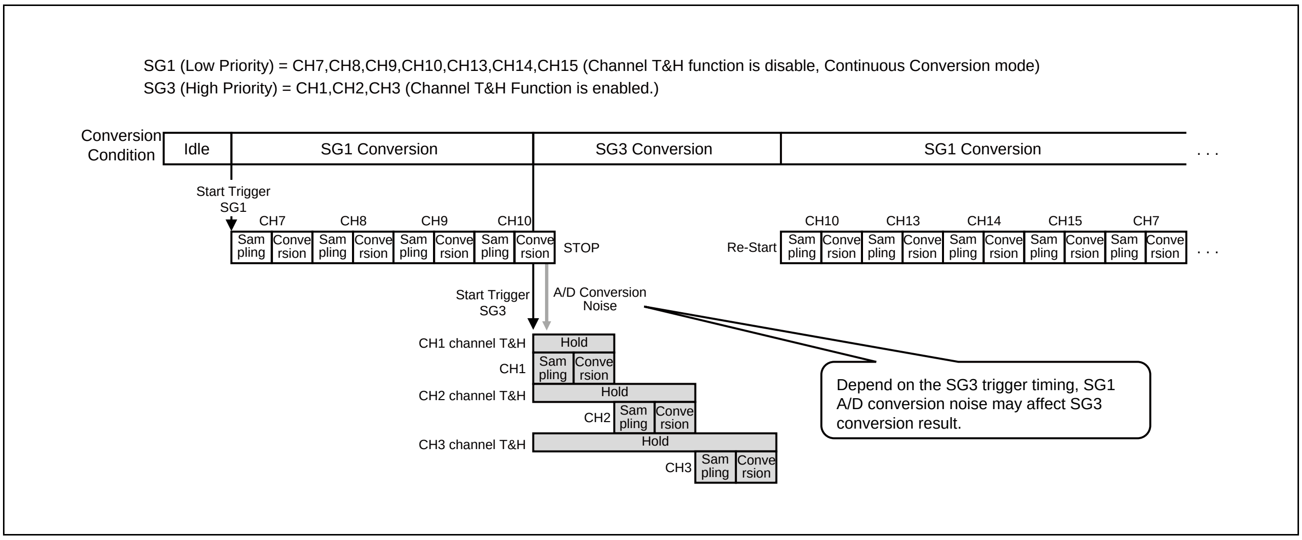


Figure 38.42 SG priority

Examples of software handling are described below.

- Low priority SG (e.g. SG1) A/D conversion have to be finished 3 ADCLK before the conversion trigger of high priority SG (e.g. SG3), if the SG3 contains the channels which channel T&H is available.
- If there is a case that high priority SG (e.g. SG3) conversion trigger occur during the conversion of low priority SG (e.g. SG1), disable the high priority SG channel T&H function.
- If the both conditions mentioned in previous page need to be used, adjust high priority SG (e.g. SG3) conversion trigger timing to synchronize with the following timing (the period shown by arrowed line in the following figure) during A/D conversion of low priority SG (e.g. SG1).

Even if trigger timing is adjusted above recommendation time, conversion error specified in **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1** cannot be removed.

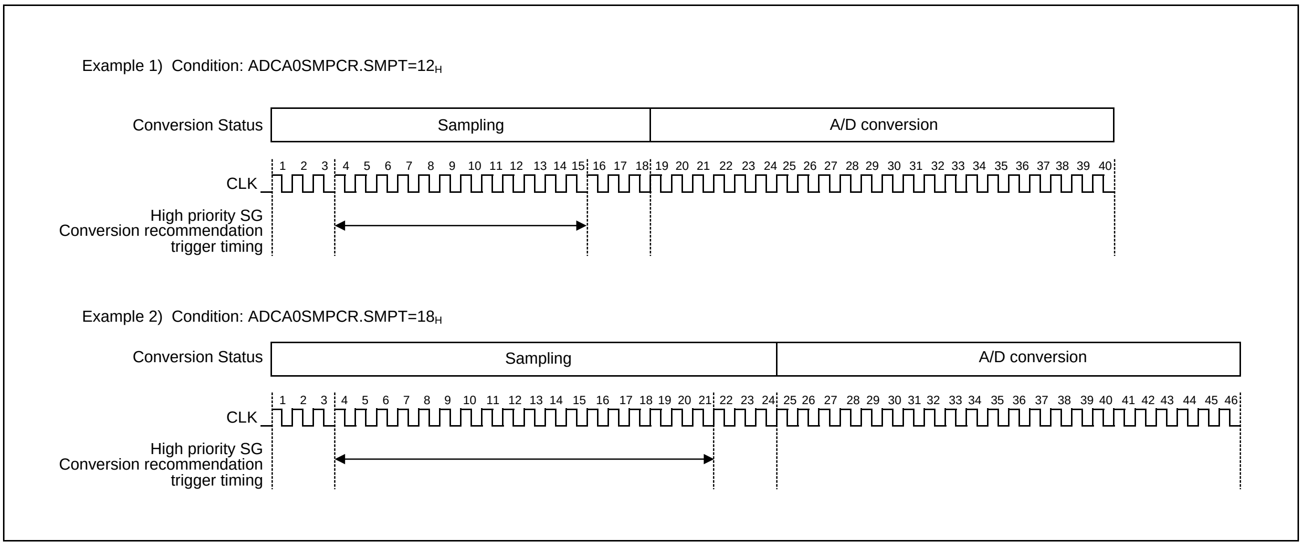


Figure 38.43 SG conversion trigger timing

When it is not possible to use the above 3 software handlings, following process is recommended.

- Doing A/D conversion several times and using average of several A/D conversion results.
- Doing continuous A/D conversion several times, remove abnormal conversion result and use only the other results.
- When abnormal A/D conversion result is detected, not to proceed abnormal operation immediately, doing one more A/D conversion before proceeding abnormal operation.

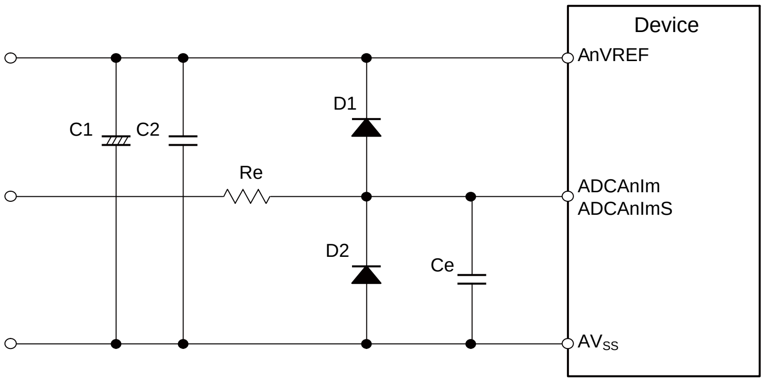
The effect of above process is depend on the external circuit and devices mounted on the customer board. Sufficient evaluation of the system is recommended.

(4) Alternative Input/Output

Analog input (ADCAnIm, ADCAnImS) pins can be used as port pins.

Do not read from input port pins or write to output port pins while an ADCAnIm or ADCAnImS pin function is selected and handling A/D conversion. Doing so may reduce the accuracy of conversion.

Fluctuations in output current from output port pins due to the effects of an external circuit connected to a port pin while A/D conversion is in progress may also reduce the accuracy of conversion. If digital pulses are applied to or digital pulses are output through a pin adjacent to a pin for which A/D conversion is in progress, the A/D converted value may not be as expected due to coupling noise. Accordingly, do not apply pulses to or output pulses from a pin adjacent to a pin for which A/D conversion is in progress.

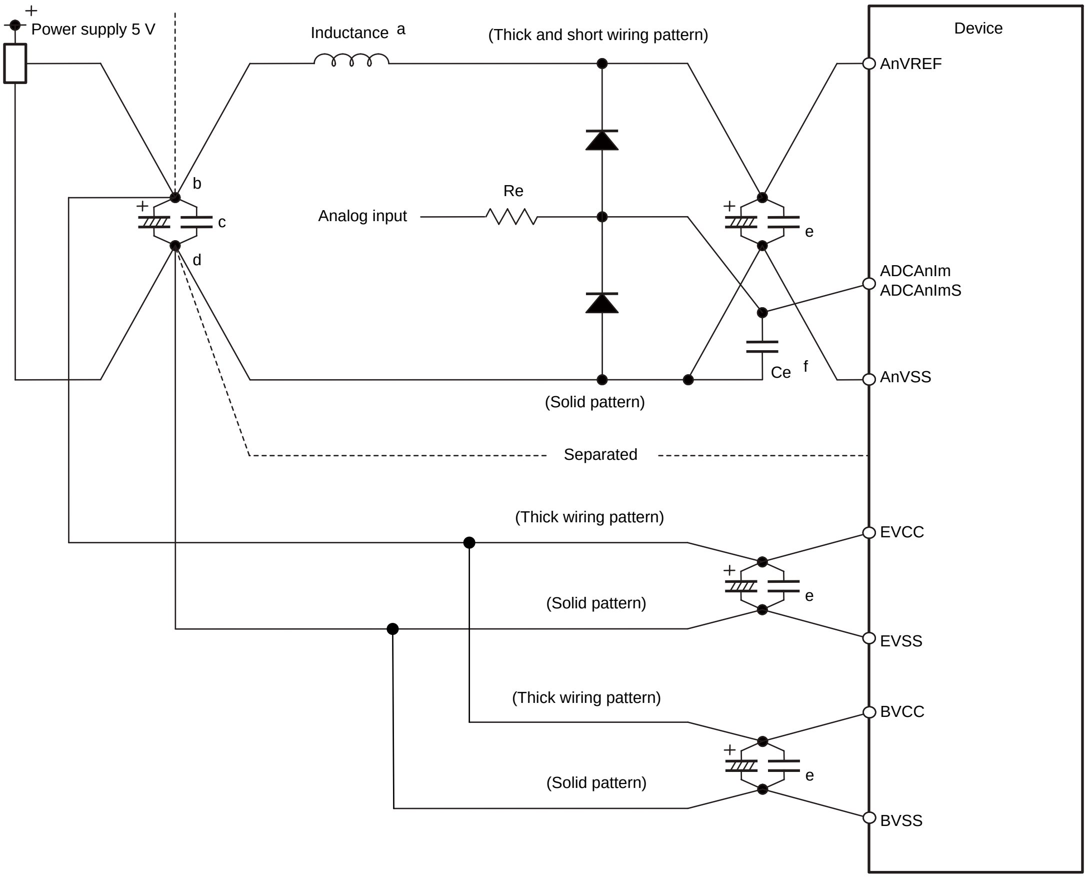


Note: C1: 4.7 µF to 10 µF
C2: 0.01 µF to 0.1 µF
Ce: 5 to 10 nF
Re: 0 to 1 kΩ
The values above are for reference.
D1, D2: VF ≤ 0.3 V

Figure 38.44 Example of Noise Protection for Analog Input Circuit

Capacitor C1 is effective for low-frequency noise, and capacitors C2 and Ce are effective for high-frequency noise.

The voltage on an AnVREF pin is undefined immediately after switching from the stopped state to the start of conversion operations, and this may have the effect of reducing the accuracy of conversion. As a countermeasure for this situation, connect capacitors C1 and C2 to the AnVREF pins.



a) Insert an inductor for the analog power input.

b) Wire the analog and digital supplies to a single point near the 5V power supply and provide separate wiring patterns on the board.

c) Place a bypass capacitor (electrolytic or laminated ceramic) near the common power supply.

d) Wire the analog and digital grounds to a single point near the power supplies, and provide separate wiring patterns for each on the board.

e) Place bypass capacitors (electrolytic or laminated ceramic) near the respective power supply pins.

f) Place capacitor Ce near the analog input pins of the device.

Figure 38.45 Example of Power Wiring

Section 39 Key Return (KR)

This section contains a generic description of the Key Return (KR).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of KR.

39.1 Features of RH850/F1KH, RH850/F1KM KR

39.1.1 Number of Units and Channels

This microcontroller has the following number of KR units and channels.

Table 39.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)

Table 39.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)

Table 39.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	1	1
Name	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)	KRn (n = 0)

Table 39.4 KRn Unit Configurations and Channels (RH850/F1KH-D8)

Unit Name KRn	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
KR0	8 ch	8 ch	8 ch

Table 39.5 KRn Unit Configurations and Channels (RH850/F1KM-S4)

Unit Name KRn	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
KR0	8 ch	8 ch	8 ch	8 ch	8 ch

Table 39.6 KRn Unit Configurations and Channels (RH850/F1KM-S1)

Unit Name KRn	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
KR0	6 ch	8 ch	8 ch	8 ch

Table 39.7 Indices (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual KR units are identified by the index “n”; for example, KRnKRM (n = 0) indicates the key return mode register.
m	Throughout this section, the individual KR channels are identified by the index “m”; for example, KRnKRMm (m = 0 to 7) indicates the key input enable bit of KRnKRM (key return mode register).

39.1.2 Register Base Address

KRn base address is listed in the following table.

KRn register addresses are given as an offset from the base address.

Table 39.8 Register Base Address (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<KR0_base>	FFF7 8000_H

39.1.3 Clock Supply

The KRn clock supply is shown in the following table.

Table 39.9 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
KRn	PCLK	CPUCLK_UL	Module clock
	Register access clock	CPUCLK_UL	Bus clock

39.1.4 Interrupt Requests

KRn interrupt requests are listed in the following table:

Table 39.10 Interrupt Requests (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
KR0			
INTKRn	Key interrupt	90	—

39.1.5 Reset Sources

KRn reset sources are listed in the following table. KRn is initialized by these reset sources.

Table 39.11 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
KRn	All reset sources (ISORES)

39.1.6 External Input/Output Signals

External input/output signals of KRn are listed below.

Table 39.12 External Input/Output Signals (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Signal Name	Description	Alternative Port Pin Signal
KR0		
KRnTPKR7 to KRnTPKR0	Key input signal	KR0I7 to KR0I0

39.2 Overview

39.2.1 Functional Overview

The Key Return function has the following features:

A key interrupt request signal (INTKRn) can be generated by inputting a falling signal, that goes from high to low, to any of the eight key input pins (KRnTPKR7 to KRnTPKR0).

39.2.2 Block Diagram

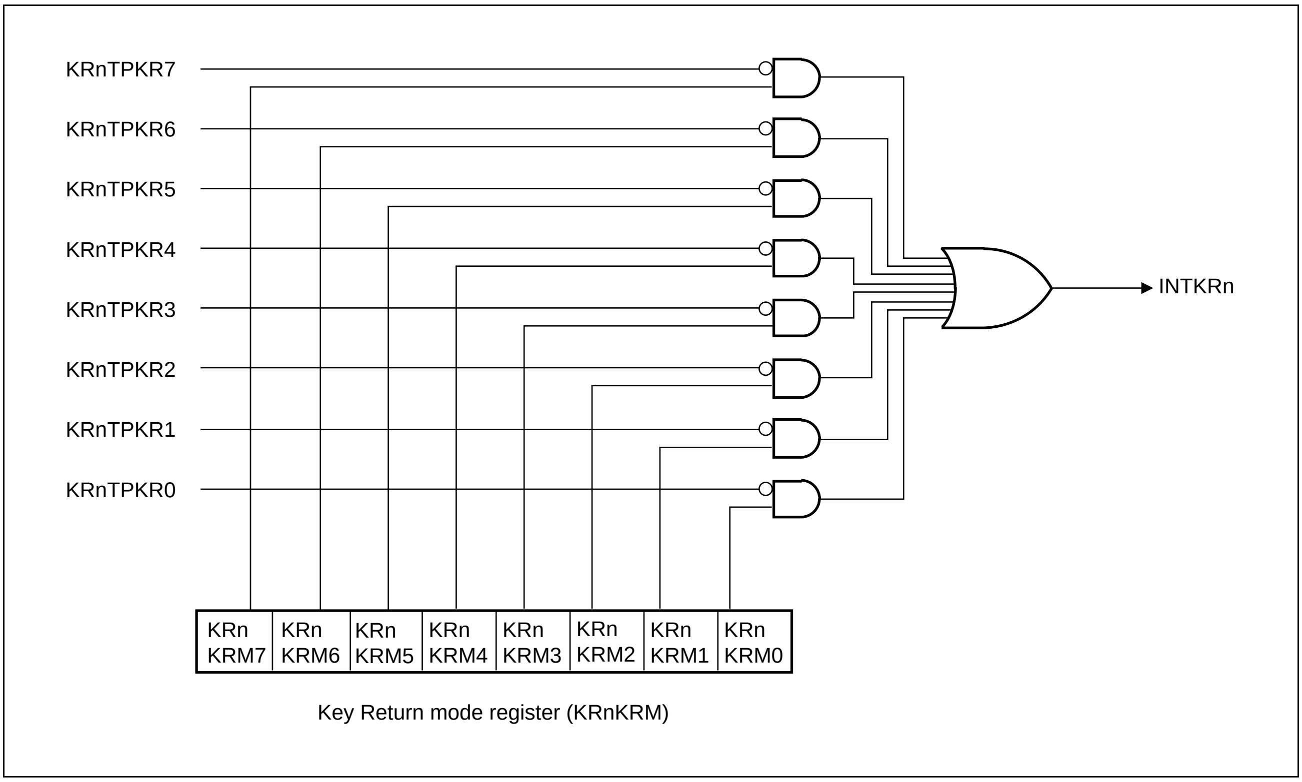


Figure 39.1 Block Diagram of the Key Return Function

39.3 Registers

39.3.1 List of Registers

KR register is listed in the following table.

For details about <KRn_base>, see **Section 39.1.2, Register Base Address**.

Table 39.13 List of Register

Module Name	Register Name	Symbol	Address
KRn	Key return mode register	KRnKRM	<KRn_base>

39.3.2 KRnKRM — Key Return Mode Register

This register enables/disables the key input signal detection.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <KRn_base>

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	KRnKRM7	KRnKRM6	KRnKRM5	KRnKRM4	KRnKRM3	KRnKRM2	KRnKRM1	KRnKRM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.14 KRnKRM Register Contents

Bit Position	Bit Name	Function
7 to 0	KRnKRMm	Enables/disables the key input signal detection. 0: Disabled 1: Enabled

39.4 Operation

39.4.1 Interrupt Request INTKRn

The interrupt request INTKRn is generated when the level of the corresponding key input pin (KRnTPKRm) is changed from high to low while input to the key input pin (KRnTPKRm) is enabled (KRnKRM.KRnKRMm = 1).

Figure 39.2, Interrupt Request Generation shows how the interrupt request is generated:

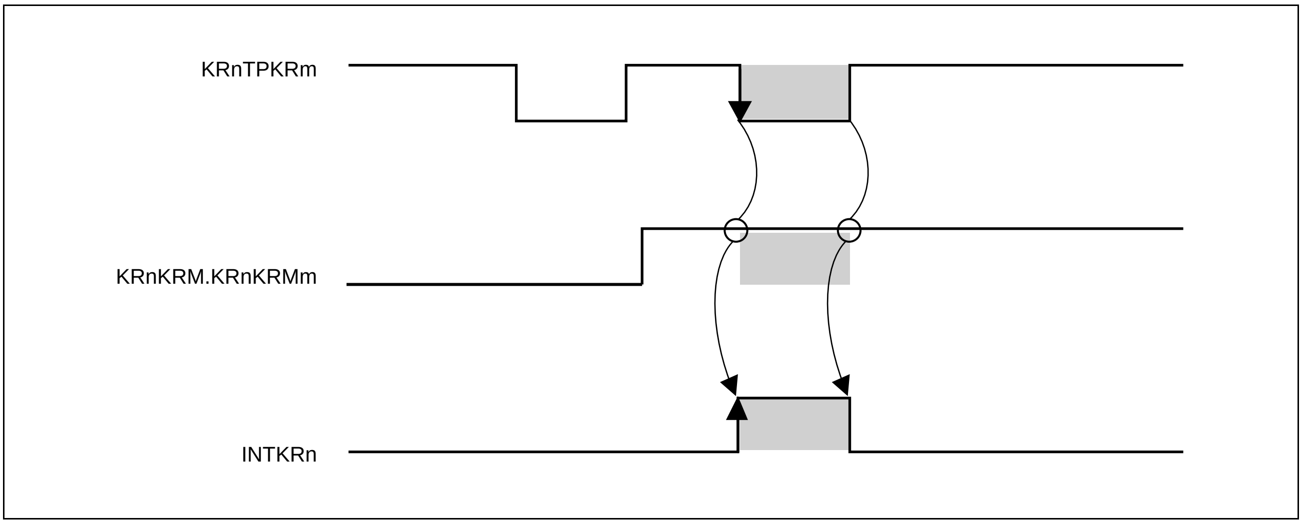


Figure 39.2 Interrupt Request Generation

CAUTIONS

1. The change of a key input pin (KRnTPKRm) level from high to low does not trigger another INTKRn if any of the key return input pins are already low. The next INTKRn is only triggered by a key input pin level changing from high to low if all other key input pins are high.
2. If the key input value changes at the same time the setting of KRnKRM.KRnKRMm is changed, an unintended key interrupt request INTKRn might be generated.
   Therefore, mask (disable) INTKRn of the interrupt controller before changing KRnKRM.KRnKRMm from 0 to 1, or from 1 to 0.

Section 40A Functional Safety of RH850/F1KH-D8

This section provides an overview of the safety mechanisms included in the RH850/F1KH Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance with ISO26262.

For more information about the development process and safety mechanisms, please contact our sales office.

The following are the failure detection functions provided by this microcontroller.

40A.1 Overview

ECC

Detects failures of memories and data transfer paths and corrects some types of failures.

Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements from erroneous access.

Clock Monitor

Monitors the clock operation to detect abnormal operations.

For details, see **Section 13, Clock Monitor (CLMA)**.

Data CRC

Generates CRC to detect data errors.

For details, see **Section 41, Data CRC (DCRA)**.

Write-Protected Registers

The write-protected registers are protected from inadvertent write access due to erroneous program execution.

For details, see **Section 5, Write-Protected Registers**.

40A.2 ECC

40A.2.1 Overview

This product incorporates an ECC for the following memories. The ECC enables detection and correction of errors of the data retained in the memories. The ECC also enables detection and correction of errors produced between the ECC encoder and memories and between memories and ECC decoder.

Table 40A.1 ECC Overview

Applicable Memory		Applicable Data Width [bits]	Operation upon Error Detection					Failure Insertion
			Detection/ Correction	SYSERR	Interrupt Notice	Error Status	Address Capture	
Code flash		128	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Data flash		32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
Local RAM (CPU1/CPU2/Self)		32	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Global RAM Retention RAM		32	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Peripheral RAM*[1]	CSIH	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	RS-CANFD	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	FlexRay	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	Ethernet	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	MMCA	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible

Note 1. For details of ECC for each peripheral IP, see **Section 40A.2.6, ECC for Peripheral RAM**.

Note 2. For details, see **Section 40A.2.2.2, Interrupt Requests, Section 40A.2.4.2, Interrupt Requests**, and **Section 40A.2.5.2, Interrupt Requests**.

Applicable Data Width

This is the data width to be ECC encoded.

To write data with a smaller data width than shown, the following processing is required. ECC is also performed for a read in (1).

(1) Reading data to be ECC-encoded including data to be rewritten

(2) Replacing data to be rewritten

(3) Writing back data generated in (2)

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected.

SED-DED: 1-bit errors and 2-bit errors can only be detected.

SYSERR

SYSERR can be generated upon error detection.

Interrupt Notice

An interrupt can be generated upon error detection.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

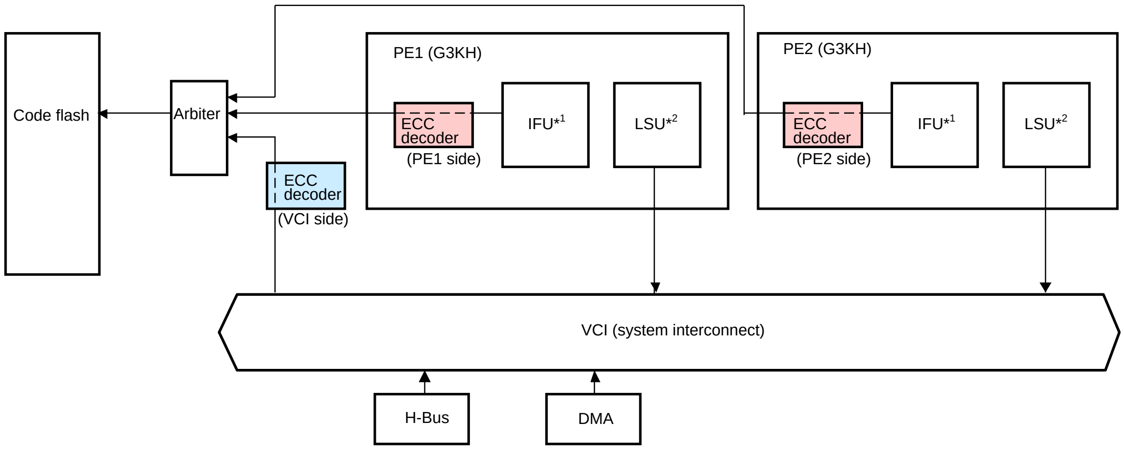
Self-diagnosis of the ECC decoder error notification function can be performed by using an intentionally generated ECC error.

40A.2.2 Code Flash ECC

40A.2.2.1 Overview

RH850/F1KH has three code flash ECC decoder circuits implemented, two inside the Processor Element (PE1/PE2) and one on the VCI (system interconnect).

Figure 40A.1, Block Diagram of Code Flash ECC shows the location of three ECC decoders for the code flash.



Note 1. Instruction Fetch Unit

Note 2. Load Store Unit

Figure 40A.1 Block Diagram of Code Flash ECC

The code flash ECC is summarized in the table below.

Table 40A.2 Summary of Code Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. (This is not supported in CFERRINT_PE1 register.) Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3A, CPU System of RH850/F1KH-D8**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.

Table 40A.2 Summary of Code Flash ECC

Item	Description
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The ECC 1-bit error status is set only when no error status has been set. The ECC 2-bit error status is set even when the ECC 1-bit error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error is also captured.
Self-diagnosis	The ECC bit can be read directly.
Inhibiting instruction execution	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching avoids the execution of incorrect instructions.

40A.2.2.2 Interrupt Requests

Interrupt requests for code flash ECC are listed below.

Table 40A.3 Code Flash ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0*[1] (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR	—

Note 1. ECCFLI0FEIF flag set can be read (not jump to handler address of FEINT) in SYSERR processing because return from SYSERR is not possible.

Table 40A.4 Code Flash ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0 (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR, INTECCFLI0	—

Table 40A.5 Code Flash ECC Interrupt Requests (During Bus Master Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	INTECCFLI0	—
—	ECC 2-bit error interrupt of code flash	INTECCFLI0	—

40A.2.2.3 List of Registers

Table 40A.6 List of Registers

Module Name	Register Name	Symbol*[1]	Address
CFECC_VCI	Code flash ECC control register (VCI)	CFECCCTL_VCI	FFC6 2200_H
	Code flash error information control register (VCI)	CFERRINT_VCI	FFC6 2204_H
	Code flash status clear register (VCI)	CFSTCLR_VCI	FFC6 2208_H
	Code flash error count overflow status register (VCI)	CFOVFSTR_VCI	FFC6 $220C_H$
	Code flash 1st error status register (VCI)	CF1STERSTR_VCI	FFC6 2210_H
	Code flash 1st error address register (VCI)	CF1STEADR0_VCI	FFC6 2250_H
CFECC_CPU1	Code flash ECC control register (PE1)	CFECCCTL_PE1	FFC6 2400_H
	Code flash error information control register (PE1)	CFERRINT_PE1	FFC6 2404_H
	Code flash status clear register (PE1)	CFSTCLR_PE1	FFC6 2408_H
	Code flash error count overflow status register (PE1)	CFOVFSTR_PE1	FFC6 $240C_H$
	Code flash 1st error status register (PE1)	CF1STERSTR_PE1	FFC6 2410_H
	Code flash 1st error address register (PE1)	CF1STEADR0_PE1	FFC6 2450_H
CFECC_CPU2	Code flash ECC control register (PE2)	CFECCCTL_PE2	FFC6 2600_H
	Code flash error information control register (PE2)	CFERRINT_PE2	FFC6 2604_H
	Code flash status clear register (PE2)	CFSTCLR_PE2	FFC6 2608_H
	Code flash error count overflow status register (PE2)	CFOVFSTR_PE2	FFC6 $260C_H$
	Code flash 1st error status register (PE2)	CF1STERSTR_PE2	FFC6 2610_H
	Code flash 1st error address register (PE2)	CF1STEADR0_PE2	FFC6 2650_H
CFECC_VCI	Code flash sub-test control register (VCI)	CFSTSTCTL_VCI	FFC6 2350_H

Note 1. The registers suffixed with symbols “_VCI”, “_PE1” and “_PE2” are provided to ECC controllers corresponding to each access port: registers with “_VCI” are provided to the ECC controller for data access from the system interconnect to the code flash, registers with “_PE1” are provided to the ECC controller for fetch access from the CPU1 and registers with “_PE2” are provided to the ECC controller for fetch access from the CPU2.

40A.2.2.4 Details of Registers

(1) CFECCCTL_VCI/PE1/PE2 — Code Flash ECC Control Register

CFECCCTL_VCI/PE1/PE2 enables or disables ECC error detection and correction and 1-bit error correction. When writing to CFECCCTL_VCI/PE1/PE2, PROT1 and PROT0 need to be 01_B.

Access: CFECCCTL_VCI, CFECCCTL_PE1 and CFECCCTL_PE2 can be read or written in 32-bit units.
CFECCCTL_VCIL, CFECCCTL_PE1L and CFECCCTL_PE2L can be read or written in 16-bit units.

Address: CFECCCTL_VCI: FFC6 2200$_H$
CFECCCTL_VCIL: FFC6 2200$_H$
CFECCCTL_PE1: FFC6 2400$_H$
CFECCCTL_PE1L: FFC6 2400$_H$
CFECCCTL_PE2: FFC6 2600$_H$
CFECCCTL_PE2L: FFC6 2600$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.7 CFECCCTL_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFECCCTL_VCI/PE1/PE2.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) CFERRINT_VCI/PE1/PE2 — Code Flash Error Information Control Register

CFERRINT_VCI/PE1/PE2 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access:	CFERRINT_VCI, CFERRINT_PE1 and CFERRINT_PE2 can be read or written in 32-bit units. CFERRINT_VCIL, CFERRINT_PE1L and CFERRINT_PE2L can be read or written in 16-bit units. CFERRINT_VCILL, CFERRINT_PE1LL and CFERRINT_PE2LL can be read or written in 8-bit units.
Address:	CFERRINT_VCI: FFC6 2204$_H$ CFERRINT_VCIL: FFC6 2204$_H$ CFERRINT_VCILL: FFC6 2204$_H$ CFERRINT_PE1: FFC6 2404$_H$ CFERRINT_PE1L: FFC6 2404$_H$ CFERRINT_PE1LL: FFC6 2404$_H$ CFERRINT_PE2: FFC6 2604$_H$ CFERRINT_PE2L: FFC6 2604$_H$ CFERRINT_PE2LL: FFC6 2604$_H$
Value after reset:	0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE *1	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.8 CFERRINT_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE*1	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

Note 1. This bit is not supported in CFERRINT_PE1/PE2. When writing to this bit in CFERRINT_PE1/PE2, always write 1.

Note 2. Regarding the interrupt request, see **Section 40A.2.2.2, Interrupt Requests**.

(3) CFSTCLR_VCI/PE1/PE2 — Code Flash Status Clear Register

CFSTCLR_VCI/PE1/PE2 clears the error flags in the error status register (CF1STERSTR_VCI/PE1/PE2), the overflow flag in the error overflow status register (CFOVFSTR_VCI/PE1/PE2), and the error address register (CF1STEADR0_VCI/PE1/PE2).

Access: CFSTCLR_VCI, CFSTCLR_PE1 and CFSTCLR_PE2 are write-only registers that can be written in 32-bit units.
CFSTCLR_VCIL, CFSTCLR_PE1L and CFSTCLR_PE2L are write-only registers that can be written in 16-bit units.
CFSTCLR_VCILL, CFSTCLR_PE1LL and CFSTCLR_PE2LL are write-only registers that can be written in 8-bit units.

Address: CFSTCLR_VCI: FFC6 2208$_H$
CFSTCLR_VCIL: FFC6 2208$_H$
CFSTCLR_VCILL: FFC6 2208$_H$
CFSTCLR_PE1: FFC6 2408$_H$
CFSTCLR_PE1L: FFC6 2408$_H$
CFSTCLR_PE1LL: FFC6 2408$_H$
CFSTCLR_PE2: FFC6 2608$_H$
CFSTCLR_PE2L: FFC6 2608$_H$
CFSTCLR_PE2LL: FFC6 2608$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40A.9 CFSTCLR_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Clear 0: No effect (Setting 0 does not affect the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1/PE2; ERROVF0 flag in CFOVFSTR_VCI/PE1/PE2; and CF1STEADR0_VCI/PE1/PE2.) 1: Writing 1 to this bit clears the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1/PE2; ERROVF0 flag in CFOVFSTR_VCI/PE1/PE2; and CF1STEADR0_VCI/PE1/PE2.

(4) CFOVFSTR_VCI/PE1/PE2 — Code Flash Error Count Overflow Status Register

CFOVFSTR_VCI/PE1/PE2 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

Access: CFOVFSTR_VCI, CFOVFSTR_PE1 and CFOVFSTR_PE2 are read-only registers that can be read in 32-bit units.
CFOVFSTR_VCIL, CFOVFSTR_PE1L and CFOVFSTR_PE2L are read-only registers that can be read in 16-bit units.
CFOVFSTR_VCILL, CFOVFSTR_PE1LL and CFOVFSTR_PE2LL are read-only registers that can be read in 8-bit units.

Address: CFOVFSTR_VCI: FFC6 220C$_H$
CFOVFSTR_VCIL: FFC6 220C$_H$
CFOVFSTR_VCILL: FFC6 220C$_H$
CFOVFSTR_PE1: FFC6 240C$_H$
CFOVFSTR_PE1L: FFC6 240C$_H$
CFOVFSTR_PE1LL: FFC6 240C$_H$
CFOVFSTR_PE2: FFC6 260C$_H$
CFOVFSTR_PE2L: FFC6 260C$_H$
CFOVFSTR_PE2LL: FFC6 260C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.10 CFOVFSTR_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 shows whether a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register has occurred, except when both of the error address and source of the second error are the same as those of the first error. 0: Did not occur. 1: Occurred. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1/PE2 to 1.

(5) CF1STERSTR_VCI/PE1/PE2 — Code Flash 1st Error Status Register

CF1STERSTR_VCI/PE1/PE2 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0. If a 2-bit ECC error occurs while the 1-bit ECC error flag is set, the 2-bit ECC error flag is set while retaining the 1-bit ECC error flag.

Access:	CF1STERSTR_VCI, CF1STERSTR_PE1 and CF1STERSTR_PE2 are read-only registers that can be read in 32-bit units. CF1STERSTR_VCIL, CF1STERSTR_PE1L and CF1STERSTR_PE2L are read-only registers that can be read in 16-bit units. CF1STERSTR_VCILL, CF1STERSTR_PE1LL and CF1STERSTR_PE2LL are read-only registers that can be read in 8-bit units.
Address:	CF1STERSTR_VCI: FFC6 2210$_H$ CF1STERSTR_VCIL: FFC6 2210$_H$ CF1STERSTR_VCILL: FFC6 2210$_H$ CF1STERSTR_PE1: FFC6 2410$_H$ CF1STERSTR_PE1L: FFC6 2410$_H$ CF1STERSTR_PE1LL: FFC6 2410$_H$ CF1STERSTR_PE2: FFC6 2610$_H$ CF1STERSTR_PE2L: FFC6 2610$_H$ CF1STERSTR_PE2LL: FFC6 2610$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.11 CF1STERSTR_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1/PE2 to 1. [Setting condition] ECC 2-bit error is detected when DEDF0 is 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected when DEDF0 flag is 0. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1/PE2 to 1. [Setting condition] ECC 1-bit error is detected when DEDF0, SEDF0 are 0.

(6) CF1STEADR0_VCI/PE1/PE2 — Code Flash 1st Error Address Register

CF1STEADR0_VCI/PE1/PE2 holds the address at which an error has occurred.

The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1/PE2. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated.

In addition, the EADR[24:4] bits in this register correspond to the [24:4] bits of the real address. The real address can be calculated by adding the upper address [31:25] bits, to which code flash is mapped, as a base address.

The CF1STEADR0_VCI/PE1/PE2 register is cleared by an internal reset, the external reset, or by setting the STCLR bit in the CFSTCLR_VCI/PE1/PE2 register to 1.

Access: CF1STEADR0_VCI, CF1STEADR0_PE1 and CF1STEADR0_PE2 are read-only registers that can be read in 32-bit units.

CF1STEADR0_VCIL, CF1STEADR0_VCIH, CF1STEADR0_PE1L, CF1STEADR0_PE1H, CF1STEADR0_PE2L and CF1STEADR0_PE2H are read-only registers that can be read in 16-bit units.

CF1STEADR0_VCILL, CF1STEADR0_VCILH, CF1STEADR0_VCIHL, CF1STEADR0_VCIHH, CF1STEADR0_PE1LL, CF1STEADR0_PE1LH, CF1STEADR0_PE1HL, CF1STEADR0_PE1HH, CF1STEADR0_PE2LL, CF1STEADR0_PE2LH, CF1STEADR0_PE2HL and CF1STEADR0_PE2HH are read-only registers that can be read in 8-bit units.

Address:
CF1STEADR0_VCI: FFC6 2250$_H$
CF1STEADR0_VCIL: FFC6 2250$_H$
CF1STEADR0_VCIH: FFC6 2252$_H$
CF1STEADR0_VCILL: FFC6 2250$_H$
CF1STEADR0_VCILH: FFC6 2251$_H$
CF1STEADR0_VCIHL: FFC6 2252$_H$
CF1STEADR0_VCIHH: FFC6 2253$_H$
CF1STEADR0_PE1: FFC6 2450$_H$
CF1STEADR0_PE1L: FFC6 2450$_H$
CF1STEADR0_PE1H: FFC6 2452$_H$
CF1STEADR0_PE1LL: FFC6 2450$_H$
CF1STEADR0_PE1LH: FFC6 2451$_H$
CF1STEADR0_PE1HL: FFC6 2452$_H$
CF1STEADR0_PE1HH: FFC6 2453$_H$
CF1STEADR0_PE2: FFC6 2650$_H$
CF1STEADR0_PE2L: FFC6 2650$_H$
CF1STEADR0_PE2H: FFC6 2652$_H$
CF1STEADR0_PE2LL: FFC6 2650$_H$
CF1STEADR0_PE2LH: FFC6 2651$_H$
CF1STEADR0_PE2HL: FFC6 2652$_H$
CF1STEADR0_PE2HH: FFC6 2653$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	EADR[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.12 CF1STEADR0_VCI/PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	EADR[24:4]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1/PE2. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1/PE2 to 1.
3 to 0	Reserved	When read, the value after reset is returned.

(7) CFSTSTCTL_VCI — Code Flash Sub-Test Control Register

CFSTSTCTL_VCI is used for the ECC test (self-diagnosis). This register is dedicated for code flash. After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read directly.

When writing to CFSTSTCTL_VCI, PROT1 and PROT0 need to be 01_B.

Access:	CFSTSTCTL_VCI can be read or written in 32-bit units. CFSTSTCTL_VCIL can be read or written in 16-bit units.
Address:	CFSTSTCTL_VCI: FFC6 2350_H CFSTSTCTL_VCIL: FFC6 2350_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40A.13 CFSTSTCTL_VCI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFSTSTCTL_VCI.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, ECC bits can be read directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3A, CPU System of RH850/F1KH-D8**.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n address. The results of reading code flash are as follows:

Table 40A.14 Results of Reading Code Flash

Bit Number	Meaning	Bit Position	Description
bit[31:10]	all-0	31 to 10	These bits are always 0.
bit[9]	reserved	9	Unknown
bit[8:0]	ECC bits	8 to 0	ECC bits

40A.2.3 Data Flash ECC

40A.2.3.1 Overview

The data flash ECC is summarized in the table below.

Table 40A.15 Summary of Data Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection / correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit errors is selectable. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit error is selectable. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the Interrupt Controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. ECC error address is captured when both SEDF and DEDF of DFERSTR are 0.
Self-diagnosis	ROM data and the ECC bit can be read directly. Arbitrary data can be written to ROM data and the ECC bit.

40A.2.3.2 Interrupt Requests

The interrupt requests for data flash ECC are shown below.

Table 40A.16 Data Flash ECC Interrupt Requests (During Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of data flash	INTECCEEP0	—
—	ECC 2-bit error interrupt of data flash	INTECCEEP0	—

40A.2.3.3 List of Registers

Table 40A.17 List of Registers

Module Name	Register Name	Symbol	Address
DFECC	Data flash ECC control register	DFECCCTL	FFC6 2A00$_H$
	Data flash error status register	DFERSTR	FFC6 2A04$_H$
	Data flash error status clear register	DFERSTC	FFC6 2A08$_H$
	Data flash error overflow status register	DFOVFSTR	FFC6 2A0C$_H$
	Data flash error overflow status clear register	DFOVFSTC	FFC6 2A10$_H$
	Data flash error notification control register	DFERRINT	FFC6 2A14$_H$
	Data flash 1st error address register	DFEADR	FFC6 2A18$_H$
	Data flash test control register	DFTSTCTL	FFC6 2A1C$_H$

40A.2.3.4 Details of Registers

(1) DFECCCTL — Data Flash ECC Control Register

DFECCCTL enables or disables ECC error detection and correction and 1-bit error correction. When writing to DFECCCTL, PROT1 and PROT0 need to be 01_B.

Access: DFECCCTL can be read or written in 16-bit units.

Address: DFECCCTL: FFC6 $2A00_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.18 DFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCCTL.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection and correction are enabled, and the DEDF bit is set if an ECC 2-bit error is detected.

Access:	DFERSTR is a read-only register that can be read in 8-bit units.
Address:	DFERSTR: FFC6 $2A04_H$
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40A.19 DFERSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 2-bit error is detected when both SEDF and DEDF are 0.
0	SEDF	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 1-bit error is detected when both SEDF and DEDF are 0.

(3) DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the data flash error status register.

Access: DFERSTC is a write-only register that can be written in 8-bit units.

Address: DFERSTC: FFC6 2A08$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERRCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40A.20 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

(4) DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of data flash error overflow.

Access: DFOVFSTR is a read-only register that can be read in 8-bit units.

Address: DFOVFSTR: FFC6 2A0C$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40A.21 DFOVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag ERROVF is set if an ECC error occurs while the error address register is full. 0: Did not occurred. 1: Occurred. [Clearing condition] Set the ERROVFCLR bit is set in data flash error overflow status clear register.

(5) DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the data flash error overflow flag.

Access: DFOVFSTC is a write-only register that can be written in 8-bit units.

Address: DFOVFSTC: FFC6 2A10$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVFCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40A.22 DFOVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the ERROVF flag in DFOVFSTR.) 1: The ERROVF flag in DFOVFSTR is cleared.

(6) DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: DFERRINT can be read or written in 8-bit units.

Address: DFERRINT: FFC6 2A14$_H$

Value after reset: 02$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 40A.23 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(7) DFEADR — Data Flash 1st Error Address Register

DFEADR holds the address at which an ECC error has occurred while both of the SEDF and DEDF bits in the data flash error status register are 0.

Access: DFEADR is a read-only register that can be read in 32-bit units.

Address: DFEADR: FFC6 2A18$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	DFEADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DFEADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.24 DFEADR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 2	DFEADR[20:2]	ECC Error Address DFEADR is read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address FF20 0000$_H$.
1, 0	Reserved	When read, the value after reset is returned.

(8) DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for the ECC test.

After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read.

When writing to DFTSTCTL, PROT1 and PROT0 need to be 01_B.

Access: DFTSTCTL can be read or written in 16-bit units.

Address: DFTSTCTL: FFC6 $2A1C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40A.25 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFTSTCTL.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Sets ECC test mode. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

40A.2.4 Local RAM ECC

40A.2.4.1 Overview

The local RAM ECC of CPU1/CPU2 is summarized in the table below.

Table 40A.26 Summary of Local RAM ECC (CPU1/CPU2)

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3A, CPU System of RH850/F1KH-D8**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no error status has been set, the address at which the first error occurred is captured. In addition, when the retained address source is an ECC 1-bit error or ECC 2-bit error, the address is also captured.
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data and the ECC bit can be read directly.
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

CAUTION

When ECC error detection/correction for the local RAM is enabled for access, initialize the RAM with the 32-bit length of RAM access before the RAM is used. If the RAM before initialization is read, an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

Moreover, if the RAM is not initialized with the 32-bit length (for example, initialized with 8- or 16-bit length of access), an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

40A.2.4.2 Interrupt Requests

The local RAM ECC interrupt requests are listed below.

Table 40A.27 Local RAM ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40A.28 Local RAM ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40A.29 Local RAM ECC Interrupt Requests (During Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	INTECCRAM	—

40A.2.4.3 List of Registers

Table 40A.30 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
LRTST	FFC6 5004$_H$	LRTSTCTL_PE1	Local RAM test control register (PE1)	R/W	0000 0000$_H$	16/32
	FFC6 5008$_H$	LRTDATBF0_PE1	Local RAM test data read buffer 0 (PE1)	R	0000 0000$_H$	32
	FFC6 5024$_H$	LRTSTCTL_PE2	Local RAM test control register (PE2)	R/W	0000 0000$_H$	16/32
	FFC6 5028$_H$	LRTDATBF0_PE2	Local RAM test data read buffer 0 (PE2)	R	0000 0000$_H$	32
LRECC	FFC6 5400$_H$	LRECCCTL_PE1	Local RAM ECC control register (PE1)	R/W	0000 0000$_H$	16/32
	FFC6 5404$_H$	LRERRINT_PE1	Local RAM error information control register (PE1)	R/W	0000 0003$_H$	8/16/32
	FFC6 5408$_H$	LRSTCLR_PE1	Local RAM status clear register (PE1)	W	0000 0000$_H$	8/16/32
	FFC6 540C$_H$	LROVFSTR_PE1	Local RAM error count overflow status register (PE1)	R	0000 0000$_H$	8/16/32
	FFC6 5410$_H$	LR1STERSTR_PE1	Local RAM 1st error status register (PE1)	R	0000 0000$_H$	8/16/32
	FFC6 5450$_H$	LR1STEADR0_PE1	Local RAM 1st error address register 0 (PE1)	R	0000 0000$_H$	8/16/32
	FFC6 5600$_H$	LRECCCTL_PE2	Local RAM ECC control register (PE2)	R/W	0000 0000$_H$	16/32
	FFC6 5604$_H$	LRERRINT_PE2	Local RAM error information control register (PE2)	R/W	0000 0003$_H$	8/16/32
	FFC6 5608$_H$	LRSTCLR_PE2	Local RAM status clear register (PE2)	W	0000 0000$_H$	8/16/32
	FFC6 560C$_H$	LROVFSTR_PE2	Local RAM error count overflow status register (PE2)	R	0000 0000$_H$	8/16/32
	FFC6 5610$_H$	LR1STERSTR_PE2	Local RAM 1st error status register (PE2)	R	0000 0000$_H$	8/16/32
	FFC6 5650$_H$	LR1STEADR0_PE2	Local RAM 1st error address register 0 (PE2)	R	0000 0000$_H$	8/16/32

40A.2.4.4 Details of Registers

(1) LRTSTCTL_PE1/PE2 — Local RAM Test Control Register

LRTSTCTL_PE1/PE2 is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST = 1, any data can be written to the ECC bits. The DATSEL bit is used to select RAM data or the ECC bits.

When writing to LRTSTCTL_PE1/PE2, PROT1 and PROT0 need to be 01$_B$.

Access:	LRTSTCTL_PE1 and LRTSTCTL_PE2 registers can be read or written in 32-bit units. LRTSTCTL_PE1L and LRTSTCTL_PE2L registers can be read or written in 16-bit units.
Address:	LRTSTCTL_PE1: FFC6 5004$_H$ LRTSTCTL_PE1L: FFC6 5004$_H$ LRTSTCTL_PE2: FFC6 5024$_H$ LRTSTCTL_PE2L: FFC6 5024$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST	DATSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.31 LRTSTCTL_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST and DATSEL bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRTSTCTL_PE1/PE2.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read by reading LRTDATBF0_PE1/PE2 register or written directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).
0	DATSEL	Data Select This bit is valid when ECCTST = 1. This bit selects the RAM bit which can be accessed when writing. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: RAM data is selected. 1: The ECC bits are selected.

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

(2) LRTDATBF0_PE1/PE2 — Local RAM Test Data Read Buffer 0

In ECC test mode (self-diagnosis), the ECC bits can be read. If the local RAM is read while ECCTST = 1 in the local RAM test control register LRTSTCTL_PE1/PE2, reading from the local RAM reads out the ECC bits, and these bits are stored in this buffer.

Access:	LRTDATBF0_PE1 and LRTDATBF0_PE2 registers are read-only registers that can be read in 32-bit units.
Address:	LRTDATBF0_PE1: FFC6 5008$_H$ LRTDATBF0_PE2: FFC6 5028$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	LRTDATBF[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.32 LRTDATBF0_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	LRTDATBF[6:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the local RAM, the ECC bits for the local RAM are stored in LRTDATBF[6:0].

(3) LRECCCTL_PE1/PE2 — Local RAM ECC Control Register

LRECCCTL_PE1/PE2 enables or disables ECC error detection and correction and 1-bit error correction.

When writing to LRECCCTL_PE1/PE2, PROT1 and PROT0 need to be 01_B.

Access: LRECCCTL_PE1 and LRECCCTL_PE2 registers can be read or written in 32-bit units.
LRECCCTL_PE1L and LRECCCTL_PE2L registers can be read or written in 16-bit units.

Address: LRECCCTL_PE1: FFC6 5400_H
LRECCCTL_PE1L: FFC6 5400_H
LRECCCTL_PE2: FFC6 5600_H
LRECCCTL_PE2L: FFC6 5600_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.33 LRECCCTL_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRECCCTL_PE1/PE2.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(4) LRERRINT_PE1/PE2 — Local RAM Error Information Control Register

LRERRINT_PE1/PE2 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: LRERRINT_PE1 and LRERRINT_PE2 registers can be read or written in 32-bit units.
LRERRINT_PE1L and LRERRINT_PE2L registers can be read or written in 16-bit units.
LRERRINT_PE1LL and LRERRINT_PE2LL registers can be read or written in 8-bit units.

Address: LRERRINT_PE1: FFC6 5404$_H$
LRERRINT_PE1L: FFC6 5404$_H$
LRERRINT_PE1LL: FFC6 5404$_H$
LRERRINT_PE2: FFC6 5604$_H$
LRERRINT_PE2L: FFC6 5604$_H$
LRERRINT_PE2LL: FFC6 5604$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.34 LRERRINT_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(5) LRSTCLR_PE1/PE2 — Local RAM Status Clear Register

LRSTCLR_PE1/PE2 clears the error flags in the error status register (LR1STERSTR_PE1/PE2), the overflow flag in the error overflow status register (LROVFSTR_PE1/PE2), and the error address register (LR1STEADR0_PE1/PE2). LRSTCLR_PE1/PE2 is a write-only register and is always read as 0.

Access: LRSTCLR_PE1 and LRSTCLR_PE2 registers are write-only registers that can be written in 32-bit units.
LRSTCLR_PE1L and LRSTCLR_PE2L registers are write-only registers that can be written in 16-bit units.
LRSTCLR_PE1LL and LRSTCLR_PE2LL registers are write-only registers that can be written in 8-bit units.

Address: LRSTCLR_PE1: FFC6 5408$_{\text{H}}$
LRSTCLR_PE1L: FFC6 5408$_{\text{H}}$
LRSTCLR_PE1LL: FFC6 5408$_{\text{H}}$
LRSTCLR_PE2: FFC6 5608$_{\text{H}}$
LRSTCLR_PE2L: FFC6 5608$_{\text{H}}$
LRSTCLR_PE2LL: FFC6 5608$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40A.35 LRSTCLR_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF0 and SEDF0 flags in LR1STERSTR_PE1/PE2; ERROVF0 flag in LROVFSTR_PE1/PE2; and LR1STEADR0_PE1/PE2.

(6) LROVFSTR_PE1/PE2 — Local RAM Error Count Overflow Status Register

LROVFSTR_PE1/PE2 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF0 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1/PE2.

Access: LROVFSTR_PE1 and LROVFSTR_PE2 registers are read-only registers that can be read in 32-bit units.
LROVFSTR_PE1L and LROVFSTR_PE2L registers are read-only registers that can be read in 16-bit units.
LROVFSTR_PE1LL and LROVFSTR_PE2LL registers are read-only registers that can be read in 8-bit units.

Address: LROVFSTR_PE1: FFC6 540C$_H$
LROVFSTR_PE1L: FFC6 540C$_H$
LROVFSTR_PE1LL: FFC6 540C$_H$
LROVFSTR_PE2: FFC6 560C$_H$
LROVFSTR_PE2L: FFC6 560C$_H$
LROVFSTR_PE2LL: FFC6 560C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.36 LROVFSTR_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

(7) LR1STERSTR_PE1/PE2 — Local RAM 1st Error Status Register

LR1STERSTR_PE1/PE2 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0.

If more than one error occurs simultaneously, all the corresponding error flags are set. LR1STERSTR_PE1/PE2 is cleared by an internal reset, the external reset, or setting 1 to the STCLR0 bit in LRSTCLR_PE1/PE2.

Access:	LR1STERSTR_PE1 and LR1STERSTR_PE2 registers are read-only registers that can be read in 32-bit units. LR1STERSTR_PE1L and LR1STERSTR_PE2L registers are read-only registers that can be read in 16-bit units. LR1STERSTR_PE1LL and LR1STERSTR_PE2LL registers are read-only registers that can be read in 8-bit units.
Address:	LR1STERSTR_PE1: FFC6 5410$_{\text{H}}$ LR1STERSTR_PE1L: FFC6 5410$_{\text{H}}$ LR1STERSTR_PE1LL: FFC6 5410$_{\text{H}}$ LR1STERSTR_PE2: FFC6 5610$_{\text{H}}$ LR1STERSTR_PE2L: FFC6 5610$_{\text{H}}$ LR1STERSTR_PE2LL: FFC6 5610$_{\text{H}}$
Value after reset:	0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.37 LR1STERSTR_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1/PE2 to 1. [Setting condition] ECC 2-bit error is detected with DEDF0 being 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1/PE2 to 1. [Setting condition] ECC 1-bit error is detected with both SEDF0 and DEDF0 being 0.

(8) LR1STEADR0_PE1/PE2 — Local RAM 1st Error Address Register 0

LR1STEADR0_PE1/PE2 holds the address at which an error has occurred.

The error address is set if an error occurs while error flags are 0 in LR1STERSTR_PE1/PE2. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set in LR1STERSTR_PE1/PE2. Once an ECC 2-bit error occurs, the address is not updated.

Since this register holds the internal address, add the base address[31:19] bits of the associated memory to transform the internal address to the real address. LR1STEADR0_PE1/PE2 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1/PE2.

Access: LR1STEADR0_PE1 and LR1STEADR0_PE2 registers are read-only registers that can be read in 32-bit units.
LR1STEADR0_PE1L, LR1STEADR0_PE1H, LR1STEADR0_PE2L and LR1STEADR0_PE2H registers are read-only registers that can be read in 16-bit units.
LR1STEADR0_PE1LL, LR1STEADR0_PE1LH, LR1STEADR0_PE1HL, LR1STEADR0_PE2LL, LR1STEADR0_PE2LH and LR1STEADR0_PE2HL registers are read-only registers that can be read in 8-bit units.

Address:
LR1STEADR0_PE1: FFC6 5450$_H$
LR1STEADR0_PE1L: FFC6 5450$_H$,
LR1STEADR0_PE1H: FFC6 5452$_H$
LR1STEADR0_PE1LL: FFC6 5450$_H$,
LR1STEADR0_PE1LH: FFC6 5451$_H$,
LR1STEADR0_PE1HL: FFC6 5452$_H$
LR1STEADR0_PE2: FFC6 5650$_H$
LR1STEADR0_PE2L: FFC6 5650$_H$,
LR1STEADR0_PE2H: FFC6 5652$_H$
LR1STEADR0_PE2LL: FFC6 5650$_H$,
LR1STEADR0_PE2LH: FFC6 5651$_H$,
LR1STEADR0_PE2HL: FFC6 5652$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	EADR[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.38 LR1STEADR0_PE1/PE2 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 2	EADR[18:2]	1st Error Address Monitors the address of the first error. The error address is held if an error occurs while all the error flags are 0 in LR1STERSTR_PE1/PE2. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.
1, 0	Reserved	When read, the value after reset is returned.

40A.2.5 Global RAM (Including the Retention RAM) ECC

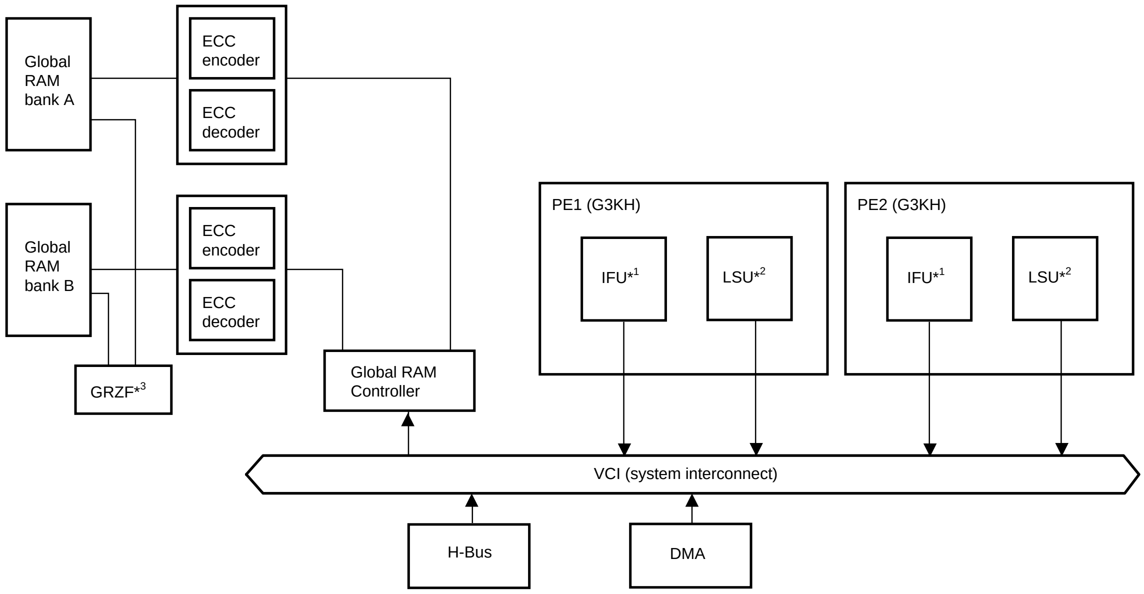
40A.2.5.1 Overview

CAUTION

The retention RAM is a part of the global RAM. The ECC for the retention RAM is shared with the global RAM's. Therefore, use the same register as the global RAM's in case of the retention RAM.

In RH850/F1KH, an ECC encoder and an ECC decoder are provided for each bank (bank A and bank B).

Figure 40A.2, Block Diagram of Global RAM ECC shows the location of two ECC decoders and two ECC encoders for the global RAM.



Note 1. Instruction Fetch Unit

Note 2. Load Store Unit

Note 3. GRZF accesses RAM directly. For details, see **Section 45.3, Global RAM Zero Fill (GRZF)**.

Figure 40A.2 Block Diagram of Global RAM ECC

The global RAM ECC is summarized in the table below.

Table 40A.39 Summary of Global RAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction ● 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit error is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3A, CPU System of RH850/F1KH-D8**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.
Error status	A status register is provided to indicate the statuses of 2-bit ECC error detection and 1-bit ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. In addition, if the address source retained is 1-bit ECC error, it is captured when a 2-bit ECC error is detected
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data can be read directly
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

40A.2.5.2 Interrupt Requests

Global RAM ECC interrupt requests are listed below.

Table 40A.40 Global RAM ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40A.41 Global RAM ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40A.42 Global RAM ECC Interrupt Requests (During Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	INTECCRAM	—

40A.2.5.3 List of Registers

Table 40A.43 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
GRECC	FFC6 4000_H	GRECCCTL_BKA	Global RAM ECC control register (Bank A)	R/W	0000 0000_H	16/32
	FFC6 4004_H	GRERRINT_BKA	Global RAM error information control register (Bank A)	R/W	0000 0001_H	8/16/32
	FFC6 4010_H	GRSTCLR_BKA	Global RAM status clear register (Bank A)	W	0000 0000_H	8/16/32
	FFC6 4014_H	GROVFSTR_BKA	Global RAM error count overflow status register (Bank A)	R	0000 0000_H	8/16/32
	FFC6 4018_H	GR1STERSTR_BKA	Global RAM 1st error status register (Bank A)	R	0000 0000_H	8/16/32
	FFC6 $401C_H$	GR1STEADR_BKA	Global RAM 1st error address register (Bank A)	R	FEE0 0000_H	8/16/32
	FFC6 4020_H	GRTSTCTL_BKA	Global RAM test control register (Bank A)	R/W	0000 0000_H	16/32
	FFC6 4024_H	GRDECINBF1_BKA	Global RAM ECC decoder input data buffer 1 (Bank A)	R/W	0000 0000_H	8/16/32
	FFC6 4200_H	GRECCCTL_BKB	Global RAM ECC control register (Bank B)	R/W	0000 0000_H	16/32
	FFC6 4204_H	GRERRINT_BKB	Global RAM error information control register (Bank B)	R/W	0000 0001_H	8/16/32
	FFC6 4210_H	GRSTCLR_BKB	Global RAM status clear register (Bank B)	W	0000 0000_H	8/16/32
	FFC6 4214_H	GROVFSTR_BKB	Global RAM error count overflow status register (Bank B)	R	0000 0000_H	8/16/32
	FFC6 4218_H	GR1STERSTR_BKB	Global RAM 1st error status register (Bank B)	R	0000 0000_H	8/16/32
	FFC6 $421C_H$	GR1STEADR_BKB	Global RAM 1st error address register (Bank B)	R	FEF0 0000_H	8/16/32
	FFC6 4220_H	GRTSTCTL_BKB	Global RAM test control register (Bank B)	R/W	0000 0000_H	16/32
	FFC6 4224_H	GRDECINBF1_BKB	Global RAM ECC decoder input data buffer 1 (Bank B)	R/W	0000 0000_H	8/16/32

Note: _BKA and _BKB in Symbol indicate on each bank. “BKA” represents for bank A and “BKB” represents for bank B.

40A.2.5.4 Details of Registers

(1) GRECCCTL_BKA/BKB — Global RAM ECC Control Register

GRECCCTL_BKA/BKB enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to GRECCCTL_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access:	GRECCCTL_BKA and GRECCCTL_BKB can be read or written in 32-bit units GRECCCTL_BKAL and GRECCCTL_BKBL can be read or written in 16-bit units
Address:	GRECCCTL_BKA: FFC6 4000_H GRECCCTL_BKAL: FFC6 4000_H GRECCCTL_BKB: FFC6 4200_H GRECCCTL_BKBL: FFC6 4200_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40A.44 GRECCCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0.Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. When writing, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. When writing, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) GRERRINT_BKA/BKB— Global RAM Error Information Control Register

GRERRINT_BKA/BKB enables or disables generation of the error notification signal upon detection of an ECC 1-bit error.

The setting of this register is used for accesses through the respective access port.

Access: GRERRINT_BKA and GRERRINT_BKB can be read or written in 32-bit units
GRERRINT_BKAL and GRERRINT_BKBL can be read or written in 16-bit units
GRERRINT_BKALL and GRERRINT_BKBLL can be read or written in 8-bit units

Address: GRERRINT_BKA: FFC6 4004$_H$
GRERRINT_BKAL: FFC6 4004$_H$
GRERRINT_BKALL: FFC6 4004$_H$
GRERRINT_BKB: FFC6 4204$_H$
GRERRINT_BKBL: FFC6 4204$_H$
GRERRINT_BKBLL: FFC6 4204$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40A.45 GRERRINT_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(3) GRSTCLR_BKA/BKB — Global RAM Status Clear Register

GRSTCLR_BKA/BKB clears the error flags in the error status register (GR1STERSTR_BKA/BKB), the overflow flag in the error count overflow status register (GROVFSTR_BKA/BKB), and the error address register (GR1STEADR_BKA/BKB). GRSTCLR_BKA/BKB is a write-only register and is always read as 0.

The setting of this register is used for accesses through the respective access port.

Access: GRSTCLR_BKA and GRSTCLR_BKB are write-only registers that can be written in 32-bit units.
GRSTCLR_BKAL and GRSTCLR_BKBL are write-only registers that can be written in 16-bit units.
GRSTCLR_BKALL and GRSTCLR_BKBLL are write-only registers that can be written in 8-bit units

Address: GRSTCLR_BKA: FFC6 4010$_H$
GRSTCLR_BKAL: FFC6 4010$_H$
GRSTCLR_BKALL: FFC6 4010$_H$
GRSTCLR_BKB: FFC6 4210$_H$
GRSTCLR_BKBL: FFC6 4210$_H$
GRSTCLR_BKBLL: FFC6 4210$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40A.46 GRSTCLR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, always write 0.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF and SEDF flags in GR1STERSTR_BKA/BKB, ERROVF0 flag in GROVFSTR_BKA/BKB, and EADR[19:0] in GR1STEADR_BKA/BKB.

(4) GROVFSTR_BKA/BKB— Global RAM Error Count Overflow Status Register

GROVFSTR_BKA/BKB monitors occurrence of error overflow. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

The setting of this register is used for accesses through the respective access port.

Access: GROVFSTR_BKA and GROVFSTR_BKB are read-only registers that can be read in 32-bit units
GROVFSTR_BKAL and GROVFSTR_BKBL are read-only registers that can be read in 16-bit units
GROVFSTR_BKALL and GROVFSTR_BKBLL are read-only registers that can be read in 8-bit units

Address: GROVFSTR_BKA: FFC6 4014$_H$
GROVFSTR_BKAL: FFC6 4014$_H$
GROVFSTR_BKALL: FFC6 4014$_H$
GROVFSTR_BKB: FFC6 4214$_H$
GROVFSTR_BKBL: FFC6 4214$_H$
GROVFSTR_BKBLL: FFC6 4214$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.47 GROVFSTR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF and SEDF) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error.

(5) GR1STERSTR_BKA/BKB— Global RAM 1st Error Status Register

GR1STERSTR_BKA/BKB monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. However, the corresponding error flag is set if an ECC 2-bit error occurs only when SEDF bit is set. GR1STERSTR_BKA/BKB is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in GRSTCLR_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access: GR1STERSTR_BKA and GR1STERSTR_BKB are read-only registers that can be read in 32-bit units.
GR1STERSTR_BKAL and GR1STERSTR_BKBL are read-only registers that can be read in 16-bit units.
GR1STERSTR_BKALL and GR1STERSTR_BKBLL are read-only registers that can be read in 8-bit units.

Address: GR1STERSTR_BKA: FFC6 4018_H
GR1STERSTR_BKAL: FFC6 4018_H
GR1STERSTR_BKALL: FFC6 4018_H
GR1STERSTR_BKB: FFC6 4218_H
GR1STERSTR_BKBL: FFC6 4218_H
GR1STERSTR_BKBLL: FFC6 4218_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.48 GR1STERSTR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in GRSTCLR_BKA/BKB to 1. [Setting condition] ECC 2-bit error is detected when DEDF is 0.
0	SEDF	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the STCLR0 bit in GRSTCLR_BKA/BKB to 1. [Setting condition] ECC 1-bit error is detected when DEDF and SEDF are 0.

(6) GR1STEADR_BKA/BKB— Global RAM 1st Error Address Register

GR1STEADR_BKA/BKB holds the address at which an error has occurred.
The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR_BKA/BKB. The address

is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error occurs, the address is not updated.
GR1STEADR_BKA/BKB is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in GRSTCLR_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access: GR1STEADR_BKA and GR1STEADR_BKB are read-only registers that can be read in 32-bit units.
GR1STEADR_BKAL, GR1STEADR_BKBL, GR1STEADR_BKAH and GR1STEADR_BKBH are read-only registers that can be read in 16-bit units.
GR1STEADR_BKALL, GR1STEADR_BKALH, GR1STEADR_BKAHL, GR1STEADR_BKBLL, GR1STEADR_BKBLH and GR1STEADR_BKBHL are read-only registers that can be read in 8-bit unit.

Address:
GR1STEADR_BKA: FFC6 401C$_H$
GR1STEADR_BKAL: FFC6 401C$_H$
GR1STEADR_BKALL: FFC6 401C$_H$
GR1STEADR_BKALH: FFC6 401D$_H$
GR1STEADR_BKAH: FFC6 401E$_H$
GR1STEADR_BKAHL: FFC6 401E$_H$
GR1STEADR_BKB: FFC6 421C$_H$
GR1STEADR_BKBL: FFC6 421C$_H$
GR1STEADR_BKBLL: FFC6 421C$_H$
GR1STEADR_BKBLH: FFC6 421D$_H$
GR1STEADR_BKBH: FFC6 421E$_H$
GR1STEADR_BKBHL: FFC6 421E$_H$

Value after reset: GR1STEADR_BKA: FEE0 0000$_H$, GR1STEADR_BKB: FEF0 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	EADR[19:16]			
Value after reset	1	1	1	1	1	1	1	0	1	1	1	0/1[*1]	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.49 GR1STEADR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	EADR[19:0]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR_BKA/BKB. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Note 1. The value after reset differs depending on bank A or bank B.

(7) GRTSTCTL_BKA/BKB — Global RAM Test Control Register

This register is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST to 1, any data can be written to the ECC bits.
Also, input and output by the ECC decoder in the global RAM controller can be controlled for testing (self-diagnosis).

Set PROT1 to 0 and PROT0 to 1 when writing to this register.

Access: GRTSTCTL_BKA and GRTSTCTL_BKB can be read or written in 32-bit units
GRTSTCTL_BKAL and GRTSTCTL_BKBL can be read or written in 16-bit units

Address: GRTSTCTL_BKA: FFC6 4020_H
GRTSTCTL_BKAL: FFC6 4020_H
GRTSTCTL_BKB: FFC6 4220_H
GRTSTCTL_BKBL: FFC6 4220_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40A.50 GRTSTCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15	PROT1	Enables or disables modification of the ECCTST. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to GRTSTCTL_BKA/BKB.
14	PROT0	
13 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ECCTST	ECC test bit. This bit permits to switch the ECC data for writing into RAM. Assert (PROT1, PROT0) = (0, 1) to write to this register. 0: Output of ECC encoder is used for ECC data to RAM 1: Register output is used for ECC data to RAM.

(8) GRDECINBF1_BKA/BKB— Global RAM ECC Decoder Input Data Buffer 1

7-bit ECC of global RAM is replaced to this register value by write access to global RAM during GRTSTCTL_BKA/BKB.ECCTST = 1 (test mode).

Access:	GRDECINBF1_BKA and GRDECINBF1_BKB can be read or written in 32-bit units. GRDECINBF1_BKAL and GRDECINBF1_BKBL can be read or written in 16-bit units. GRDECINBF1_BKALL and GRDECINBF1_BKBLL can be read or written in 8-bit units.
Address:	GRDECINBF1_BKA: FFC6 4024_H GRDECINBF1_BKAL: FFC6 4024_H GRDECINBF1_BKALL: FFC6 4024_H GRDECINBF1_BKB: FFC6 4224_H GRDECINBF1_BKBL: FFC6 4224_H GRDECINBF1_BKBLL: FFC6 4224_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	GRDECINBF1[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40A.51 GRDECINBF1_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, these bits are always read as 0. When writing, always write 0.
6 to 0	GRDECINBF1[6:0]	This field is valid ECCTST = 1 (test mode). When the write operation is executed, the data in this register is used for writing to 7-bit ECC field RAM.

40A.2.6 ECC for Peripheral RAM

40A.2.6.1 Overview

This is an ECC module for the RAM of the following peripheral modules.

MMCA, CSIH, RS-CANFD, FlexRay and Ethernet AVB.

Table 40A.52 List of the ECC Functions for the peripheral RAM

Item	Description
ECC error detection/correction	ECC error detection and correction can be enabled or disabled. Either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction ● 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. In the initial state, 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Error notification can be enabled or disabled when an ECC 2-bit error is detected. ● Error notification can be enabled or disabled when an ECC 1-bit error is detected. In the initial state of ECC controller, 2-bit error notification is enabled and 1-bit error notification is disabled. However, if an interrupt is masked by the FEINTFMSK register, an interrupt processing is not executed.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC1-bit errors is available. A bit for clearing the error status is provided.
Address capture	Only one address at which an ECC error has occurred can be captured. A signal is generated upon detection of ECC 2-bit or 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed about RS-CANFD or FlexRay, initialize the RAM area before it is used.

40A.2.6.2 Interrupt Requests

ECC of peripheral interrupt requests are listed below.

Table 40A.53 MMCA ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	MMCA0 ECC 1-bit error or 2-bit error interrupt	INTECCMMCA0RAM	—

Table 40A.54 CSIHn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
INTECCCSIHn	CSIHn ECC 1-bit error or 2-bit error interrupt	INTECCCSIH0	—
		INTECCCSIH1	
		INTECCCSIH2	
		INTECCCSIH3	
		INTECCCSIH4	

Table 40A.55 RCFDCn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
ECCCNFDRAMFEIF	RCFDCn ECC 1-bit error or 2-bit error interrupt	INTECCCNFDRAM0	—
		INTECCCNFDRAM1	

Table 40A.56 FlexRay ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	FLXA0 ECC 1-bit error or 2-bit error interrupt	INTECCFLRAM	—

Table 40A.57 Ethernet AVB ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ETNBn ECC 1-bit error or 2-bit error interrupt	INTECCETH0	—
		INTECCETH1	

40A.2.6.3 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 40A.58 List of ECC Modules

Peripheral Functions		Module Name	ECC Module Names and Register	
			Base Address Name	Base Address <base_addr>
MMCA0	Transfer buffer A (TBFA)	ECCMMCA0A	<ECCMMCA0A_base>	FFED A000$_H$
	Transfer buffer B (TBFB)	ECCMMCA0B	<ECCMMCA0B_base>	FFED A100$_H$
CSIHn	Buffered I/O (CSIH RAM)	ECCCSIH0	<ECCCSIH0_base>	FFC7 0100$_H$
		ECCCSIH1	<ECCCSIH1_base>	FFC7 0200$_H$
		ECCCSIH2	<ECCCSIH2_base>	FFC7 0300$_H$
		ECCCSIH3	<ECCCSIH3_base>	FFC7 0400$_H$
		ECCCSIH4	<ECCCSIH4_base>	FFC7 0500$_H$
RCFDC0	Message bufferRAM (MB RAM)	ECCCFD0MB	<ECCCFD0MB_base>	FFC7 1300$_H$
	Acceptance filter list RAM0 (AFL0 RAM)	ECCCFD0AFL0	<ECCCFD0AFL0_base>	FFC7 1400$_H$
	Acceptance filter list RAM1 (AFL1 RAM)	ECCCFD0AFL1	<ECCCFD0AFL1_base>	FFC7 1500$_H$
RCFDC1	Message bufferRAM (MB RAM)	ECCCFD1MB	<ECCCFD1MB_base>	FFC7 1A00$_H$
	Acceptance filter list RAM0 (AFL0 RAM)	ECCCFD1AFL0	<ECCCFD1AFL0_base>	FFC7 1B00$_H$
	Acceptance filter list RAM1 (AFL1 RAM)	ECCCFD1AFL1	<ECCCFD1AFL1_base>	FFC7 1C00$_H$
FLXA0	Message RAM (MRAM)	ECCFLXA0	<ECCFLXA0_base>	FFC7 3100$_H$
	Temporary buffer (TBFA)	ECCFLXA0T0	<ECCFLXA0T0_base>	FFC7 3200$_H$
	Temporary buffer (TBFB)	ECCFLXA0T1	<ECCFLXA0T1_base>	FFC7 3300$_H$
ETNB0	Transmit FIFO RAM.(TXRAM)	ECCETNB0TX	<ECCETNB0TX_base>	FFC7 4100$_H$
	Receive FIFO RAM.(RXRAM)	ECCETNB0RX	<ECCETNB0RX_base>	FFC7 4200$_H$
ETNB1	Transmit FIFO RAM.(TXRAM)	ECCETNB1TX	<ECCETNB1TX_base>	FFC7 4300$_H$
	Receive FIFO RAM.(RXRAM)	ECCETNB1RX	<ECCETNB1RX_base>	FFC7 4400$_H$

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 40A.59 List of Registers

Module Name	Register Name	Symbol	Address
See (1) List of ECC Modules.	ECC control register	<Module_Name>CTL	<base_addr> + 00_H
	ECC test mode control register	<Module_Name>TMC	<base_addr> + 04_H
	ECC encode/decode input/output replacement test register	<Module_Name>TED	<base_addr> + $0C_H$
	ECC redundant bit data control test register	<Module_Name>TRC	<base_addr> + 08_H
	ECC decode syndrome data register	<Module_Name>SYND	<base_addr> + $0B_H$
	ECC 7-bit redundant bit data hold test register	<Module_Name>HORD	<base_addr> + $0A_H$
	ECC encode test register	<Module_Name>ECRD	<base_addr> + 09_H
	ECC redundant bit input/output replacement register	<Module_Name>ERDB	<base_addr> + 08_H
	ECC error address register 0	<Module_Name>AD0	<base_addr> + 10_H
SL_READTEST	ECCREAD test select register	SELB_READTEST	FFC7 8000H

40A.2.6.4 Details of Registers

(1) <Module_Name>CTL — ECC Control Register

The <Module_Name>CTL register controls the mode of the ECC and the status for target peripheral modules.

Bits 7, 5, 4 and 3 should be set (written) while the target peripheral module’s operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: $001X_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	ECCOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 40A.60 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
15	EMCA1	Access Control Bits 1 and 0 to ECC Mode Selection These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ECCOVFF	By detecting an error while the error status is set and the new error has another address than the already latched (not cleared or reset is not issued), this bit is set. 0: Overflow is not occurred after reset of clearing ECER2F and ECER1F. 1: Error address register overflowed **NOTE:** This bit clear condition is as follows. (1) Reset (2) Writing ECER2C = 1 when ECER2F = 1 or ECER1C = 1 when ECER1F = 1 (3) Selecting through mode enable (ECTHM = 1)
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority. ECER2C = 1 also clears the ECCOVFF bit while ECER2F bit is set.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority. ECER1C = 1 also clears the ECCOVFF bit while ECER1F bit is set.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40A.60 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
7	ECTHM	ECC Function through Mode Selection Set this bit to select whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, write 01_B to the EMCA1 and EMCA0 bits at the same time. Set this bit to 1 to disable the ECC function. 0: Passing through mode is disabled (normal operation mode). 1: Passing through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 2-bit error is detected, a target peripheral module's interrupt will be generated.
3	EC1EDIC	1-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 1-bit error is detected, a target peripheral module's interrupt will be generated.
2	ECER2F	2-Bit Error Detection Flag This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (a target peripheral module's interrupt) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. When through mode is enable (ECTHM = 1), this bit is cleared. If 2-bit error is detected again while this bit is set, an interrupt will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
1	ECER1F	1-Bit Error Detection/Correction Flag This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. When through mode is enabled (ECTHM = 1), this bit is cleared. 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
0	ECEMF	ECC Error Message Flag This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. This bit is also cleared when through mode is enabled (ECTHM = 1) and there is no 1-bit error in decode circuit input data. 0: The current RAM output data does not have bit errors. 1: The current RAM output data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

(2) <Module_Name>TMC — ECC Test Mode Control Register

The <Module_Name>TMC register is used to switch to the test mode, and this register is for test mode control.

This register can be used when a target peripheral module is not accessed to RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*[1]	R/W*[1]	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 40A.61 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
15	ETMA1	Access Control Bits 1 and 0 to ECC Test Mode These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to test control bits of the test registers and this register. When writing to this bit, write 10_B to the ETMA1 and ETMA0 bits at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: <Module_Name>TED, <Module_Name>TRC, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, <Module_Name>ERDB Register test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the <Module_Name>TED and <Module_Name>ERDB registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the <Module_Name>TED register will be the write value of the <Module_Name>TED register. Read value of the <Module_Name>ERDB register will be the write value of the <Module_Name>ERDB register. 1: Read value of the <Module_Name>TED register can read RAM data. Read value of the <Module_Name>ERDB register will be the ECC Data to be written to RAM.
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies which is output to the ECC to be stored in RAM, the ECC data generated for write data or the value of the <Module_Name>ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated for write data is stored in RAM. 1: The value of <Module_Name>ERDB Register is stored in RAM.

Table 40A.61 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Selection This bit specifies data written to RAM or the value of the <Module_Name>TED register as the input to the ECC encoder. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the <Module_Name>TED.
1	ECDCS	ECC Decoder Input Selection This bit specifies which data is for generation of syndrome code and error detection, RAM data or the value of <Module_Name>TED. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from RAM Data. 1: Syndrome code generation and error detection are performed from <Module_Name>TED register value.
0	ECREIS	ECC Redundant Bit Input Data Selection This bit specifies which ECC data is for generation of syndrome code and error detection, ECC data stored in RAM or the value of the <Module_Name>ERDB. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from <Module_Name>ERDB register value.

(3) <Module_Name>TED — ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

This register value is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access: This register can be read or written in 32-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40A.62 <Module_Name>TED Register Contents

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When <Module_Name>TMC.ECENS = 1, the value of this register is used to generate ECC data and that is stored to RAM. When <Module_Name>TMC.ECDCS = 1, the value of this register is used to generate syndrome code that is stored in ECC decode syndrome data register (<Module_Name>SYND). In addition, when <Module_Name>TMC.ECTRRS = 1, RAM data [31:0] instead of written data is read for the value of this register.

(4) <Module_Name>TRC — ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and consists of four 8-bit registers, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, and <Module_Name>ERDB.

This register can be accessed when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access:	This register can be read or written in 32-bit units.
Address:	See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	<Module_Name>SYND (see **Section (6)**)								<Module_Name>HORD (see **Section (7)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<Module_Name>ECRD (see **Section (8)**)								<Module_Name>ERDB (see **Section (9)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(5) <Module_Name>AD0 — Target ECC Error Address Register 0

This is read only register to hold the ECC error occurred address.

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as trigger and it is hold as the error occurring address.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ECEAD[30:16]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.63 <Module_Name>AD0 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 0	ECEAD[30:0]	ECEAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECEAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECEAD0

(6) <Module_Name>SYND — ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register is read-only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40A.64 <Module_Name>SYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

(7) <Module_Name>HORD — ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40A.65 <Module_Name>HORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. When <Module_Name>TMC.ECTRRS =1 and if <Module_Name>TED register is read, ECC code is stored.

(8) <Module_Name>ECRD — ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40A.66 <Module_Name>ECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing and can read ECC data for data written in the <Module_Name>TED register when <Module_Name>MC.ECENS =1.

(9) <Module_Name>ERDB — ECC Redundant Bit Input/Output Replacement Register

In ECC test mode, this register handles test data.

This register value can be handled as generated ECC data at the time of writing to RAM or as read ECC data at the time of reading RAM data.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Access:	This register can be read or written in 8-bit units.
Address:	See **Table 40A.58, List of ECC Modules** and **Table 40A.59, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40A.67 <Module_Name>ERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	These bits can store this register value as ECC data when <Module_Name>TMC.ECREOS = 1. When the register is read while <Module_Name>TMC.ECREIS =1, the value read from these bits is ECC data read from the RAM. When <Module_Name>TMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instated of written data.

(10) SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the target peripheral module's RAM ECC registers.

Setting 1 to the bit corresponding to each function will enable writing to the read-only bit.

Access: SELB_READTESTW can be read or written in 32-bit units.
SELB_READTEST and SELB_READTESTH can be read or written in 16-bit units.

Address: SELB_READTESTW: FFC7 8000_H
SELB_READTEST: FFC7 8000_H
SELB_READTESTH: FFC7 8002_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RTMMC0BRAM	RTMMC0ARAM	RTCANFD1E7A03	RTCANFD1E7A02	RTCANFD1E7A01	RTETH1RXRAM	RTETH1TXRAM	RTCSIHE7A4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCANFD0E7A03	RTCANFD0E7A02	RTCANFD0E7A01	RTETH0RXRAM	RTETH0TXRAM	—	—	—	—	RTFLXAE7A0	RTFLXATRAM1	RTFLXATRAM0	RTCSIHE7A3	RTCSIHE7A2	RTCSIHE7A1	RTCSIHE7A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40A.68 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	RTMMC0BRAM	MMCA0 (TBFB) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (MMCA0 (TBFB) ECC read-only bit can be written).
22	RTMMC0ARAM	MMCA0 (TBFA) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (MMCA0 (TBFA) ECC read-only bit can be written).
21	RTCANFD1E7A03	RCFDC1 (AFL Buffer 1) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC1 (AFL Buffer 1) ECC read-only bit can be written).
20	RTCANFD1E7A02	RCFDC1 (AFL Buffer 0) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC1 (AFL Buffer 0) ECC read-only bit can be written).
19	RTCANFD1E7A01	RCFDC1 (Message Buffer) ECC Register Write Access for Testing Purpose Enable /Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC1 (Message Buffer) ECC read-only bit can be written).
18	RTETH1RXRAM	ETNB1 (RXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB1 (RXRAM) ECC read-only bit can be written).

Table 40A.68 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
17	RTETH1TXRAM	ETNB1 (TXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB1 (TXRAM) ECC read-only bit can be written).
16	RTCSIHE7A4	CSIH4 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH4 ECC read-only bit can be written).
15	RTCANFD0E7A03	RCFDC0 (AFL Buffer 1) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 (AFL Buffer 1) ECC read-only bit can be written).
14	RTCANFD0E7A02	RCFDC0 (AFL Buffer 0) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 (AFL Buffer 0) ECC read-only bit can be written).
13	RTCANFD0E7A01	RCFDC0 (Message Buffer) ECC Register Write Access for Testing Purpose Enable /Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 (Message Buffer) ECC read-only bit can be written).
12	RTETH0RXRAM	ETNB0 (RXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB0 (RXRAM) ECC read-only bit can be written).
11	RTETH0TXRAM	ETNB0 (TXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB0 (TXRAM) ECC read-only bit can be written).
10 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	RTFLXAE7A0	FLXA0 (MRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (MRAM) ECC read-only bit can be written).
5	RTFLXATRAM1	FLXA0 (TBFB) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (TBFB) ECC read-only bit can be written).
4	RTFLXATRAM0	FLXA0 (TBFA) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (TBFA) ECC read-only bit can be written).
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH3 ECC read-only bit can be written).
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH2 ECC read-only bit can be written).
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH0 ECC read-only bit can be written).

40A.3 Memory Protection

40A.3.1 Overview

This product incorporates the memory protection function to prevent erroneous accesses to data in memories and control registers of the peripheral circuits.

- MPU
  The CPU protects memory against illegal access by itself. The CPU does not generate the signals for access to addresses where access is prohibited by the MPU. For details, see *the RH850 Family User's Manual: Software*.
- Slave Guard
  A specific memory is protected against illegal accesses from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.
  - PEG
    The local RAM is protected against illegal accesses. However, accesses from the CPU incorporating the local RAM itself are excluded. For example, accesses from the CPU1 to local RAM in the CPU1 are not rejected by the PEG. For details, see **Section 3A, CPU System of RH850/F1KH-D8**.
  - IPG
    The CPU Peripheral is protected against illegal accesses.
    For details, see **Section 3A, CPU System of RH850/F1KH-D8**.
  - GRG
    The global RAM is protected against illegal accesses.
  - PBG/HBG
    The control registers in the peripheral circuits are protected against illegal accesses. For details, see **Section 40A.3.3, PBG/HBG**.
  - PBGC
    The CPU system has its dedicated PBG function which is called PBG for CPU system. For details, see **Section 40A.3.4, PBG for CPU System**.

40A.3.1.1 Identifiers for Slave Guard

For the slave guard function, the type of illegal accesses to be rejected can be designated using the following identifiers.

Table 40A.69 Identifiers for Slave Guard

Identifier	Function
UM	When the CPU makes an access, this indicates the operating mode of the CPU. 0: Supervisor mode 1: User mode When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 0.
SPID	When the CPU makes an access, this indicates the system protection identifier SPID that is assigned to the CPU. When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 00_B.
PEID	This indicates the access source bus master. 001_B: CPU1 010_B: CPU2 100_B: Other bus master (H-Bus bus master) When the PDMA makes an access, the value of this identifier is the value in the channel master setting register.

40A.3.2 GRG (Global RAM Guard)

This product is provided with 4-channel GRG, which is described in detail in the following sections.

CAUTION

The retention RAM is a part of the global RAM. The Guard for the retention RAM is shared with the global RAM's. Therefore, use the same register as the global RAM's in case of the retention RAM.

40A.3.2.1 List of Registers

Table 40A.70 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
MGDGR	FFC4 9000$_H$	MGDGRPROT0_BKA	GRG protection setting register0 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9004$_H$	MGDGRBAD0_BKA	GRG compare base address register0 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9008$_H$	MGDGRADV0_BKA	GRG compare address valid register0 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9010$_H$	MGDGRPROT1_BKA	GRG protection setting register1 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9014$_H$	MGDGRBAD1_BKA	GRG compare base address register1 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9018$_H$	MGDGRADV1_BKA	GRG compare address valid register1 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9020$_H$	MGDGRPROT2_BKA	GRG protection setting register2 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9024$_H$	MGDGRBAD2_BKA	GRG compare base address register2 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9028$_H$	MGDGRADV2_BKA	GRG compare address valid register2 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9030$_H$	MGDGRPROT3_BKA	GRG protection setting register3 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9034$_H$	MGDGRBAD3_BKA	GRG compare base address register3 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9038$_H$	MGDGRADV3_BKA	GRG compare address valid register3 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9040$_H$	MGDGRSCTL_BKA	GRG control register (Bank A)	W	0000 0000$_H$	8/16/32
	FFC4 9044$_H$	MGDGRSSTAT_BKA	GRG error status register (Bank A)	R	0000 0000$_H$	8/16/32
	FFC4 9048$_H$	MGDGRSTYPE_BKA	GRG error access type register (Bank A)	R	0000 0000$_H$	8/16/32
	FFC4 904C$_H$	MGDGRSAD_BKA	GRG error address register (Bank A)	R	FEE0 0000$_H$	8/16/32
	FFC4 9200$_H$	MGDGRPROT0_BKB	GRG protection setting register0 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9204$_H$	MGDGRBAD0_BKB	GRG compare base address register0 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9208$_H$	MGDGRADV0_BKB	GRG compare address valid register0 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9210$_H$	MGDGRPROT1_BKB	GRG protection setting register1 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9214$_H$	MGDGRBAD1_BKB	GRG compare base address register1 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9218$_H$	MGDGRADV1_BKB	GRG compare address valid register1 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9220$_H$	MGDGRPROT2_BKB	GRG protection setting register2 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9224$_H$	MGDGRBAD2_BKB	GRG compare base address register2 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9228$_H$	MGDGRADV2_BKB	GRG compare address valid register2 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9230$_H$	MGDGRPROT3_BKB	GRG protection setting register3 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9234$_H$	MGDGRBAD3_BKB	GRG compare base address register3 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9238$_H$	MGDGRADV3_BKB	GRG compare address valid register3 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9240$_H$	MGDGRSCTL_BKB	GRG control register (Bank B)	W	0000 0000$_H$	8/16/32
	FFC4 9244$_H$	MGDGRSSTAT_BKB	GRG error status register (Bank B)	R	0000 0000$_H$	8/16/32
	FFC4 9248$_H$	MGDGRSTYPE_BKB	GRG error access type register (Bank B)	R	0000 0000$_H$	8/16/32
	FFC4 924C$_H$	MGDGRSAD_BKB	GRG error address register (Bank B)	R	FEF0 0000$_H$	8/16/32

Note: MGDGRPROTn_BKA/BKB, MGDGRBADn_BKA/BKB, and MGDGRADVn_BKA/BKB set the protection specifications for each channel (n = 0 to 3).
MGDGRSCTL_*, MGDGRSSTAT_*, MGDGRSTYPE_* and MGDGRSAD_* indicate error information on each bank.
“BKA” represents for Bank A and “BKB” represents for Bank B.

40A.3.2.2 Details of Registers

(1) MGDGRPROTn_BKA/BKB — GRG Protection Setting Register n (n = 0 to 3)

Access: MGDGRPROTn_BKA and MGDGRPROTn_BKB can be read or written in 32-bit units
MGDGRPROTn_BKAL, MGDGRPROTn_BKBL, MGDGRPROTn_BKAH and MGDGRPROTn_BKBH can be read or written in 16-bit units
MGDGRPROTn_BKALL, MGDGRPROTn_BKAHL, MGDGRPROTn_BKAHH, MGDGRPROTn_BKBLL, MGDGRPROTn_BKBHL and MGDGRPROTn_BKBHH can be read or written in 8-bit units.

Address: MGDGRPROTn_BKA: FFC4 9000$_H$ + n × 10$_H$
MGDGRPROTn_BKAL: FFC4 9000$_H$ + n × 10$_H$
MGDGRPROTn_BKALL: FFC4 9000$_H$ + n × 10$_H$
MGDGRPROTn_BKAH: FFC4 9002$_H$ + n × 10$_H$
MGDGRPROTn_BKAHL: FFC4 9002$_H$ + n × 10$_H$
MGDGRPROTn_BKAHH: FFC4 9003$_H$ + n × 10$_H$
MGDGRPROTn_BKB: FFC4 9200$_H$ + n × 10$_H$
MGDGRPROTn_BKBL: FFC4 9200$_H$ + n × 10$_H$
MGDGRPROTn_BKBLL: FFC4 9200$_H$ + n × 10$_H$
MGDGRPROTn_BKBH: FFC4 9202$_H$ + n × 10$_H$
MGDGRPROTn_BKBHL: FFC4 9202$_H$ + n × 10$_H$
MGDGRPROTn_BKBHH: FFC4 9203$_H$ + n × 10$_H$

Value after reset: 03FE 01F0$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	EN	—	—	—	—	UM	—	—	—	PEID4	—	PEID2	PEID1	—	—
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
R/W	R	R/W	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SPID3	SPID2	SPID1	SPID0	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R

Table 40A.71 MGDGRPROTn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	EN	Protection Enable 0: Disables protection. 1: Enables protection. Only access permitted by this register is possible.
29 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	UM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40A.71 MGDGRPROTn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
19	PEID2	Access with PEID = 2 (CPU2)*[1] 0: Disables access with PEID2. 1: Enables access with PEID2.
18	PEID1	Access with PEID = 1 (CPU1)*[1] 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SPID3	Access with SPID = 3*[2] 0: Disables access with SPID3. 1: Enables access with SPID3.
7	SPID2	Access with SPID = 2 (CPU2*[3])*[2] 0: Disables access with SPID2. 1: Enables access with SPID2.
6	SPID1	Access with SPID = 1 (CPU1*[3])*[2] 0: Disables access with SPID1. 1: Enables access with SPID1.
5	SPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*[2] 0: Disables access with SPID0. 1: Enables access with SPID0.
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. Access with PEID
PEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
SPID is a bit list with each bit corresponding to a SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

CAUTION

Global RAM Guard (GRG) only supports write accesses. The guard setting is not enabled for read accesses.

(2) MGDGRBADn_BKA/BKB — GRG Compare Base Address Register n (n = 0 to 3)

Access: MGDGRBADn_BKA and MGDGRBADn_BKB can be read or written in 32-bit units.
MGDGRBADn_BKAL, MGDGRBADn_BKAH, MGDGRBADn_BKBL and MGDGRBADn_BKBH can be read or written in 16-bit units.
MGDGRBADn_BKALH, MGDGRBADn_BKAHL, MGDGRBADn_BKBLH and MGDGRBADn_BKBHL can be read or written in 8-bit units.

Address:
MGDGRBADn_BKA: FFC4 9004_H + n × 10_H
MGDGRBADn_BKAL: FFC4 9004_H + n × 10_H
MGDGRBADn_BKALH: FFC4 9005_H + n × 10_H
MGDGRBADn_BKAH: FFC4 9006_H + n × 10_H
MGDGRBADn_BKAHL: FFC4 9006_H + n × 10_H
MGDGRBADn_BKB: FFC4 9204_H + n × 10_H
MGDGRBADn_BKBL: FFC4 9204_H + n × 10_H
MGDGRBADn_BKBLH: FFC4 9205_H + n × 10_H
MGDGRBADn_BKBH: FFC4 9206_H + n × 10_H
MGDGRBADn_BKBHL: FFC4 9206_H + n × 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	AD[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD[15:11]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 40A.72 MGDGRBADn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 11	AD[19:11]	Compare Base Address*[1] These bits set the base address of protection setting domain specified by the GRG protection setting register n (MGDGRPROTn_BKA/BKB).
10 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. For details, see **Section 40A.3.2.2(3), MGDGRADVn_BKA/BKB — GRG Compare Address Valid Register n (n = 0 to 3)**.

(3) MGDGRADVn_BKA/BKB — GRG Compare Address Valid Register n (n = 0 to 3)

Access: MGDGRADVn_BKA and MGDGRADVn_BKB can be read or written in 32-bit units.

MGDGRADVn_BKAL, MGDGRADVn_BKAH, MGDGRADVn_BKBL and MGDGRADVn_BKBH can be read or written in 16-bit units.

MGDGRADVn_BKALH, MGDGRADVn_BKAHL, MGDGRADVn_BKBLH and MGDGRADVn_BKBHL can be read or written in 8-bit units.

Address:
MGDGRADVn_BKA: FFC4 9008_H + n × 10_H
MGDGRADVn_BKAL: FFC4 9008_H + n × 10_H
MGDGRADVn_BKALH: FFC4 9009_H + n × 10_H
MGDGRADVn_BKAH: FFC4 $900A_H$ + n × 10_H
MGDGRADVn_BKAHL: FFC4 $900A_H$ + n × 10_H
MGDGRADVn_BKB: FFC4 9208_H + n × 10_H
MGDGRADVn_BKBL: FFC4 9208_H + n × 10_H
MGDGRADVn_BKBLH: FFC4 9209_H + n × 10_H
MGDGRADVn_BKBH: FFC4 $920A_H$ + n × 10_H
MGDGRADVn_BKBHL: FFC4 $920A_H$ + n × 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	ADV[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:11]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 40A.73 MGDGRADVn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 11	ADV[19:11]	Valid Compare Address Setting MGDGRADVn_BKA/BKB[19:11] to 1 executes address compare. If all the bits of MGDGRADVn_BKA/BKB[19:11] are 1, 2 Kbytes, which is the minimum unit based on the address specified by MGDGRBADn_BKA/BKB, are protected. However, if all the bits of MGDGRADVn_BKA/BKB[19:11] are 0, all the areas of global RAM are protected.
10 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Setting example

If MGDGRBADn_BKA.AD[19:11] is 170$_H$ and MGDGRADVn_BKA.ADV[19:11] is 1FD$_H$, global RAM guard protection area n is from FEEB 8000$_H$ to FEEB 87FF$_H$ and FEEB 9000$_H$ to FEEB 97FF$_H$.

If MGDGRBADn_BKB.AD[19:11] is 170$_H$ and MGDGRADVn_BKB.ADV[19:11] is 1FD$_H$, global RAM guard protection area n is from FEFB 8000$_H$ to FEFB 87FF$_H$ and FEFB 9000$_H$ to FEFB 97FF$_H$.

Concept (e.g. bank A)

When MGDGRBADn_BKA.AD[19:11] is 170$_H$, the base address is FEEB 8000$_H$.
The settable range is shown by an underline as follows:

1 1 1 1	1 1 1 0	1 1 1 0	1 0 1 1	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	B	8	0	0	0	H

The settable range is as follows when MGDGRADVn_BKA.ADV[19:11] is 1FD$_H$:

0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 0	1 0 0 0	0 0 0 0	0 0 0 0	B

The area to which protection applies is as shown below because the bits set to 0 and the lower eleven bits are not applicable:

0 0 0 0	0 0 0 0	0 0 0 0	1 0 1 1	1 0 0 x	0 x x x	x x x x	x x x x	B

Therefore the protection range is as follows:

1 1 1 1	1 1 1 0	1 1 1 0	1 0 1 1	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	B	8	0	0	0	H

to

1 1 1 1	1 1 1 0	1 1 1 0	1 0 1 1	1 0 0 0	0 1 1 1	1 1 1 1	1 1 1 1	B
F	E	E	B	8	7	F	F	H

and

1 1 1 1	1 1 1 0	1 1 1 0	1 0 1 1	1 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	B	9	0	0	0	H

to

1 1 1 1	1 1 1 0	1 1 1 0	1 0 1 1	1 0 0 1	0 1 1 1	1 1 1 1	1 1 1 1	B
F	E	E	B	9	7	F	F	H

The following 2 Kbytes each (a total of 4 Kbytes) are protected (in hexadecimal):

- FEEB 8000$_H$ to FEEB 87FF$_H$
- FEEB 9000$_H$ to FEEB 97FF$_H$

(4) MGDGRSCTL_BKA/BKB — GRG Control Register

Access: MGDGRSCTL_BKA and MGDGRSCTL_BKB registers are write-only registers that can be written in 32-bit units.
MGDGRSCTL_BKAL and MGDGRSCTL_BKBL registers are write-only registers that can be written in 16-bit units.
MGDGRSCTL_BKALL and MGDGRSCTL_BKBLL registers are write-only registers that can be written in 8-bit units.

Address: MGDGRSCTL_BKA: FFC4 9040$_{\text{H}}$
MGDGRSCTL_BKAL: FFC4 9040$_{\text{H}}$
MGDGRSCTL_BKALL: FFC4 9040$_{\text{H}}$
MGDGRSCTL_BKB: FFC4 9240$_{\text{H}}$
MGDGRSCTL_BKBL: FFC4 9240$_{\text{H}}$
MGDGRSCTL_BKBLL: FFC4 9240$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERRCLO	ERRCLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40A.74 MGDGRSCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	ERRCLO	Error Entry Overflow Flag Clear 0: No operation 1: Clears the overflow flag.
0	ERRCLE	GRG Protection Violation Flag Clear 0: No operation 1: Clears the error flag. Set this bit at the same time as ERRCLO as shown in **Table 40A.75, Combinations of ERRCLO and ERRCLE**.

Table 40A.75 Combinations of ERRCLO and ERRCLE

ERRCLO	ERRCLE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(5) MGDGRSSTAT_BKA/BKB— GRG Error Status Register

Access:	MGDGRSSTAT_BKA and MGDGRSSTAT_BKB are read-only registers that can be read in 32-bit units. MGDGRSSTAT_BKAL and MGDGRSSTAT_BKBL are read-only registers that can be read in 16-bit units. MGDGRSSTAT_BKALL and MGDGRSSTAT_BKBLL are read-only registers that can be read in 8-bit units.
Address:	MGDGRSSTAT_BKA: FFC4 9044_H MGDGRSSTAT_BKAL: FFC4 9044_H MGDGRSSTAT_BKALL: FFC4 9044_H MGDGRSSTAT_BKB: FFC4 9244_H MGDGRSSTAT_BKBL: FFC4 9244_H MGDGRSSTAT_BKBLL: FFC4 9244_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.76 MGDGRSSTAT_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of GRG error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether overflow has occurred in INTGUARD. Check the value of this bit. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	GRG protection violation Error Flag 0: No GRG protection violation has occurred. 1: A GRG protection violation has occurred.

(6) MGDGRSTYPE_BKA/BKB — GRG Error Access Type Register

Access: MGDGRSTYPE_BKA and MGDGRSTYPE_BKB are read-only registers that can be read in 32-bit units.
MGDGRSTYPE_BKAL and MGDGRSTYPE_BKBL are read-only registers that can be read in 16-bit units.
MGDGRSTYPE_BKALL, MGDGRSTYPE_BKALH, MGDGRSTYPE_BKBLL and MGDGRSTYPE_BKBLH are read-only registers that can be read in 8-bit units.

Address:
MGDGRSTYPE_BKA: FFC4 9048$_H$
MGDGRSTYPE_BKAL: FFC4 9048$_H$
MGDGRSTYPE_BKALL: FFC4 9048$_H$
MGDGRSTYPE_BKALH: FFC4 9049$_H$
MGDGRSTYPE_BKB: FFC4 9248$_H$
MGDGRSTYPE_BKBL: FFC4 9248$_H$
MGDGRSTYPE_BKBLL: FFC4 9248$_H$
MGDGRSTYPE_BKBLH: FFC4 9249$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.77 MGDGRSTYPE_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	The PEID of the originator of the access which led to a GRG protection violation
12 to 10	Reserved	When read, the value after reset is returned.
9 to 8	SPID[1:0]	The SPID of the originator of the access which led to a GRG protection violation
7	Reserved	When read, the value after reset is returned.
6	UM	The UM of the originator of the access which led to a GRG protection violation
5 to 0	Reserved	When read, the value after reset is returned.

(7) MGDGRSAD_BKA/BKB — GRG Error Address Register

Access: MGDGRSAD_BKA and MGDGRSAD_BKB are read-only registers that can be read in 32-bit units.
MGDGRSAD_BKAL, MGDGRSAD_BKAH, MGDGRSAD_BKBL and MGDGRSAD_BKBH are read-only registers that can be read in 16-bit units.
MGDGRSAD_BKALL, MGDGRSAD_BKALH, MGDGRSAD_BKAHL, MGDGRSAD_BKBLL, MGDGRSAD_BKBLH and MGDGRSAD_BKBHL are read-only registers that can be read in 8-bit units.

Address:
MGDGRSAD_BKA: FFC4 904C$_H$
MGDGRSAD_BKAL: FFC4 904C$_H$
MGDGRSAD_BKALL: FFC4 904C$_H$
MGDGRSAD_BKALH: FFC4 904D$_H$
MGDGRSAD_BKAH: FFC4 904E$_H$
MGDGRSAD_BKAHL: FFC4 904E$_H$
MGDGRSAD_BKB: FFC4 924C$_H$
MGDGRSAD_BKBL: FFC4 924C$_H$
MGDGRSAD_BKBLL: FFC4 924C$_H$
MGDGRSAD_BKBLH: FFC4 924D$_H$
MGDGRSAD_BKBH: FFC4 924E$_H$
MGDGRSAD_BKBHL: FFC4 924E$_H$

Value after reset: MGDGRSAD_BKA: FEE0 0000$_H$, MGDGRSAD_BKB: FEF0 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SAD[19:16]			
Value after reset	1	1	1	1	1	1	1	0	1	1	1	0/1^{*1}	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.78 MGDGRSAD_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	SAD[19:0]	Address at GRG violation was occurred.

Note 1. The value after reset differs depending on bank A or bank B.

40A.3.3 PBG/HBG

The PBG/HBG module is divided into several PBG/HBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG/HBG channel can designate the access against which a single peripheral circuit should be protected.

Each PBG/HBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG/HBG group names, and the PBG/HBG channel numbers.

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG10	00	0	ECON_NMI	WDTNMIF WDTNMIFC
		1	ECON_FEINT	FEINTF FEINTFMSK FEINTFC
		2	SL_INTC	SELB_INTC1
		3	ADCA0	All registers in ADCA0
		4	KR0	KR0KRM
		5	PORT (Group A_ISO)	Registers of P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24, and AP1 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		6	PORT (Group A_AWO)	Registers of P0, P1, P2, P3, P8, AP0, and IP0 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		7	JTAG (Group A)	JP0 register (except JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)
		8	RLN30	All registers in RLIN30
		9	RLN31	All registers in RLIN31
		10	RLN32	All registers in RLIN32
		11	RLN33	All registers in RLIN33
		12	RLN34	All registers in RLIN34
		13	RLN35	All registers in RLIN35
		14	RLN36	All registers in RLIN36
		15	RLN37	All registers in RLIN37
PBG11	01	0	DNF (TAUD0)	DNFATAUD0ICTL DNFATAUD0IEN
		1	DNF (ADCA0)	DNFAADCTL0CTL DNFAADCTL0EN
		2	FCLA0 (ADCA0)	FCLA0CTLn_ADC0 (n = 0 to 2)
		3	FCLA0 (NMI)	FCLA0CTL0_NMI
		4	FCLA0 (INTPn)	FCLA0CTLn_INTPL (n = 0 to 7)
		5	FCLA0 (INTPn)	FCLA0CTLn_INTPH (n = 0 to 7)
		6	DNF (ENCA0)	DNFAENCA0ICTL DNFAENCA0IEN
		7	DNF (ADCA1)	DNFAADCTL1CTL DNFAADCTL1EN
		8	FCLA0 (ADCA1)	FCLA0CTLn_ADC1 (n=0 to 2)

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG11	01	9	DNF (TAUB0)	DNFATAUB0ICTL DNFATAUB0IEN
		10	DNF (TAUB1)	DNFATAUB1ICTL DNFATAUB1IEN
		11	Reserved area	—
		12	PORT (Group B_ISO)	Registers of P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24, and AP1 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		13	PORT (Group B_AWO)	Registers of P0, P1, P2, P3, P8, AP0, and IP0 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		14	JTAG (Group B)	JP0 register (JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)
		15	FCLA0 (INTPn)	FCLA0CTLn_INTPU (n = 0 to 7)
PBG12	07	0	RLN240 (Global)	All global registers in RLN240
		1	RLN2400	All channel registers in RLN2400
		2	RLN2401	All channel registers in RLN2401
		3	RLN2402	All channel registers in RLN2402
		4	RLN2403	All channel registers in RLN2403
		5	RLN241 (Global)	All global registers in RLN241
		6	RLN2414	All channel registers in RLN2414
		7	RLN2415	All channel registers in RLN2415
		8	RLN2416	All channel registers in RLN2416
		9	RLN2417	All channel registers in RLN2417
		10	RLN242 (Global)	All global registers in RLN242
		11	RLN2428	All channel registers in RLN2428
		12	RLN2429	All channel registers in RLN2429
		13	RLN24210	All channel registers in RLN24210
		14	RLN24211	All channel registers in RLN24211
		15	DNF (RSENTn)	DNFASENTICTL DNFASENTIEN
PBG13	08	0	DCRA0	All registers in DCRA0
		1	DCRA1	All registers in DCRA1
		2	DCRA2	All registers in DCRA2
		3	DCRA3	All registers in DCRA3
		4	RIIC0	All registers in RIIC0
		5	SL_READTEST	SELB_READTEST
		6	SL_DMAC	DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1
		7	GRZF	All registers in GRZF
		8	RIIC1	All registers in RIIC1
		9	Reserved area	—
		10	Reserved area	—
		11	Reserved area	—
		12	Reserved area	—
		13	RSENT0	All registers in RSENT0
		14	RSENT1	All registers in RSENT1

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG14	12	0	RLN243 (Global)	All global registers in RLN243
		1	RLN24312	All channel registers in RLN24312
		2	RLN24313	All channel registers in RLN24313
		3	RLN24314	All channel registers in RLN24314
		4	RLN24315	All channel registers in RLN24315
PBG20	02	0	TAUD0	All registers in TAUD0 (except SELB_TAUD0I)
		1	SL_TAUD0	SELB_TAUD0I
		2	TAUJ0	All registers in TAUJ0 (except SELB_TAUJ0I)
		3	SL_TAUJ0	SELB_TAUJ0I
		4	RTCA0	All registers in RTCA0
		5	WDTA0	All registers in WDTA0
		6	WDTA1	All registers in WDTA1
		7	WDTA2	All registers in WDTA2
		8	PIC0	All registers in PIC0
		9	TAPA0	All registers in TAPA0
		10	ENCA0	All registers in ENCA0
		11	TAUJ1	All registers in TAUJ1
		12	TAUB0	All registers in TAUB0 (except SELB_TAUB0I)
		13	TAUB1	All registers in TAUB1 (except SELB_TAUB1I)
		14	PWBAn PWGAn PWSAn SLPWG PWGA_INTF	All registers in PWM-Diag
		15	SL_TAUJ2	SELB_TAUJ2I
PBG21	09	0	Reserved area	—
		1	Reserved area	—
		2	Reserved area	—
		3	Reserved area	—
		4	Reserved area	—
		5	SL_TAUB0	SELB_TAUB0I
		6	SL_TAUB1	SELB_TAUB1I
		7	Reserved area	—
		8	TAUJ2	All registers in TAUJ2 (except SELB_TAUJ2I)
		9	TAUJ3	All registers in TAUJ3
		10	MMCA0	All registers in MMCA0
		11	ECCMMCA0A	All registers in ECC MMCA0A
		12	ECCMMCA0B	All registers in ECC MMCA0B

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG30	03	0	Reserved area	—
		1	Reserved area	—
		2	Reserved area	—
		3	Reserved area	—
		4	Reserved area	—
		5	Reserved area	—
		6	Reserved area	—
		7	Reserved area	—
		8	Reserved area	—
		9	Reserved area	—
		10	ETNB0	All registers in ETNB0 (except ETNB0IFCTL)
		11	Reserved area	—
		12	ADCA1	All registers in ADCA1
		13	ECCETNB0TX	All register in ECCETNB0TX group
		14	ECCETNB0RX	All register in ECCETNB0RX group
PBG31	04	0	OSTM0	All registers in OSTM0
		1	OSTM1-4	All registers in OSTM1, OSTM2, OSTM3, and OSTM4
		2	OSTM5	All registers in OSTM5
		3	OSTM6-9	All registers in OSTM6, OSTM7, OSTM8, and OSTM9
		4	ECCCSIH0	All registers in ECC CSIH0
		5	ECCCSIH1	All registers in ECC CSIH1
		6	ECCCSIH2	All registers in ECC CSIH2
		7	ECCCSIH3	All registers in ECC CSIH3
		8	Reserved area	—
		9	Reserved area	—
		10	Reserved area	—
		11	Reserved area	—
		12	ECCFLXA0	All register in ECCFLXA0 group
		13	ECCFLXA0T0	All register in ECCFLXA0T0 group
		14	ECCFLXA0T1	All register in ECCFLXA0T1 group
PBG32	05	0	CSIH0 (Group A)	CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU
		1	CSIH0 (Group B)	CSIH0 registers other than the above
		2	CSIH1 (Group A)	CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU
		3	CSIH1 (Group B)	CSIH1 registers other than the above
		4	CSIH2 (Group A)	CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU
		5	CSIH2 (Group B)	CSIH2 registers other than the above
		6	CSIH3 (Group A)	CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU
		7	CSIH3 (Group B)	CSIH3 registers other than the above
		8	CSIG0 (Group A)	CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU
		9	CSIG0 (Group B)	CSIG0 registers other than the above

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG32	05	10	CSIG1 (Group A)	CSIG1CTL0-2, CSIG1STR0, CSIG1STCR0, CSIG1EMU
		11	CSIG1 (Group B)	CSIG1 registers other than the above
		12	CSIG2 (Group A)	CSIG2CTL0-2, CSIG2STR0, CSIG2STCR0, CSIG2EMU
		13	CSIG2 (Group B)	CSIG2 registers other than the above
		14	CSIG3 (Group A)	CSIG3CTL0-2, CSIG3STR0, CSIG3STCR0, CSIG3EMU
		15	CSIG3 (Group B)	CSIG3 registers other than the above
PBG33	13	0	CSIH4 (Group A)	CSIH4CTL0-2, CSIH4STR0, CSIH4STCR0, CSIH4EMU
		1	CSIH4 (Group B)	CSIH4 registers other than the above
		2	ECCCSIH4	All registers in ECC CSIH4
		3	CSIG4 (Group A)	CSIG4CTL0-2, CSIG4STR0, CSIG4STCR0, CSIG4EMU
		4	CSIG4 (Group B)	CSIG4 registers other than the above
		5	ETNB1	All registers in ETNB1 (except ETNB1IFCTL)
		6	ECCETNB1TX	All register in ECCETNB1TX group
		7	ECCETNB1RX	All register in ECCETNB1RX group
PBG40	10	0	Flash memory (DCIB)	EEPRDCYCL
		1	DFECC	DFECCCTL, DFERSTR, DFERSTC, DFOVFSTR, DFOVFSTC, DFERRINT, DFEADR, DFTSTCTL
PBG50	06	0	System control*1	All registers in Write-Protect Function, Reset Controller, Power Supply Circuit, Supply Voltage Monitor, Clock Controller, Clock Monitor, Stand-By Controller, and Low-Power Sampler*1 (except STBC0PSC, STBC0STPT, SWRESA, PROTCMD0, PROTS0, JPPCMD0, JPPROTS0, PPCMD0-3, PPCMD8-13, PPCMD18-24, PPROTS0-3, PPROTS8-13, PPROTS18-24, FLMDPCMD, FLMDPS)
		1	STBC0	STBC0PSC, STBC0STPT
		2	Reserved area	—
		3	Reserved area	—
		4	RESCTL	SWRESA
		5	Flash memory (Self Programming)	FLMD, *2
		6	Flash memory (Control)	—*2
		7	Flash memory (SCDS)	PRDNAME1-3, CHIPID1-2
		8	WPROTR	PROTCMD0 PROTS0

Table 40A.79 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG60	11	0	RCFDC0 (channel 0)	All registers in RCFDC0 Ch0 group*4
		1	RCFDC0 (channel 1)	All registers in RCFDC0 Ch1 group*4
		2	RCFDC0 (channel 2)	All registers in RCFDC0 Ch2 group*4
		3	RCFDC0 (channel 3)	All registers in RCFDC0 Ch3 group*4
		4	RCFDC0 (channel 4)	All registers in RCFDC0 Ch4 group*4
		5	RCFDC0 (channel 5)	All registers in RCFDC0 Ch5 group*4
		6	RCFDC0 (channel 6)	All registers in RCFDC0 Ch6 group*4
		7	RCFDC0 (channel 7)	All registers in RCFDC0 Ch7 group*4
		8	RCFDC0 (Global)	All registers in RCFDC0 Global group*4
		9	ECCCFD0MB	All registers in ECCCFD0MB
		10	ECCCFD0AFL0	All registers in ECCCFD0AFL0
		11	ECCCFD0AFL1	All registers in ECCCFD0AFL1
PBG61	14	0	RCFDC1_CAN0	All registers in RCFDC1 Ch0 group
		1	RCFDC1_CAN1	All registers in RCFDC1 Ch1 group
		2	RCFDC1_CAN2	All registers in RCFDC1 Ch2 group
		3	RCFDC1_CAN3	All registers in RCFDC1 Ch3 group
		4	RCFDC1_Global	All registers in RCFDC1 Global group
		5	ECCCFD1MB	All registers in ECCCFD1MB
		6	ECCCFD1AFL0	All registers in ECCCFD1AFL0
		7	ECCCFD1AFL1	All registers in ECCCFD1AFL1
HBG00	00	0	MEMC0	All registers in MEMC0
		1	MEMC0_CS0	External Memory Area(CS0)
		2	MEMC0_CS1	External Memory Area(CS1)
		3	MEMC0_CS2	External Memory Area(CS2)
		4	MEMC0_CS3	External Memory Area(CS3)
HBG01	01	0	SFMA0	All registers in SFMA0
		1	SFMA0_MEM	External Serial Flash Memory Area
HBG02	02	0	FLXA0	All registers in FLXA0

Note 1. For details, see **Section 5, Write-Protected Registers**, **Section 9A, Reset Controller of RH850/F1KH-D8**, **Section 10A, Power Supply Circuit of RH850/F1KH-D8**, **Section 11A, Supply Voltage Monitor of RH850/F1KH-D8**, **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**, **Section 13, Clock Monitor (CLMA)**, **Section 14, Stand-By Controller (STBC)**, and **Section 15, Low-Power Sampler (LPS)**.

Note 2. Regarding the PBG registers for the flash memory, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Note 3. Regarding the PBG register addresses, see **Table 40A.80, List of PBG/HBG Protection Registers**.

Note 4. Regarding the RS-CANFD guard group, see **Section 24, CANFD Interface (RS-CANFD)**.

NOTE

Be sure to enable PBG/HBG before disabling register access clock of each clock domain.

40A.3.3.1 List of Registers

The following table lists the registers provided for each PBG/HBG group. And PBG/HBG group is equal to module name.

Table 40A.80 List of PBG/HBG Protection Registers

PBG/ HBG Group^{*1}	Group No.^{*2}	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG10	00	FSGD00PROT0	PBG00 protection register 0	R/W	066F FFF7$_H$	FFC4 0000$_H$	8/16/32	ISO
		FSGD00PROT1	PBG00 protection register 1	R/W	066F FFF7$_H$	FFC4 0004$_H$	8/16/32	
		FSGD00PROT2	PBG00 protection register 2	R/W	066F FFF7$_H$	FFC4 0008$_H$	8/16/32	
		FSGD00PROT3	PBG00 protection register 3	R/W	066F FFF7$_H$	FFC4 000C$_H$	8/16/32	
		FSGD00PROT4	PBG00 protection register 4	R/W	066F FFF7$_H$	FFC4 0010$_H$	8/16/32	
		FSGD00PROT5	PBG00 protection register 5	R/W	066F FFF7$_H$	FFC4 0014$_H$	8/16/32	
		FSGD00PROT6	PBG00 protection register 6	R/W	066F FFF7$_H$	FFC4 0018$_H$	8/16/32	
		FSGD00PROT7	PBG00 protection register 7	R/W	066F FFF7$_H$	FFC4 001C$_H$	8/16/32	
		FSGD00PROT8	PBG00 protection register 8	R/W	066F FFF7$_H$	FFC4 0020$_H$	8/16/32	
		FSGD00PROT9	PBG00 protection register 9	R/W	066F FFF7$_H$	FFC4 0024$_H$	8/16/32	
		FSGD00PROT10	PBG00 protection register 10	R/W	066F FFF7$_H$	FFC4 0028$_H$	8/16/32	
		FSGD00PROT11	PBG00 protection register 11	R/W	066F FFF7$_H$	FFC4 002C$_H$	8/16/32	
		FSGD00PROT12	PBG00 protection register 12	R/W	066F FFF7$_H$	FFC4 0030$_H$	8/16/32	
		FSGD00PROT13	PBG00 protection register 13	R/W	066F FFF7$_H$	FFC4 0034$_H$	8/16/32	
		FSGD00PROT14	PBG00 protection register 14	R/W	066F FFF7$_H$	FFC4 0038$_H$	8/16/32	
		FSGD00PROT15	PBG00 protection register 15	R/W	066F FFF7$_H$	FFC4 003C$_H$	8/16/32	
PBG11	01	FSGD01PROT0	PBG01 protection register 0	R/W	066F FFF7$_H$	FFC4 0100$_H$	8/16/32	ISO
		FSGD01PROT1	PBG01 protection register 1	R/W	066F FFF7$_H$	FFC4 0104$_H$	8/16/32	
		FSGD01PROT2	PBG01 protection register 2	R/W	066F FFF7$_H$	FFC4 0108$_H$	8/16/32	
		FSGD01PROT3	PBG01 protection register 3	R/W	066F FFF7$_H$	FFC4 010C$_H$	8/16/32	
		FSGD01PROT4	PBG01 protection register 4	R/W	066F FFF7$_H$	FFC4 0110$_H$	8/16/32	
		FSGD01PROT5	PBG01 protection register 5	R/W	066F FFF7$_H$	FFC4 0114$_H$	8/16/32	
		FSGD01PROT6	PBG01 protection register 6	R/W	066F FFF7$_H$	FFC4 0118$_H$	8/16/32	
		FSGD01PROT7	PBG01 protection register 7	R/W	066F FFF7$_H$	FFC4 011C$_H$	8/16/32	
		FSGD01PROT8	PBG01 protection register 8	R/W	066F FFF7$_H$	FFC4 0120$_H$	8/16/32	
		FSGD01PROT9	PBG01 protection register 9	R/W	066F FFF7$_H$	FFC4 0124$_H$	8/16/32	
		FSGD01PROT10	PBG01 protection register 10	R/W	066F FFF7$_H$	FFC4 0128$_H$	8/16/32	
		—	Reserved	—	—	FFC4 012C$_H$	—	
		FSGD01PROT12	PBG01 protection register 12	R/W	066F FFF7$_H$	FFC4 0130$_H$	8/16/32	
		FSGD01PROT13	PBG01 protection register 13	R/W	066F FFF7$_H$	FFC4 0134$_H$	8/16/32	
		FSGD01PROT14	PBG01 protection register 14	R/W	066F FFF7$_H$	FFC4 0138$_H$	8/16/32	
		FSGD01PROT15	PBG01 protection register 15	R/W	066F FFF7$_H$	FFC4 013C$_H$	8/16/32	

Table 40A.80 List of PBG/HBG Protection Registers

PBG/ HBG Group^{*1}	Group No.^{*2}	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG12	07	FSGD07PROT0	PBG07 protection register 0	R/W	066F FFF7$_H$	FFC4 0400$_H$	8/16/32	ISO
		FSGD07PROT1	PBG07 protection register 1	R/W	066F FFF7$_H$	FFC4 0404$_H$	8/16/32	
		FSGD07PROT2	PBG07 protection register 2	R/W	066F FFF7$_H$	FFC4 0408$_H$	8/16/32	
		FSGD07PROT3	PBG07 protection register 3	R/W	066F FFF7$_H$	FFC4 040C$_H$	8/16/32	
		FSGD07PROT4	PBG07 protection register 4	R/W	066F FFF7$_H$	FFC4 0410$_H$	8/16/32	
		FSGD07PROT5	PBG07 protection register 5	R/W	066F FFF7$_H$	FFC4 0414$_H$	8/16/32	
		FSGD07PROT6	PBG07 protection register 6	R/W	066F FFF7$_H$	FFC4 0418$_H$	8/16/32	
		FSGD07PROT7	PBG07 protection register 7	R/W	066F FFF7$_H$	FFC4 041C$_H$	8/16/32	
		FSGD07PROT8	PBG07 protection register 8	R/W	066F FFF7$_H$	FFC4 0420$_H$	8/16/32	
		FSGD07PROT9	PBG07 protection register 9	R/W	066F FFF7$_H$	FFC4 0424$_H$	8/16/32	
		FSGD07PROT10	PBG07 protection register 10	R/W	066F FFF7$_H$	FFC4 0428$_H$	8/16/32	
		FSGD07PROT11	PBG07 protection register 11	R/W	066F FFF7$_H$	FFC4 042C$_H$	8/16/32	
		FSGD07PROT12	PBG07 protection register 12	R/W	066F FFF7$_H$	FFC4 0430$_H$	8/16/32	
		FSGD07PROT13	PBG07 protection register 13	R/W	066F FFF7$_H$	FFC4 0434$_H$	8/16/32	
		FSGD07PROT14	PBG07 protection register 14	R/W	066F FFF7$_H$	FFC4 0438$_H$	8/16/32	
		FSGD07PROT15	PBG07 protection register 15	R/W	066F FFF7$_H$	FFC4 043C$_H$	8/16/32	
PBG13	08	FSGD08PROT0	PBG08 protection register 0	R/W	066F FFF7$_H$	FFC4 0500$_H$	8/16/32	ISO
		FSGD08PROT1	PBG08 protection register 1	R/W	066F FFF7$_H$	FFC4 0504$_H$	8/16/32	
		FSGD08PROT2	PBG08 protection register 2	R/W	066F FFF7$_H$	FFC4 0508$_H$	8/16/32	
		FSGD08PROT3	PBG08 protection register 3	R/W	066F FFF7$_H$	FFC4 050C$_H$	8/16/32	
		FSGD08PROT4	PBG08 protection register 4	R/W	066F FFF7$_H$	FFC4 0510$_H$	8/16/32	
		FSGD08PROT5	PBG08 protection register 5	R/W	066F FFF7$_H$	FFC4 0514$_H$	8/16/32	
		FSGD08PROT6	PBG08 protection register 6	R/W	066F FFF7$_H$	FFC4 0518$_H$	8/16/32	
		FSGD08PROT7	PBG08 protection register 7	R/W	066F FFF7$_H$	FFC4 051C$_H$	8/16/32	
		FSGD08PROT8	PBG08 protection register 8	R/W	066F FFF7$_H$	FFC4 0520$_H$	8/16/32	
		—	Reserved	—	—	FFC4 0524$_H$	—	
		—	Reserved	—	—	FFC4 0528$_H$	—	
		—	Reserved	—	—	FFC4 052C$_H$	—	
		—	Reserved	—	—	FFC4 0530$_H$	—	
		FSGD08PROT13	PBG08 protection register 13	R/W	066F FFF7$_H$	FFC4 0534$_H$	8/16/32	
		FSGD08PROT14	PBG08 protection register 14	R/W	066F FFF7$_H$	FFC4 0538$_H$	8/16/32	
PBG14	12	FSGD12PROT0	PBG12 protection register 0	R/W	066F FFF7$_H$	FFC4 0600$_H$	8/16/32	ISO
		FSGD12PROT1	PBG12 protection register 1	R/W	066F FFF7$_H$	FFC4 0604$_H$	8/16/32	
		FSGD12PROT2	PBG12 protection register 2	R/W	066F FFF7$_H$	FFC4 0608$_H$	8/16/32	
		FSGD12PROT3	PBG12 protection register 3	R/W	066F FFF7$_H$	FFC4 060C$_H$	8/16/32	
		FSGD12PROT4	PBG12 protection register 4	R/W	066F FFF7$_H$	FFC4 0610$_H$	8/16/32	

Table 40A.80 List of PBG/HBG Protection Registers

PBG/ HBG Group*1	Group No.*2	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG20	02	FSGD02PROT0	PBG02 protection register 0	R/W	066F FFF7$_H$	FFDD D000$_H$	8/16/32	ISO
		FSGD02PROT1	PBG02 protection register 1	R/W	066F FFF7$_H$	FFDD D004$_H$	8/16/32	
		FSGD02PROT2	PBG02 protection register 2	R/W	066F FFF7$_H$	FFDD D008$_H$	8/16/32	
		FSGD02PROT3	PBG02 protection register 3	R/W	066F FFF7$_H$	FFDD D00C$_H$	8/16/32	
		FSGD02PROT4	PBG02 protection register 4	R/W	066F FFF7$_H$	FFDD D010$_H$	8/16/32	
		FSGD02PROT5	PBG02 protection register 5	R/W	066F FFF7$_H$	FFDD D014$_H$	8/16/32	
		FSGD02PROT6	PBG02 protection register 6	R/W	0607 FE77$_H$	FFDD D018$_H$	8/16/32	
		FSGD02PROT7	PBG02 protection register 7	R/W	060B FEB7$_H$	FFDD D01C$_H$	8/16/32	
		FSGD02PROT8	PBG02 protection register 8	R/W	066F FFF7$_H$	FFDD D020$_H$	8/16/32	
		FSGD02PROT9	PBG02 protection register 9	R/W	066F FFF7$_H$	FFDD D024$_H$	8/16/32	
		FSGD02PROT10	PBG02 protection register 10	R/W	066F FFF7$_H$	FFDD D028$_H$	8/16/32	
		FSGD02PROT11	PBG02 protection register 11	R/W	066F FFF7$_H$	FFDD D02C$_H$	8/16/32	
		FSGD02PROT12	PBG02 protection register 12	R/W	066F FFF7$_H$	FFDD D030$_H$	8/16/32	
		FSGD02PROT13	PBG02 protection register 13	R/W	066F FFF7$_H$	FFDD D034$_H$	8/16/32	
		FSGD02PROT14	PBG02 protection register 14	R/W	066F FFF7$_H$	FFDD D038$_H$	8/16/32	
		FSGD02PROT15	PBG02 protection register 15	R/W	066F FFF7$_H$	FFDD D03C$_H$	8/16/32	
PBG21	09	—	Reserved	—	—	FFDD D100$_H$	—	ISO
		—	Reserved	—	—	FFDD D104$_H$	—	
		—	Reserved	—	—	FFDD D108$_H$	—	
		—	Reserved	—	—	FFDD D10C$_H$	—	
		—	Reserved	—	—	FFDD D110$_H$	—	
		FSGD09PROT5	PBG09 protection register 5	R/W	066F FFF7$_H$	FFDD D114$_H$	8/16/32	
		FSGD09PROT6	PBG09 protection register 6	R/W	066F FFF7$_H$	FFDD D118$_H$	8/16/32	
		—	Reserved	—	—	FFDD D11C$_H$	—	
		FSGD09PROT8	PBG09 protection register 8	R/W	066F FFF7$_H$	FFDD D120$_H$	8/16/32	
		FSGD09PROT9	PBG09 protection register 9	R/W	066F FFF7$_H$	FFDD D124$_H$	8/16/32	
		FSGD09PROT10	PBG09 protection register 10	R/W	066F FFF7$_H$	FFDD D128$_H$	8/16/32	
		FSGD09PROT11	PBG09 protection register 11	R/W	066F FFF7$_H$	FFDD D12C$_H$	8/16/32	
		FSGD09PROT12	PBG09 protection register 12	R/W	066F FFF7$_H$	FFDD D130$_H$	8/16/32	
PBG30	03	—	Reserved	—	—	FFF9 4000$_H$	—	ISO
		—	Reserved	—	—	FFF9 4004$_H$	—	
		—	Reserved	—	—	FFF9 4008$_H$	—	
		—	Reserved	—	—	FFF9 400C$_H$	—	
		—	Reserved	—	—	FFF9 4010$_H$	—	
		—	Reserved	—	—	FFF9 4014$_H$	—	
		—	Reserved	—	—	FFF9 4018$_H$	—	
		—	Reserved	—	—	FFF9 401C$_H$	—	
		—	Reserved	—	—	FFF9 4020$_H$	—	
		—	Reserved	—	—	FFF9 4024$_H$	—	
		FSGD03PROT10	PBG03 protection register 10	R/W	066F FFF7$_H$	FFF9 4028$_H$	8/16/32	
		—	Reserved	—	—	FFF9 402C$_H$	—	
		FSGD03PROT12	PBG03 protection register 12	R/W	066F FFF7$_H$	FFF9 4030$_H$	8/16/32	
		FSGD03PROT13	PBG03 protection register 13	R/W	066F FFF7$_H$	FFF9 4034$_H$	8/16/32	
		FSGD03PROT14	PBG03 protection register 14	R/W	066F FFF7$_H$	FFF9 4038$_H$	8/16/32	

Table 40A.80 List of PBG/HBG Protection Registers

PBG/ HBG Group*1	Group No.*2	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG31	04	FSGD04PROT0	PBG04 protection register 0	R/W	066F FFF7$_{H}$	FFF9 4100$_{H}$	8/16/32	ISO
		FSGD04PROT1	PBG04 protection register 1	R/W	0607 FE77$_{H}$	FFF9 4104$_{H}$	8/16/32	
		FSGD04PROT2	PBG04 protection register 2	R/W	066F FFF7$_{H}$	FFF9 4108$_{H}$	8/16/32	
		FSGD04PROT3	PBG04 protection register 3	R/W	060B FEB7$_{H}$	FFF9 410C$_{H}$	8/16/32	
		FSGD04PROT4	PBG04 protection register 4	R/W	066F FFF7$_{H}$	FFF9 4110$_{H}$	8/16/32	
		FSGD04PROT5	PBG04 protection register 5	R/W	066F FFF7$_{H}$	FFF9 4114$_{H}$	8/16/32	
		FSGD04PROT6	PBG04 protection register 6	R/W	066F FFF7$_{H}$	FFF9 4118$_{H}$	8/16/32	
		FSGD04PROT7	PBG04 protection register 7	R/W	066F FFF7$_{H}$	FFF9 411C$_{H}$	8/16/32	
		—	Reserved	—	—	FFF9 4120$_{H}$	—	
		—	Reserved	—	—	FFF9 4124$_{H}$	—	
		—	Reserved	—	—	FFF9 4128$_{H}$	—	
		—	Reserved	—	—	FFF9 412C$_{H}$	—	
		FSGD04PROT12	PBG04 protection register 12	R/W	066F FFF7$_{H}$	FFF9 4130$_{H}$	8/16/32	
		FSGD04PROT13	PBG04 protection register 13	R/W	066F FFF7$_{H}$	FFF9 4134$_{H}$	8/16/32	
		FSGD04PROT14	PBG04 protection register 14	R/W	066F FFF7$_{H}$	FFF9 4138$_{H}$	8/16/32	
PBG32	05	FSGD05PROT0	PBG05 protection register 0	R/W	066F FFF7$_{H}$	FFF9 4200$_{H}$	8/16/32	ISO
		FSGD05PROT1	PBG05 protection register 1	R/W	066F FFF7$_{H}$	FFF9 4204$_{H}$	8/16/32	
		FSGD05PROT2	PBG05 protection register 2	R/W	066F FFF7$_{H}$	FFF9 4208$_{H}$	8/16/32	
		FSGD05PROT3	PBG05 protection register 3	R/W	066F FFF7$_{H}$	FFF9 420C$_{H}$	8/16/32	
		FSGD05PROT4	PBG05 protection register 4	R/W	066F FFF7$_{H}$	FFF9 4210$_{H}$	8/16/32	
		FSGD05PROT5	PBG05 protection register 5	R/W	066F FFF7$_{H}$	FFF9 4214$_{H}$	8/16/32	
		FSGD05PROT6	PBG05 protection register 6	R/W	066F FFF7$_{H}$	FFF9 4218$_{H}$	8/16/32	
		FSGD05PROT7	PBG05 protection register 7	R/W	066F FFF7$_{H}$	FFF9 421C$_{H}$	8/16/32	
		FSGD05PROT8	PBG05 protection register 8	R/W	066F FFF7$_{H}$	FFF9 4220$_{H}$	8/16/32	
		FSGD05PROT9	PBG05 protection register 9	R/W	066F FFF7$_{H}$	FFF9 4224$_{H}$	8/16/32	
		FSGD05PROT10	PBG05 protection register 10	R/W	066F FFF7$_{H}$	FFF9 4228$_{H}$	8/16/32	
		FSGD05PROT11	PBG05 protection register 11	R/W	066F FFF7$_{H}$	FFF9 422C$_{H}$	8/16/32	
		FSGD05PROT12	PBG05 protection register 12	R/W	066F FFF7$_{H}$	FFF9 4230$_{H}$	8/16/32	
		FSGD05PROT13	PBG05 protection register 13	R/W	066F FFF7$_{H}$	FFF9 4234$_{H}$	8/16/32	
		FSGD05PROT14	PBG05 protection register 14	R/W	066F FFF7$_{H}$	FFF9 4238$_{H}$	8/16/32	
		FSGD05PROT15	PBG05 protection register 15	R/W	066F FFF7$_{H}$	FFF9 423C$_{H}$	8/16/32	
PBG33	13	FSGD13PROT0	PBG13 protection register 0	R/W	066F FFF7$_{H}$	FFF9 4300$_{H}$	8/16/32	ISO
		FSGD13PROT1	PBG13 protection register 1	R/W	066F FFF7$_{H}$	FFF9 4304$_{H}$	8/16/32	
		FSGD13PROT2	PBG13 protection register 2	R/W	066F FFF7$_{H}$	FFF9 4308$_{H}$	8/16/32	
		FSGD13PROT3	PBG13 protection register 3	R/W	066F FFF7$_{H}$	FFF9 430C$_{H}$	8/16/32	
		FSGD13PROT4	PBG13 protection register 4	R/W	066F FFF7$_{H}$	FFF9 4310$_{H}$	8/16/32	
		FSGD13PROT5	PBG13 protection register 5	R/W	066F FFF7$_{H}$	FFF9 4314$_{H}$	8/16/32	
		FSGD13PROT6	PBG13 protection register 6	R/W	066F FFF7$_{H}$	FFF9 4318$_{H}$	8/16/32	
		FSGD13PROT7	PBG13 protection register 7	R/W	066F FFF7$_{H}$	FFF9 431C$_{H}$	8/16/32	
PBG40	10	FSGD10PROT0	PBG10 protection register 0	R/W	066F FFF7$_{H}$	FFC5 9C00$_{H}$	8/16/32	ISO
		FSGD10PROT1	PBG10 protection register 1	R/W	066F FFF7$_{H}$	FFC5 9C04$_{H}$	8/16/32	

Table 40A.80 List of PBG/HBG Protection Registers

PBG/ HBG Group*1	Group No.*2	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG50	06	FSGD06PROT0	PBG06 protection register 0	R/W	066F FFF7$_H$	FFF9 0000$_H$	8/16/32	ISO
		FSGD06PROT1	PBG06 protection register 1	R/W	0647 FF77$_H$	FFF9 0004$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0008$_H$	—	
		—	Reserved	—	—	FFF9 000C$_H$	—	
		FSGD06PROT4	PBG06 protection register 4	R/W	066F FFF7$_H$	FFF9 0010$_H$	8/16/32	
		FSGD06PROT5	PBG06 protection register 5	R/W	066F FFF7$_H$	FFF9 0014$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0018$_H$	—	
		FSGD06PROT7	PBG06 protection register 7	R/W	066F FFF7$_H$	FFF9 001C$_H$	8/16/32	
		FSGD06PROT8	PBG06 protection register 8	R/W	066F FFF7$_H$	FFF9 0020$_H$	8/16/32	
PBG60	11	FSGD11PROT0	PBG11 protection register 0	R/W	066F FFF7$_H$	FFC7 1800$_H$	8/16/32	ISO
		FSGD11PROT1	PBG11 protection register 1	R/W	066F FFF7$_H$	FFC7 1804$_H$	8/16/32	
		FSGD11PROT2	PBG11 protection register 2	R/W	066F FFF7$_H$	FFC7 1808$_H$	8/16/32	
		FSGD11PROT3	PBG11 protection register 3	R/W	066F FFF7$_H$	FFC7 180C$_H$	8/16/32	
		FSGD11PROT4	PBG11 protection register 4	R/W	066F FFF7$_H$	FFC7 1810$_H$	8/16/32	
		FSGD11PROT5	PBG11 protection register 5	R/W	066F FFF7$_H$	FFC7 1814$_H$	8/16/32	
		FSGD11PROT6	PBG11 protection register 6	R/W	066F FFF7$_H$	FFC7 1818$_H$	8/16/32	
		FSGD11PROT7	PBG11 protection register 7	R/W	066F FFF7$_H$	FFC7 181C$_H$	8/16/32	
		FSGD11PROT8	PBG11 protection register 8	R/W	066F FFF7$_H$	FFC7 1820$_H$	8/16/32	
		FSGD11PROT9	PBG11 protection register 9	R/W	066F FFF7$_H$	FFC7 1824$_H$	8/16/32	
		FSGD11PROT10	PBG11 protection register 10	R/W	066F FFF7$_H$	FFC7 1828$_H$	8/16/32	
		FSGD11PROT11	PBG11 protection register 11	R/W	066F FFF7$_H$	FFC7 182C$_H$	8/16/32	
PBG61	14	FSGD14PROT0	PBG14 protection register 0	R/W	066F FFF7$_H$	FFC7 1900$_H$	8/16/32	ISO
		FSGD14PROT1	PBG14 protection register 1	R/W	066F FFF7$_H$	FFC7 1904$_H$	8/16/32	
		FSGD14PROT2	PBG14 protection register 2	R/W	066F FFF7$_H$	FFC7 1908$_H$	8/16/32	
		FSGD14PROT3	PBG14 protection register 3	R/W	066F FFF7$_H$	FFC7 190C$_H$	8/16/32	
		FSGD14PROT4	PBG14 protection register 4	R/W	066F FFF7$_H$	FFC7 1910$_H$	8/16/32	
		FSGD14PROT5	PBG14 protection register 5	R/W	066F FFF7$_H$	FFC7 1914$_H$	8/16/32	
		FSGD14PROT6	PBG14 protection register 6	R/W	066F FFF7$_H$	FFC7 1918$_H$	8/16/32	
		FSGD14PROT7	PBG14 protection register 7	R/W	066F FFF7$_H$	FFC7 191C$_H$	8/16/32	
HBG00	00	HFSGD00PROT0	HBG00 protection register 0	R/W	066F FFF7$_H$	FFF9 C000$_H$	8/16/32	ISO
		HFSGD00PROT1	HBG00 protection register 1	R/W	066F FFF7$_H$	FFF9 C004$_H$	8/16/32	
		HFSGD00PROT2	HBG00 protection register 2	R/W	066F FFF7$_H$	FFF9 C008$_H$	8/16/32	
		HFSGD00PROT3	HBG00 protection register 3	R/W	066F FFF7$_H$	FFF9 C00C$_H$	8/16/32	
		HFSGD00PROT4	HBG00 protection register 4	R/W	066F FFF7$_H$	FFF9 C010$_H$	8/16/32	
HBG01	01	HFSGD01PROT0	HBG01 protection register 0	R/W	066F FFF7$_H$	FFF9 C100$_H$	8/16/32	ISO
		HFSGD01PROT1	HBG01 protection register 1	R/W	066F FFF7$_H$	FFF9 C104$_H$	8/16/32	
HBG02	02	HFSGD02PROT0	HBG02 protection register 0	R/W	066F FFF7$_H$	FFF9 C200$_H$	8/16/32	ISO

Note 1. The Group indicates the module name.

Note 2. Register name index.

The following table lists the registers provided for each PBG/HBG group.

Table 40A.81 List of PBG/HBG Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGxx	ERRSLVxxCTL	PBGxx error control register	W	0000 0000_H	<base_addr0> + 0_H	8/16/32
	ERRSLVxxSTAT	PBGxx error status register	R	0000 0000_H	<base_addr0> + 4_H	8/16/32
	ERRSLVxxADDR	PBGxx error address register	R	0000 0000_H	<base_addr0> + 8_H	32
	ERRSLVxxTYPE	PBGxx error type register	R	0000 0000_H	<base_addr0> + C_H	16/32
HBGxx	HERRSLVxxCTL	HBGxx error control register	W	0000 0000_H	<base_addr1> + 0_H	8/16/32
	HERRSLVxxSTAT	HBGxx error status register	R	0000 0000_H	<base_addr1> + 4_H	8/16/32
	HERRSLVxxADDR	HBGxx error address register	R	0000 0000_H	<base_addr1> + 8_H	32
	HERRSLVxxTYPE	HBGxx error type register	R	0000 0000_H	<base_addr1> + C_H	16/32

In the above table, "xx" in the register names and symbols represents the PBG group numbers. The table below shows the base address values <base_addr0>, which correspond to each of the PBG group numbers and channel numbers as well as base address values <base_addr1>, which correspond to each of the HBG group numbers and channel numbers.

Table 40A.82 PBG Group Numbers and Error Base Addresses

PBG Group	PBG Group Number	<base_addr0>
PBG10	00	FFC4 0040_H
PBG11	01	FFC4 0140_H
PBG12	07	FFC4 0440_H
PBG13	08	FFC4 0540_H
PBG14	12	FFC4 0640_H
PBG20	02	FFDD $D040_H$
PBG21	09	FFDD $D140_H$
PBG30	03	FFF9 4040_H
PBG31	04	FFF9 4140_H
PBG32	05	FFF9 4240_H
PBG33	13	FFF9 4340_H
PBG40	10	FFC5 $9C40_H$
PBG50	06	FFF9 0040_H
PBG60	11	FFC7 1840_H
PBG61	14	FFC7 1940_H

Table 40A.83 HBG Group Numbers and Error Base Addresses

HBG Group	HBG Group Number	<base_addr1>
HBG00	00	FFF9 $C040_H$
HBG01	01	FFF9 $C140_H$
HBG02	02	FFF9 $C240_H$

40A.3.3.2 Details of Registers

(1) FSGDxxPROTn — PBGxx Protection Register n HFSGDxxPROTn — HBGxx Protection Register n

FSGDxxPROTn and HFSGDxxPROTn specify the access to be rejected for protecting the target peripheral circuit control registers. Any access that is disabled with any of the identifiers is rejected as an illegal access. "n" in the register names and symbols represents the PBG/HBG channel number.

Access:	FSGDxxPROTn and HFSGDxxPROTn can be read or written in 32-bit units. FSGDxxPROTnL, FSGDxxPROTnH, HFSGDxxPROTnL, and HFSGDxxPROTnH can be read or written in 16-bit units. FSGDxxPROTnLL, FSGDxxPROTnHL, FSGDxxPROTnHH, HFSGDxxPROTnLL, HFSGDxxPROTnHL, and HFSGDxxPROTnHH can be read or written in 8-bit units.
Address:	See **Table 40A.80, List of PBG/HBG Protection Registers**.
Value after reset:	See **Table 40A.80, List of PBG/HBG Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	—	—	—	PROTPEID4	—	PROTPEID2	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	*1	*1	*1	*1	*1	*1	*1	*1	1
R/W	R/W	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	*1	*1	*1	*1	1	0	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Note 1. It varies depending on each register. See **Section 40A.3.3.1, List of Registers**.

Table 40A.84 FSGDxxPROTn/HFSGxxPROTn Register Contents

Bit Position	Bit Name	Function
31	PROTLOCK	Register Lock 0: Enables FSGDxxPROTn/HFSGDxxPROTn rewrite. 1: Disables FSGDxxPROTn/HFSGDxxPROTn rewrite. When PROTLOCK is set to 1, the value is held until reset is asserted.
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PROTPEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	PROTPEID2	Access with PEID = 2 (CPU2)*1 0: Disables access with PEID2. 1: Enables access with PEID2.
18	PROTPEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40A.84 FSGDxxPROTn/HFSGxxPROTn Register Contents

Bit Position	Bit Name	Function
8	PROTSPID3	Access with SPID = 3*2 0: Disables access with SPID3. 1: Enables access with SPID3.
7	PROTSPID2	Access with SPID = 2 (CPU2*3)*2 0: Disables access with SPID2. 1: Enables access with SPID2.
6	PROTSPID1	Access with SPID = 1 (CPU1*3)*2 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*2 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVxxCTL — PBGxx Error Control Register HERRSLVxxCTL — HBGxx Error Control Register

ERRSLVxxCTL and HERRSLVxxCTL clear the status in the error status register with the PBGxx/HBGxx.

Access: ERRSLVxxCTL and HERRSLVxxCTL are write-only registers that can be written in 32-bit units.
ERRSLVxxCTLL and HERRSLVxxCTLL are write-only registers that can be written in 16-bit units.
ERRSLVxxCTLLL and HERRSLVxxCTLLL are write-only registers that can be written in 8-bit units.

Address: ERRSLVxxCTL: <base_addr0> + 0_H
ERRSLVxxCTLL: <base_addr0> + 0_H
ERRSLVxxCTLLL: <base_addr0> + 0_H
HERRSLVxxCTL: <base_addr1> + 0_H
HERRSLVxxCTLL: <base_addr1> + 0_H
HERRSLVxxCTLLL: <base_addr1> + 0_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40A.85 ERRSLVxxCTL/HERRSLVxxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40A.86 CLRO and CLRE in ERRSLVxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVxxSTAT — PBGxx Error Status Register
HERRSLVxxSTAT — HBGxx Error Status Register

ERRSLVxxSTAT and HERRSLVxxSTAT hold the status of the illegal access rejected with the PBGxx/HBGxx.

Access: ERRSLVxxSTAT and HERRSLVxxSTAT are read-only registers that can be read in 32-bit units.
ERRSLVxxSTATL and HERRSLVxxSTATL are read-only registers that can be read in 16-bit units.
ERRSLVxxSTATLL and HERRSLVxxSTATLL are read-only registers that can be read in 8-bit units.

Address: ERRSLVxxSTAT: <base_addr0> + 4_H
ERRSLVxxSTATL: <base_addr0> + 4_H
ERRSLVxxSTATLL: <base_addr0> + 4_H
HERRSLVxxSTAT: <base_addr1> + 4_H
HERRSLVxxSTATL: <base_addr1> + 4_H
HERRSLVxxSTATLL: <base_addr1> + 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.87 ERRSLVxxSTAT/HERRSLVxxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBG/HBG error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBG/HBG protection violation 1: PBG/HBG protection violation occurred.

CAUTION

After writing HERRSLVxxCTL, the value of HERRSLVxxSTAT takes the following procedure for a reflection:

- Execute SYNCM
- Wait for 24 × CPU clock*1 cycles

Note 1. CPU clock: Clock selected by CKSC_CPUCLKS_CTRL and CKSC_CPUCLKD_CTL

(4) ERRSLVxxADDR — PBGxx Error Address Register
HERRSLVxxADDR — HBGxx Error Address Register

ERRSLVxxADDR holds the address of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.
HERRSLVxxADDR holds the address of the illegal access rejected with the HBGxx. The register is not updated when corresponding HERRSLVxxSTAT.ERR is 1.

Access:	ERRSLVxxADDR and HERRSLVxxADDR are read-only registers that can be read in 32-bit units.
Address:	ERRSLVxxADDR: <base_addr0> + 8_H HERRSLVxxADDR: <base_addr1> + 8_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.88 ERRSLVxxADDR/HERRSLVxxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBG/HBG protection violation is generated.
1, 0	Reserved	ERRSLVxxADDR: When read, the value after reset is returned. HERRSLVxxADDR: When read, an undefined value is returned.

(5) ERRSLVxxTYPE — PBGxx Error Type Register
HERRSLVxxTYPE — HBGxx Error Type Register

ERRSLVxxTYPE holds the type of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.

HERRSLVxxTYPE holds the type of the illegal access rejected with the HBGxx. The register is not updated when corresponding HERRSLVxxSTAT.ERR is 1.

Access: ERRSLVxxTYPE and HERRSLVxxTYPE are read-only registers that can be read in 32-bit units.
ERRSLVxxTYPEL and HERRSLVxxTYPEL are read-only registers that can be read in 16-bit units.

Address: ERRSLVxxTYPE: <base_addr0> + C_H
ERRSLVxxTYPEL: <base_addr0> + C_H
HERRSLVxxTYPE: <base_addr1> + C_H
HERRSLVxxTYPEL: <base_addr1> + C_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.89 ERRSLVxxTYPE/HERRSLVxxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBG/HBG protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBG/HBG protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBG/HBG protection violation is generated.
5	Reserved	When read, the value after reset is returned.
4 to 1	Reserved	These bits are read as an undefined value.
0	WRITE	This bit is set to 1 when the access that has generated the PBG/HBG protection violation is a write access.

40A.3.4 PBG for CPU System

The PBGC module is divided into two PBGC groups, PBGC0 and PBGC1. PBGC0 group contains protection registers for INTC2 and DMA functions. PBGC1 group contains protection registers for ECC control function etc. Each PBGC group holds the information of the access that has been rejected.

The following table lists the target registers to be protected and the corresponding PBGC group names.

Table 40A.90 Target Registers of PBG for CPU Subsystem

PBGC Group	Group No.	Channel Number	Protection Target Module	Target Register
PBGC0	C0 (PBC group 0 for CPU system)	0	INTC2	ICxxx, IBDxxx (xxx =32 to 381) IMRm (m = 1 to 11) (described in **Section 7A, Exception/Interrupts of RH850/F1KH-D8**)
		1	PDMA0	All registers inside DMA controller (described in **Section 8, DMA Controller**)
		2	PDMA1	All registers inside DMA controller (described in **Section 8, DMA Controller**)

Table 40A.90 Target Registers of PBG for CPU Subsystem

PBGC Group	Group No.	Channel Number	Protection Target Module	Target Register
PBGC1	C1 (PBC group 1 for CPU system)	0	Flash memory (Programing function)	BFASELR*1
		1	Code flash ECC control register (VCI)	CFECCCTL_VCI CFERRINT_VCI CFSTCLR_VCI CFOVFSTR_VCI CF1STERSTR_VCI CF1STEADR0_VCI CFSTSTCTL_VCI
		2	Code flash ECC control register (PE1)	CFECCCTL_PE1 CFERRINT_PE1 CFSTCLR_PE1 CFOVFSTR_PE1 CF1STERSTR_PE1 CF1STEADR0_PE1
		3	Local RAM ECC control register (PE1)	LRTSTCTL_PE1 LRTDATBF0_PE1 LRECCCTL_PE1 LRERRINT_PE1 LRSTCLR_PE1 LROVFSTR_PE1 LR1STERSTR_PE1 LR1STEADR0_PE1
		4	System control module	BOOTCTRL (described in **Section 3A, CPU System of RH850/F1KH-D8**) EPC (described in **Section 43, On-Chip Debug Unit (OCD)**)
		5	Buffer controller	FBUFCCTL (described in **Section 3A, CPU System of RH850/F1KH-D8**)
		6	Global RAM ECC Control register (bank A)	GRECCCTL_BKA GRERRINT_BKA GRSTCLR_BKA GROVFSTR_BKA GR1STERSTR_BKA GR1STEADR_BKA GRTSTCTL_BKA GRDECINBF1_BKA
		7	Global RAM ECC Control register (bank B)	GRECCCTL_BKB GRERRINT_BKB GRSTCLR_BKB GROVFSTR_BKB GR1STERSTR_BKB GR1STEADR_BKB GRTSTCTL_BKB GRDECINBF1_BKB
		8	Code flash ECC control register (PE2)	CFECCCTL_PE2 CFERRINT_PE2 CFSTCLR_PE2 CFOVFSTR_PE2 CF1STERSTR_PE2 CF1STEADR0_PE2
		9	Local RAM ECC control register (PE2)	LRTSTCTL_PE2 LRTDATBF0_PE2 LRECCCTL_PE2 LRERRINT_PE2 LRSTCLR_PE2 LROVFSTR_PE2 LR1STERSTR_PE2 LR1STEADR0_PE2

Note 1. Regarding the PBGC registers for the flash memory, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

40A.3.4.1 List of Registers

The following table lists the registers provided for each PBGC group. And PBG group is equal to module name.

Table 40A.91 List of PBGC Protection Registers

PBG Group*1	Group No.	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBGC0	C0	FSGDC0PROT0	PBGC0 protection register 0	R/W	07FF FFFF$_H$	FFC4 C000$_H$	8/16/32	ISO
		FSGDC0PROT1	PBGC0 protection register 1	R/W	07FF FFFF$_H$	FFC4 C004$_H$	8/16/32	
		FSGDC0PROT2	PBGC0 protection register 2	R/W	07FF FFFF$_H$	FFC4 C008$_H$	8/16/32	
PBGC1	C1	FSGDC1PROT0	PBGC1 protection register 0	R/W	07FF FFFF$_H$	FFC4 C120$_H$	8/16/32	ISO
		FSGDC1PROT1	PBGC1 protection register 1	R/W	07FF FFFF$_H$	FFC4 C124$_H$	8/16/32	
		FSGDC1PROT2	PBGC1 protection register 2	R/W	07FF FFFF$_H$	FFC4 C128$_H$	8/16/32	
		FSGDC1PROT3	PBGC1 protection register 3	R/W	07FF FFFF$_H$	FFC4 C12C$_H$	8/16/32	
		FSGDC1PROT4	PBGC1 protection register 4	R/W	07FF FFFF$_H$	FFC4 C130$_H$	8/16/32	
		FSGDC1PROT5	PBGC1 protection register 5	R/W	07FF FFFF$_H$	FFC4 C134$_H$	8/16/32	
		FSGDC1PROT6	PBGC1 protection register 6	R/W	07FF FFFF$_H$	FFC4 C118$_H$	8/16/32	
		FSGDC1PROT7	PBGC1 protection register 7	R/W	07FF FFFF$_H$	FFC4 C11C$_H$	8/16/32	
		FSGDC1PROT8	PBGC1 protection register 8	R/W	07FF FFFF$_H$	FFC4 C138$_H$	8/16/32	
		FSGDC1PROT9	PBGC1 protection register 9	R/W	07FF FFFF$_H$	FFC4 C13C$_H$	8/16/32	

Note 1. The Group indicates the module name.

Table 40A.92 List of PBGC Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGC0	ERRSLVC0CTL	PBGC0 error control register	W	0000 0000$_H$	FFC4 C800$_H$	8/16/32
	ERRSLVC0STAT	PBGC0 error status register	R	0000 0000$_H$	FFC4 C804$_H$	8/16/32
	ERRSLVC0ADDR	PBGC0 error address register	R	0000 0000$_H$	FFC4 C808$_H$	32
	ERRSLVC0TYPE	PBGC0 error type register	R	0000 0000$_H$	FFC4 C80C$_H$	16/32
PBGC1	ERRSLVC1CTL	PBGC1 error control register	W	0000 0000$_H$	FFC4 C900$_H$	8/16/32
	ERRSLVC1STAT	PBGC1 error status register	R	0000 0000$_H$	FFC4 C904$_H$	8/16/32
	ERRSLVC1ADDR	PBGC1 error address register	R	0000 0000$_H$	FFC4 C908$_H$	32
	ERRSLVC1TYPE	PBGC1 error type register	R	0000 0000$_H$	FFC4 C90C$_H$	16/32

40A.3.4.2 Details of Registers

(1) FSGDCxPROTn — PBGCx Protection Register n (x = 0, 1)

FSGDCxPROTn specifies the access to be rejected for protecting the target registers. Any access that is disabled with any of the identifiers is rejected as an illegal access.

"n" in the register names and symbols represents the PBGC channel number.

Access: FSGDCxPROTn register can be read or written in 32-bit units.
FSGDCxPROTnL and FSGDCxPROTnH registers can be read or written in 16-bit units.
FSGDCxPROTnLL, FSGDCxPROTnHL, and FSGDCxPROTnHH registers can be read or written in 8-bit units.

Address: See **Table 40A.91, List of PBGC Protection Registers**.

Value after reset: See **Table 40A.91, List of PBGC Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	—	—	—	PROTPEID4	—	PROTPEID2	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 40A.93 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
31	PROTLOCK	Lock of write to this register 0: Register can be re-written. 1: Any further write to this register is ignored. This bit can be cleared by RESET.
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PROTPEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	PROTPEID2	Access with PEID = 2 (CPU2)*1 0: Disables access with PEID2. 1: Enables access with PEID2.
18	PROTPEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PROTSPID3	Access with SPID = 3*2 0: Disables access with SPID3. 1: Enables access with SPID3.

Table 40A.93 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
7	PROTSPID2	Access with SPID = 2 (CPU2*[3])*[2] 0: Disables access with SPID2. 1: Enables access with SPID2.
6	PROTSPID1	Access with SPID = 1 (CPU1*[3])*[2] 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*[2] 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVCxCTL — PBGCx Error Control Register (x = 0, 1)

ERRSLVCxCTL clears the status in error status register PBGCx.

Access:	ERRSLVCxCTL register is a write-only register that can be written in 32-bit units. ERRSLVCxCTLL register is a write-only register that can be written in 16-bit units. ERRSLVCxCTLLL register is a write-only register that can be written in 8-bit units.
Address:	ERRSLVCxCTL: FFC4 C800$_H$ + (100$_H$ × x) ERRSLVCxCTLL: FFC4 C800$_H$ + (100$_H$ × x) ERRSLVCxCTLLL: FFC4 C800$_H$ + (100$_H$ × x)
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40A.94 ERRSLVCxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40A.95 CLRO and CLRE in ERRSLVCxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVCxSTAT — PBGCx Error Status Register (x = 0, 1)

ERRSLVCxSTAT holds the status of the illegal access rejected with the PBGCx.

Access: ERRSLVCxSTAT register is a read-only register that can be read in 32-bit units.
ERRSLVCxSTATL register is a read-only register that can be read in 16-bit units.
ERRSLVCxSTATLL register is a read-only register that can be read in 8-bit units.

Address: ERRSLVCxSTAT: FFC4 C804$_H$ + (100$_H$ × x)
ERRSLVCxSTATL: FFC4 C804$_H$ + (100$_H$ × x)
ERRSLVCxSTATLL: FFC4 C804$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.96 ERRSLVCxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBGC error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBGC protection violation 1: A PBGC protection violation occurred.

(4) ERRSLVCxADDR — PBGCx Error Address Register (x = 0, 1)

ERRSLVCxADDR holds the address of the illegal access rejected with the PBGCx.

The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access: ERRSLVCxADDR register is a read-only register that can be read in 32-bit units.

Address: FFC4 C808$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.97 ERRSLVCxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBGC protection violation is generated.
1, 0	Reserved	When read, the value after reset is returned.

(5) ERRSLVCxTYPE — PBGCx Error Type Register (x = 0, 1)

ERRSLVCxTYPE holds the type of the illegal access rejected with the PBGCx.

The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access: ERRSLVCxTYPE register is a read-only register that can be read in 32-bit units.
ERRSLVCxTYPEL register is a read-only register that can be read in 16-bit units.

Address: ERRSLVCxTYPE: FFC4 C80C_H + (100_H × x)
ERRSLVCxTYPEL: FFC4 C80C_H + (100_H × x)

Value after reset: $\text{0000 0000}_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40A.98 ERRSLVCxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBGC protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBGC protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBGC protection violation is generated.
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	This bit is set to 1 when an access that has generated the PBGC protection violation is the write.

Section 40B Functional Safety of RH850/F1KM-S4

This section provides an overview of the safety mechanisms included in the RH850/F1KM Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance with ISO26262.

For more information about the development process and safety mechanisms, please contact our sales office.

The following are the failure detection functions provided by this microcontroller.

40B.1 Overview

ECC

Detects failures of memories and data transfer paths and corrects some types of failures.

Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements from erroneous access.

Clock Monitor

Monitors the clock operation to detect abnormal operations.

For details, see **Section 13, Clock Monitor (CLMA)**.

Data CRC

Generates CRC to detect data errors.

For details, see **Section 41, Data CRC (DCRA)**.

Write-Protected Registers

The write-protected registers are protected from inadvertent write access due to erroneous program execution.

For details, see **Section 5, Write-Protected Registers**.

40B.2 ECC

40B.2.1 Overview

This product incorporates an ECC for the following memories. The ECC enables detection and correction of errors of the data retained in the memories. The ECC also enables detection and correction of errors produced between the ECC encoder and memories and between memories and ECC decoder.

Table 40B.1 ECC Overview

Applicable Memory		Applicable Data Width [bits]	Operation upon Error Detection					Failure Insertion
			Detection/ Correction	SYSERR	Interrupt Notice	Error Status	Address Capture	
Code flash		128	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Data flash		32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
Local RAM (CPU1/Self)		32	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Global RAM Retention RAM		32	SEC-DED	SED/DED*[2]	SEC-DED	Possible	Possible	Possible
Peripheral RAM*[1]	CSIH	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	RS-CANFD	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	FlexRay	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	Ethernet	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible

Note 1. For details of ECC for each peripheral IP, see **Section 40B.2.6, ECC for Peripheral RAM**.

Note 2. For details, see **Section 40B.2.2.2, Interrupt Requests, Section 40B.2.4.2, Interrupt Requests**, and **Section 40B.2.5.2, Interrupt Requests**.

Applicable Data Width

This is the data width to be ECC encoded.

To write data with a smaller data width than shown, the following processing is required. ECC is also performed for a read in (1).

(1) Reading data to be ECC-encoded including data to be rewritten

(2) Replacing data to be rewritten

(3) Writing back data generated in (2)

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected.

SED-DED: 1-bit errors and 2-bit errors can only be detected.

SYSERR

SYSERR can be generated upon error detection.

Interrupt Notice

An interrupt can be generated upon error detection.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

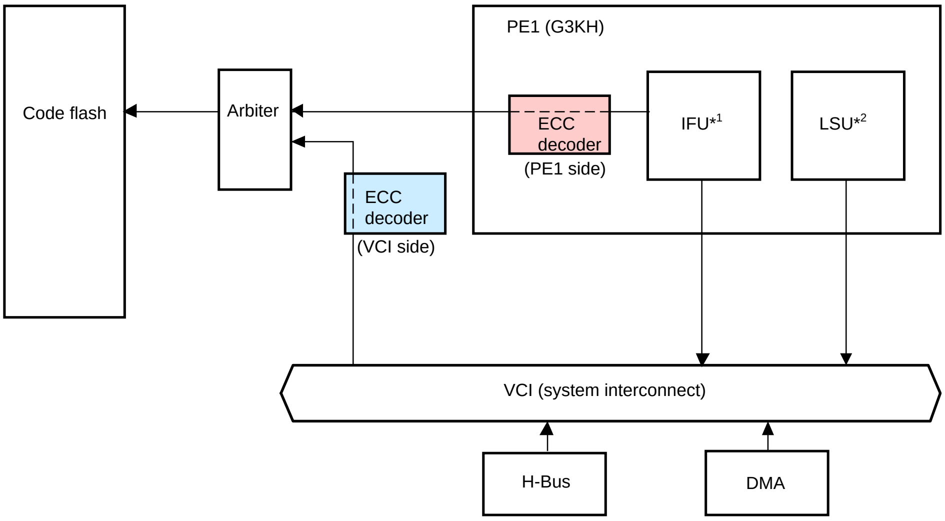
Self-diagnosis of the ECC decoder error notification function can be performed by using an intentionally generated ECC error.

40B.2.2 Code Flash ECC

40B.2.2.1 Overview

RH850/F1KM has two code flash ECC decoder circuits implemented, one inside the Processor Element (PE1) and one on the VCI (system interconnect).

Figure 40B.1, Block Diagram of Code Flash ECC shows the location of two ECC decoders for the code flash.



Note 1. Instruction Fetch Unit

Note 2. Load Store Unit

Figure 40B.1 Block Diagram of Code Flash ECC

The code flash ECC is summarized in the table below.

Table 40B.2 Summary of Code Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. (This is not supported in CFERRINT_PE1 register.) Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3BC, CPU System of RH850/F1KM**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.

Table 40B.2 Summary of Code Flash ECC

Item	Description
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The ECC 1-bit error status is set only when no error status has been set. The ECC 2-bit error status is set even when the ECC 1-bit error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error is also captured.
Self-diagnosis	The ECC bit can be read directly.
Inhibiting instruction execution	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching avoids the execution of incorrect instructions.

40B.2.2.2 Interrupt Requests

Interrupt requests for code flash ECC are listed below.

Table 40B.3 Code Flash ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0*[1] (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR	—

Note 1. ECCFLI0FEIF flag set can be read (not jump to handler address of FEINT) in SYSERR processing because return from SYSERR is not possible.

Table 40B.4 Code Flash ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0 (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR, INTECCFLI0	—

Table 40B.5 Code Flash ECC Interrupt Requests (During Bus Master Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	INTECCFLI0	—
—	ECC 2-bit error interrupt of code flash	INTECCFLI0	—

40B.2.2.3 List of Registers

Table 40B.6 List of Registers

Module Name	Register Name	Symbol*1	Address
CFECC_VCI	Code flash ECC control register (VCI)	CFECCCTL_VCI	FFC6 2200$_H$
	Code flash error information control register (VCI)	CFERRINT_VCI	FFC6 2204$_H$
	Code flash status clear register (VCI)	CFSTCLR_VCI	FFC6 2208$_H$
	Code flash error count overflow status register (VCI)	CFOVFSTR_VCI	FFC6 220C$_H$
	Code flash 1st error status register (VCI)	CF1STERSTR_VCI	FFC6 2210$_H$
	Code flash 1st error address register (VCI)	CF1STEADR0_VCI	FFC6 2250$_H$
CFECC_CPU1	Code flash ECC control register (PE1)	CFECCCTL_PE1	FFC6 2400$_H$
	Code flash error information control register (PE1)	CFERRINT_PE1	FFC6 2404$_H$
	Code flash status clear register (PE1)	CFSTCLR_PE1	FFC6 2408$_H$
	Code flash error count overflow status register (PE1)	CFOVFSTR_PE1	FFC6 240C$_H$
	Code flash 1st error status register (PE1)	CF1STERSTR_PE1	FFC6 2410$_H$
	Code flash 1st error address register (PE1)	CF1STEADR0_PE1	FFC6 2450$_H$
CFECC_VCI	Code flash sub-test control register (VCI)	CFSTSTCTL_VCI	FFC6 2350$_H$

Note 1. The registers suffixed with symbols "_VCI" and "_PE1" are provided to ECC controllers corresponding to each access port: registers with "_VCI" are provided to the ECC controller for data access from the system interconnect to the code flash and registers with "_PE1" are provided to the ECC controller for fetch access from the CPU1.

40B.2.2.4 Details of Registers

(1) CFECCCTL_VCI/PE1 — Code Flash ECC Control Register

CFECCCTL_VCI/PE1 enables or disables ECC error detection and correction and 1-bit error correction. When writing to CFECCCTL_VCI/PE1, PROT1 and PROT0 need to be 01_B.

Access: CFECCCTL_VCI and CFECCCTL_PE1 can be read or written in 32-bit units.
CFECCCTL_VCIL and CFECCCTL_PE1L can be read or written in 16-bit units.

Address: CFECCCTL_VCI: FFC6 2200_H
CFECCCTL_VCIL: FFC6 2200_H
CFECCCTL_PE1: FFC6 2400_H
CFECCCTL_PE1L: FFC6 2400_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.7 CFECCCTL_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFECCCTL_VCI/PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) CFERRINT_VCI/PE1 — Code Flash Error Information Control Register

CFERRINT_VCI/PE1 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: CFERRINT_VCI and CFERRINT_PE1 can be read or written in 32-bit units.
CFERRINT_VCIL and CFERRINT_PE1L can be read or written in 16-bit units.
CFERRINT_VCILL and CFERRINT_PE1LL can be read or written in 8-bit units.

Address: CFERRINT_VCI: FFC6 2204$_H$
CFERRINT_VCIL: FFC6 2204$_H$
CFERRINT_VCILL: FFC6 2204$_H$
CFERRINT_PE1: FFC6 2404$_H$
CFERRINT_PE1L: FFC6 2404$_H$
CFERRINT_PE1LL: FFC6 2404$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE *1	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.8 CFERRINT_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE*[1]	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

Note 1. This bit is not supported in CFERRINT_PE1. When writing to this bit in CFERRINT_PE1, always write 1.

Note 2. Regarding the interrupt request, see **Section 40B.2.2.2, Interrupt Requests**.

(3) CFSTCLR_VCI/PE1 — Code Flash Status Clear Register

CFSTCLR_VCI/PE1 clears the error flags in the error status register (CF1STERSTR_VCI/PE1), the overflow flag in the error overflow status register (CFOVFSTR_VCI/PE1), and the error address register (CF1STEADR0_VCI/PE1).

Access: CFSTCLR_VCI and CFSTCLR_PE1 are write-only registers that can be written in 32-bit units.
CFSTCLR_VCIL and CFSTCLR_PE1L are write-only registers that can be written in 16-bit units.
CFSTCLR_VCILL and CFSTCLR_PE1LL are write-only registers that can be written in 8-bit units.

Address: CFSTCLR_VCI: FFC6 2208$_H$
CFSTCLR_VCIL: FFC6 2208$_H$
CFSTCLR_VCILL: FFC6 2208$_H$
CFSTCLR_PE1: FFC6 2408$_H$
CFSTCLR_PE1L: FFC6 2408$_H$
CFSTCLR_PE1LL: FFC6 2408$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40B.9 CFSTCLR_VCI/_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Clear 0: No effect (Setting 0 does not affect the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1; ERROVF0 flag in CFOVFSTR_VCI/PE1; and CF1STEADR0_VCI/PE1.) 1: Writing 1 to this bit clears the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1; ERROVF0 flag in CFOVFSTR_VCI/PE1; and CF1STEADR0_VCI/PE1.

(4) CFOVFSTR_VCI/PE1 — Code Flash Error Count Overflow Status Register

CFOVFSTR_VCI/PE1 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

Access:	CFOVFSTR_VCI and CFOVFSTR_PE1 are read-only registers that can be read in 32-bit units. CFOVFSTR_VCIL and CFOVFSTR_PE1L are read-only registers that can be read in 16-bit units. CFOVFSTR_VCILL and CFOVFSTR_PE1LL are read-only registers that can be read in 8-bit units.
Address:	CFOVFSTR_VCI: FFC6 220C$_H$ CFOVFSTR_VCIL: FFC6 220C$_H$ CFOVFSTR_VCILL: FFC6 220C$_H$ CFOVFSTR_PE1: FFC6 240C$_H$ CFOVFSTR_PE1L: FFC6 240C$_H$ CFOVFSTR_PE1LL: FFC6 240C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.10 CFOVFSTR_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 shows whether a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register has occurred, except when both of the error address and source of the second error are the same as those of the first error. 0: Did not occur. 1: Occurred. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1.

(5) CF1STERSTR_VCI/PE1 — Code Flash 1st Error Status Register

CF1STERSTR_VCI/PE1 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0. If a 2-bit ECC error occurs while the 1-bit ECC error flag is set, the 2-bit ECC error flag is set while retaining the 1-bit ECC error flag.

Access: CF1STERSTR_VCI and CF1STERSTR_PE1 are read-only registers that can be read in 32-bit units.
CF1STERSTR_VCIL and CF1STERSTR_PE1L are read-only registers that can be read in 16-bit units.
CF1STERSTR_VCILL and CF1STERSTR_PE1LL are read-only registers that can be read in 8-bit units.

Address: CF1STERSTR_VCI: FFC6 2210$_H$
CF1STERSTR_VCIL: FFC6 2210$_H$
CF1STERSTR_VCILL: FFC6 2210$_H$
CF1STERSTR_PE1: FFC6 2410$_H$
CF1STERSTR_PE1L: FFC6 2410$_H$
CF1STERSTR_PE1LL: FFC6 2410$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.11 CF1STERSTR_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1. [Setting condition] ECC 2-bit error is detected when DEDF0 is 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected when DEDF0 flag is 0. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1. [Setting condition] ECC 1-bit error is detected when DEDF0, SEDF0 are 0.

(6) CF1STEADR0_VCI/PE1 — Code Flash 1st Error Address Register

CF1STEADR0_VCI/PE1 holds the address at which an error has occurred.

The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated.

In addition, the EADR[24:4] bits in this register correspond to the [24:4] bits of the real address. The real address can be calculated by adding the upper address [31:25] bits, to which code flash is mapped, as a base address.

The CF1STEADR0_VCI/PE1 register is cleared by an internal reset, the external reset, or by setting the STCLR bit in the CFSTCLR_VCI/PE1 register to 1.

Access: CF1STEADR0_VCI and CF1STEADR0_PE1 are read-only registers that can be read in 32-bit units.
CF1STEADR0_VCIL, CF1STEADR0_VCIH, CF1STEADR0_PE1L and CF1STEADR0_PE1H are read-only registers that can be read in 16-bit units.
CF1STEADR0_VCILL, CF1STEADR0_VCILH, CF1STEADR0_VCIHL, CF1STEADR0_VCIHH, CF1STEADR0_PE1LL, CF1STEADR0_PE1LH, CF1STEADR0_PE1HL and CF1STEADR0_PE1HH are read-only registers that can be read in 8-bit units.

Address: CF1STEADR0_VCI: FFC6 2250$_H$
CF1STEADR0_VCIL: FFC6 2250$_H$
CF1STEADR0_VCIH: FFC6 2252$_H$
CF1STEADR0_VCILL: FFC6 2250$_H$
CF1STEADR0_VCILH: FFC6 2251$_H$
CF1STEADR0_VCIHL: FFC6 2252$_H$
CF1STEADR0_VCIHH: FFC6 2253$_H$
CF1STEADR0_PE1: FFC6 2450$_H$
CF1STEADR0_PE1L: FFC6 2450$_H$
CF1STEADR0_PE1H: FFC6 2452$_H$
CF1STEADR0_PE1LL: FFC6 2450$_H$
CF1STEADR0_PE1LH: FFC6 2451$_H$
CF1STEADR0_PE1HL: FFC6 2452$_H$
CF1STEADR0_PE1HH: FFC6 2453$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	EADR[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.12 CF1STEADR0_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	EADR[24:4]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1.
3 to 0	Reserved	When read, the value after reset is returned.

(7) CFSTSTCTL_VCI — Code Flash Sub-Test Control Register

CFSTSTCTL_VCI is used for the ECC test (self-diagnosis). This register is dedicated for code flash. After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read directly.

When writing to CFSTSTCTL_VCI, PROT1 and PROT0 need to be 01_B.

Access:	CFSTSTCTL_VCI can be read or written in 32-bit units. CFSTSTCTL_VCIL can be read or written in 16-bit units.
Address:	CFSTSTCTL_VCI: FFC6 2350_H CFSTSTCTL_VCIL: FFC6 2350_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40B.13 CFSTSTCTL_VCI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFSTSTCTL_VCI.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, ECC bits can be read directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3BC, CPU System of RH850/F1KM**.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n address. The results of reading code flash are as follows:

Table 40B.14 Results of Reading Code Flash

Bit Number	Meaning	Bit Position	Description
bit[31:10]	all-0	31 to 10	These bits are always 0.
bit[9]	reserved	9	Unknown
bit[8:0]	ECC bits	8 to 0	ECC bits

40B.2.3 Data Flash ECC

40B.2.3.1 Overview

The data flash ECC is summarized in the table below.

Table 40B.15 Summary of Data Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection / correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit errors is selectable. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit error is selectable. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the Interrupt Controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. ECC error address is captured when both SEDF and DEDF of DFERSTR are 0.
Self-diagnosis	ROM data and the ECC bit can be read directly. Arbitrary data can be written to ROM data and the ECC bit.

40B.2.3.2 Interrupt Requests

The interrupt requests for data flash ECC are shown below.

Table 40B.16 Data Flash ECC Interrupt Requests (During Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of data flash	INTECCEEP0	—
—	ECC 2-bit error interrupt of data flash	INTECCEEP0	—

40B.2.3.3 List of Registers

Table 40B.17 List of Registers

Module Name	Register Name	Symbol	Address
DFECC	Data flash ECC control register	DFECCCTL	FFC6 2A00$_H$
	Data flash error status register	DFERSTR	FFC6 2A04$_H$
	Data flash error status clear register	DFERSTC	FFC6 2A08$_H$
	Data flash error overflow status register	DFOVFSTR	FFC6 2A0C$_H$
	Data flash error overflow status clear register	DFOVFSTC	FFC6 2A10$_H$
	Data flash error notification control register	DFERRINT	FFC6 2A14$_H$
	Data flash 1st error address register	DFEADR	FFC6 2A18$_H$
	Data flash test control register	DFTSTCTL	FFC6 2A1C$_H$

40B.2.3.4 Details of Registers

(1) DFECCCTL — Data Flash ECC Control Register

DFECCCTL enables or disables ECC error detection and correction and 1-bit error correction. When writing to DFECCCTL, PROT1 and PROT0 need to be 01_B.

Access: DFECCCTL can be read or written in 16-bit units.

Address: DFECCCTL: FFC6 $2A00_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.18 DFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCCTL.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection and correction are enabled, and the DEDF bit is set if an ECC 2-bit error is detected.

Access: DFERSTR is a read-only register that can be read in 8-bit units.

Address: DFERSTR: FFC6 2A04$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40B.19 DFERSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 2-bit error is detected when both SEDF and DEDF are 0.
0	SEDF	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 1-bit error is detected when both SEDF and DEDF are 0.

(3) DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the data flash error status register.

Access: DFERSTC is a write-only register that can be written in 8-bit units.

Address: DFERSTC: FFC6 2A08$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERRCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40B.20 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

(4) DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of data flash error overflow.

Access: DFOVFSTR is a read-only register that can be read in 8-bit units.

Address: DFOVFSTR: FFC6 2A0C$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40B.21 DFOVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag ERROVF is set if an ECC error occurs while the error address register is full. 0: Did not occurred. 1: Occurred. [Clearing condition] Set the ERROVFCLR bit is set in data flash error overflow status clear register.

(5) DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the data flash error overflow flag.

Access: DFOVFSTC is a write-only register that can be written in 8-bit units.

Address: DFOVFSTC: FFC6 2A10$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVFCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40B.22 DFOVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the ERROVF flag in DFOVFSTR.) 1: The ERROVF flag in DFOVFSTR is cleared.

(6) DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: DFERRINT can be read or written in 8-bit units.

Address: DFERRINT: FFC6 2A14$_H$

Value after reset: 02$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 40B.23 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(7) DFEADR — Data Flash 1st Error Address Register

DFEADR holds the address at which an ECC error has occurred while both of the SEDF and DEDF bits in the data flash error status register are 0.

Access: DFEADR is a read-only register that can be read in 32-bit units.

Address: DFEADR: FFC6 2A18$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	DFEADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DFEADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.24 DFEADR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 2	DFEADR[20:2]	ECC Error Address DFEADR is read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address FF20 0000$_H$.
1, 0	Reserved	When read, the value after reset is returned.

(8) DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for the ECC test.

After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read.

When writing to DFTSTCTL, PROT1 and PROT0 need to be 01_B.

Access: DFTSTCTL can be read or written in 16-bit units.

Address: DFTSTCTL: FFC6 2A1C$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40B.25 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFTSTCTL.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Sets ECC test mode. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

40B.2.4 Local RAM ECC

40B.2.4.1 Overview

The local RAM ECC of CPU1 is summarized in the table below.

Table 40B.26 Summary of Local RAM ECC (CPU1)

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3BC, CPU System of RH850/F1KM**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no error status has been set, the address at which the first error occurred is captured. In addition, when the retained address source is an ECC 1-bit error or ECC 2-bit error, the address is also captured.
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data and the ECC bit can be read directly.
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

CAUTION

When ECC error detection/correction for the local RAM is enabled for access, initialize the RAM with the 32-bit length of RAM access before the RAM is used. If the RAM before initialization is read, an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

Moreover, if the RAM is not initialized with the 32-bit length (for example, initialized with 8- or 16-bit length of access), an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

40B.2.4.2 Interrupt Requests

The local RAM ECC interrupt requests are listed below.

Table 40B.27 Local RAM ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40B.28 Local RAM ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40B.29 Local RAM ECC Interrupt Requests (During Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	INTECCRAM	—

40B.2.4.3 List of Registers

Table 40B.30 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
LRTST	FFC6 5004_H	LRTSTCTL_PE1	Local RAM test control register (PE1)	R/W	0000 0000_H	16/32
	FFC6 5008_H	LRTDATBF0_PE1	Local RAM test data read buffer 0 (PE1)	R	0000 0000_H	32
LRECC	FFC6 5400_H	LRECCCTL_PE1	Local RAM ECC control register (PE1)	R/W	0000 0000_H	16/32
	FFC6 5404_H	LRERRINT_PE1	Local RAM error information control register (PE1)	R/W	0000 0003_H	8/16/32
	FFC6 5408_H	LRSTCLR_PE1	Local RAM status clear register (PE1)	W	0000 0000_H	8/16/32
	FFC6 $540C_H$	LROVFSTR_PE1	Local RAM error count overflow status register (PE1)	R	0000 0000_H	8/16/32
	FFC6 5410_H	LR1STERSTR_PE1	Local RAM 1st error status register (PE1)	R	0000 0000_H	8/16/32
	FFC6 5450_H	LR1STEADR0_PE1	Local RAM 1st error address register 0 (PE1)	R	0000 0000_H	8/16/32

40B.2.4.4 Details of Registers

(1) LRTSTCTL_PE1 — Local RAM Test Control Register

LRTSTCTL_PE1 is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST = 1, any data can be written to the ECC bits. The DATSEL bit is used to select RAM data or the ECC bits.

When writing to LRTSTCTL_PE1, PROT1 and PROT0 need to be 01_B.

Access: LRTSTCTL_PE1 register can be read or written in 32-bit units.
LRTSTCTL_PE1L register can be read or written in 16-bit units.

Address: LRTSTCTL_PE1: FFC6 5004_H
LRTSTCTL_PE1L: FFC6 5004_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST	DATSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.31 LRTSTCTL_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST and DATSEL bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRTSTCTL_PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read by reading LRTDATBF0_PE1 register or written directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).
0	DATSEL	Data Select This bit is valid when ECCTST = 1. This bit selects the RAM bit which can be accessed when writing. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: RAM data is selected. 1: The ECC bits are selected.

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

(2) LRTDATBF0_PE1 — Local RAM Test Data Read Buffer 0

In ECC test mode (self-diagnosis), the ECC bits can be read. If the local RAM is read while ECCTST = 1 in the local RAM test control register LRTSTCTL_PE1, reading from the local RAM reads out the ECC bits, and these bits are stored in this buffer.

Access:	LRTDATBF0_PE1 register is a read-only register that can be read in 32-bit units.
Address:	LRTDATBF0_PE1: FFC6 5008$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	LRTDATBF[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.32 LRTDATBF0_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	LRTDATBF[6:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the local RAM, the ECC bits for the local RAM are stored in LRTDATBF[6:0].

(3) LRECCCTL_PE1 — Local RAM ECC Control Register

LRECCCTL_PE1 enables or disables ECC error detection and correction and 1-bit error correction.

When writing to LRECCCTL_PE1, PROT1 and PROT0 need to be 01$_B$.

Access: LRECCCTL_PE1 register can be read or written in 32-bit units.
LRECCCTL_PE1L register can be read or written in 16-bit units.

Address: LRECCCTL_PE1: FFC6 5400$_H$
LRECCCTL_PE1L: FFC6 5400$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.33 LRECCCTL_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRECCCTL_PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(4) LRERRINT_PE1 — Local RAM Error Information Control Register

LRERRINT_PE1 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access:	LRERRINT_PE1 register can be read or written in 32-bit units. LRERRINT_PE1L register can be read or written in 16-bit units. LRERRINT_PE1LL register can be read or written in 8-bit units.
Address:	LRERRINT_PE1: FFC6 5404$_H$ LRERRINT_PE1L: FFC6 5404$_H$ LRERRINT_PE1LL: FFC6 5404$_H$
Value after reset:	0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.34 LRERRINT_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(5) LRSTCLR_PE1 — Local RAM Status Clear Register

LRSTCLR_PE1 clears the error flags in the error status register (LR1STERSTR_PE1), the overflow flag in the error overflow status register (LROVFSTR_PE1), and the error address register (LR1STEADR0_PE1). LRSTCLR_PE1 is a write-only register and is always read as 0.

Access:	LRSTCLR_PE1 register is a write-only register that can be written in 32-bit units. LRSTCLR_PE1L register is a write-only register that can be written in 16-bit units. LRSTCLR_PE1LL register is a write-only register that can be written in 8-bit units.
Address:	LRSTCLR_PE1: FFC6 5408_{H} LRSTCLR_PE1L: FFC6 5408_{H} LRSTCLR_PE1LL: FFC6 5408_{H}
Value after reset:	0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40B.35 LRSTCLR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF0 and SEDF0 flags in LR1STERSTR_PE1; ERROVF0 flag in LROVFSTR_PE1; and LR1STEADR0_PE1.

(6) LROVFSTR_PE1 — Local RAM Error Count Overflow Status Register

LROVFSTR_PE1 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF0 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1.

Access:	LROVFSTR_PE1 register is a read-only register that can be read in 32-bit units. LROVFSTR_PE1L register is a read-only register that can be read in 16-bit units. LROVFSTR_PE1LL register is a read-only register that can be read in 8-bit units.
Address:	LROVFSTR_PE1: FFC6 540C$_H$ LROVFSTR_PE1L: FFC6 540C$_H$ LROVFSTR_PE1LL: FFC6 540C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.36 LROVFSTR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

(7) LR1STERSTR_PE1 — Local RAM 1st Error Status Register

LR1STERSTR_PE1 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0.

If more than one error occurs simultaneously, all the corresponding error flags are set. LR1STERSTR_PE1 is cleared by an internal reset, the external reset, or setting 1 to the STCLR0 bit in LRSTCLR_PE1.

Access: LR1STERSTR_PE1 register is a read-only register that can be read in 32-bit units.
LR1STERSTR_PE1L register is a read-only register that can be read in 16-bit units.
LR1STERSTR_PE1LL register is a read-only register that can be read in 8-bit units.

Address: LR1STERSTR_PE1: FFC6 5410$_H$
LR1STERSTR_PE1L: FFC6 5410$_H$
LR1STERSTR_PE1LL: FFC6 5410$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.37 LR1STERSTR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1 to 1. [Setting condition] ECC 2-bit error is detected with DEDF0 being 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1 to 1. [Setting condition] ECC 1-bit error is detected with both SEDF0 and DEDF0 being 0.

(8) LR1STEADR0_PE1 — Local RAM 1st Error Address Register 0

LR1STEADR0_PE1 holds the address at which an error has occurred.

The error address is set if an error occurs while error flags are 0 in LR1STERSTR_PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set in LR1STERSTR_PE1. Once an ECC 2-bit error occurs, the address is not updated.

Since this register holds the internal address, add the base address[31:19] bits of the associated memory to transform the internal address to the real address. LR1STEADR0_PE1 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1.

Access:	LR1STEADR0_PE1 register is a read-only register that can be read in 32-bit units. LR1STEADR0_PE1L and LR1STEADR0_PE1H registers are read-only registers that can be read in 16-bit units. LR1STEADR0_PE1LL, LR1STEADR0_PE1LH, and LR1STEADR0_PE1HL registers are read-only registers that can be read in 8-bit units.
Address:	LR1STEADR0_PE1: FFC6 5450$_H$ LR1STEADR0_PE1L: FFC6 5450$_H$, LR1STEADR0_PE1H: FFC6 5452$_H$ LR1STEADR0_PE1LL: FFC6 5450$_H$, LR1STEADR0_PE1LH: FFC6 5451$_H$, LR1STEADR0_PE1HL: FFC6 5452$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	EADR[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.38 LR1STEADR0_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 2	EADR[18:2]	1st Error Address Monitors the address of the first error. The error address is held if an error occurs while all the error flags are 0 in LR1STERSTR_PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.
1, 0	Reserved	When read, the value after reset is returned.

40B.2.5 Global RAM (Including the Retention RAM) ECC

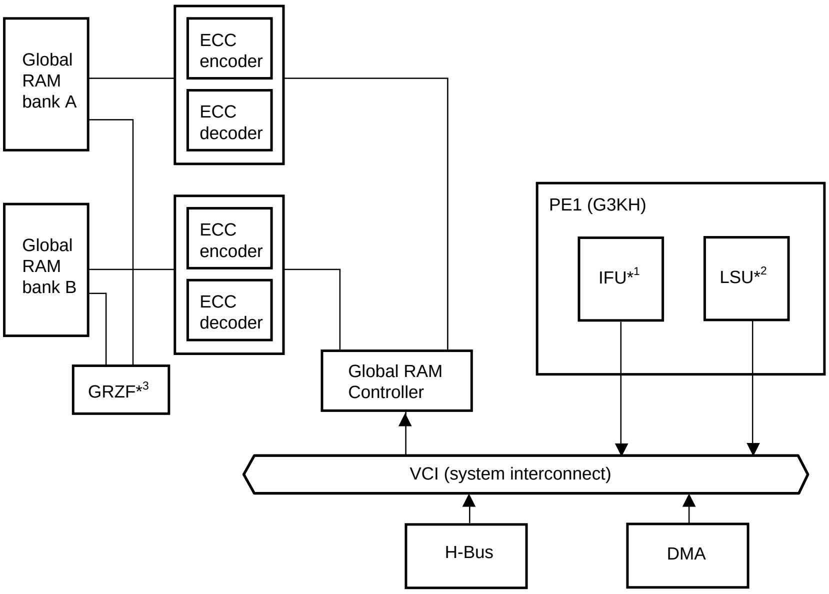
40B.2.5.1 Overview

CAUTION

The retention RAM is a part of the global RAM. The ECC for the retention RAM is shared with the global RAM's. Therefore, use the same register as the global RAM's in case of the retention RAM.

In RH850/F1KM-S4, an ECC encoder and an ECC decoder are provided for each bank (bank A and bank B).

Figure 40B.2, Block Diagram of Global RAM ECC shows the location of two ECC decoders and two ECC encoders for the global RAM.



Note 1. Instruction Fetch Unit

Note 2. Load Store Unit

Note 3. GRZF accesses RAM directly. For details, see **Section 45.3, Global RAM Zero Fill (GRZF)**.

Figure 40B.2 Block Diagram of Global RAM ECC

The global RAM ECC is summarized in the table below.

Table 40B.39 Summary of Global RAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction ● 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit error is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3BC, CPU System of RH850/F1KM**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.
Error status	A status register is provided to indicate the statuses of 2-bit ECC error detection and 1-bit ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. In addition, if the address source retained is 1-bit ECC error, it is captured when a 2-bit ECC error is detected
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data can be read directly
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

40B.2.5.2 Interrupt Requests

Global RAM ECC interrupt requests are listed below.

Table 40B.40 Global RAM ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40B.41 Global RAM ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40B.42 Global RAM ECC Interrupt Requests (During Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	INTECCRAM	—

40B.2.5.3 List of Registers

Table 40B.43 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
GRECC	FFC6 4000_H	GRECCCTL_BKA	Global RAM ECC control register (Bank A)	R/W	0000 0000_H	16/32
	FFC6 4004_H	GRERRINT_BKA	Global RAM error information control register (Bank A)	R/W	0000 0001_H	8/16/32
	FFC6 4010_H	GRSTCLR_BKA	Global RAM status clear register (Bank A)	W	0000 0000_H	8/16/32
	FFC6 4014_H	GROVFSTR_BKA	Global RAM error count overflow status register (Bank A)	R	0000 0000_H	8/16/32
	FFC6 4018_H	GR1STERSTR_BKA	Global RAM 1st error status register (Bank A)	R	0000 0000_H	8/16/32
	FFC6 $401C_H$	GR1STEADR_BKA	Global RAM 1st error address register (Bank A)	R	FEE0 0000_H	8/16/32
	FFC6 4020_H	GRTSTCTL_BKA	Global RAM test control register (Bank A)	R/W	0000 0000_H	16/32
	FFC6 4024_H	GRDECINBF1_BKA	Global RAM ECC decoder input data buffer 1 (Bank A)	R/W	0000 0000_H	8/16/32
	FFC6 4200_H	GRECCCTL_BKB	Global RAM ECC control register (Bank B)	R/W	0000 0000_H	16/32
	FFC6 4204_H	GRERRINT_BKB	Global RAM error information control register (Bank B)	R/W	0000 0001_H	8/16/32
	FFC6 4210_H	GRSTCLR_BKB	Global RAM status clear register (Bank B)	W	0000 0000_H	8/16/32
	FFC6 4214_H	GROVFSTR_BKB	Global RAM error count overflow status register (Bank B)	R	0000 0000_H	8/16/32
	FFC6 4218_H	GR1STERSTR_BKB	Global RAM 1st error status register (Bank B)	R	0000 0000_H	8/16/32
	FFC6 $421C_H$	GR1STEADR_BKB	Global RAM 1st error address register (Bank B)	R	FEF0 0000_H	8/16/32
	FFC6 4220_H	GRTSTCTL_BKB	Global RAM test control register (Bank B)	R/W	0000 0000_H	16/32
	FFC6 4224_H	GRDECINBF1_BKB	Global RAM ECC decoder input data buffer 1 (Bank B)	R/W	0000 0000_H	8/16/32

Note: _BKA and _BKB in Symbol indicate on each bank. “BKA” represents for bank A and “BKB” represents for bank B.

40B.2.5.4 Details of Registers

(1) GRECCCTL_BKA/BKB — Global RAM ECC Control Register

GRECCCTL_BKA/BKB enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01$_B$ when writing to GRECCCTL_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access:	GRECCCTL_BKA and GRECCCTL_BKB can be read or written in 32-bit units GRECCCTL_BKAL and GRECCCTL_BKBL can be read or written in 16-bit units
Address:	GRECCCTL_BKA: FFC6 4000$_H$ GRECCCTL_BKAL: FFC6 4000$_H$ GRECCCTL_BKB: FFC6 4200$_H$ GRECCCTL_BKBL: FFC6 4200$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40B.44 GRECCCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0.Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. When writing, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. When writing, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) GRERRINT_BKA/BKB— Global RAM Error Information Control Register

GRERRINT_BKA/BKB enables or disables generation of the error notification signal upon detection of an ECC 1-bit error.

The setting of this register is used for accesses through the respective access port.

Access: GRERRINT_BKA and GRERRINT_BKB can be read or written in 32-bit units
GRERRINT_BKAL and GRERRINT_BKBL can be read or written in 16-bit units
GRERRINT_BKALL and GRERRINT_BKBLL can be read or written in 8-bit units

Address: GRERRINT_BKA: FFC6 4004$_H$
GRERRINT_BKAL: FFC6 4004$_H$
GRERRINT_BKALL: FFC6 4004$_H$
GRERRINT_BKB: FFC6 4204$_H$
GRERRINT_BKBL: FFC6 4204$_H$
GRERRINT_BKBLL: FFC6 4204$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40B.45 GRERRINT_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(3) GRSTCLR_BKA/BKB — Global RAM Status Clear Register

GRSTCLR_BKA/BKB clears the error flags in the error status register (GR1STERSTR_BKA/BKB), the overflow flag in the error count overflow status register (GROVFSTR_BKA/BKB), and the error address register (GR1STEADR_BKA/BKB). GRSTCLR_BKA/BKB is a write-only register and is always read as 0.

The setting of this register is used for accesses through the respective access port.

Access:	GRSTCLR_BKA and GRSTCLR_BKB are write-only registers that can be written in 32-bit units. GRSTCLR_BKAL and GRSTCLR_BKBL are write-only registers that can be written in 16-bit units. GRSTCLR_BKALL and GRSTCLR_BKBLL are write-only registers that can be written in 8-bit units
Address:	GRSTCLR_BKA: FFC6 4010_H GRSTCLR_BKAL: FFC6 4010_H GRSTCLR_BKALL: FFC6 4010_H GRSTCLR_BKB: FFC6 4210_H GRSTCLR_BKBL: FFC6 4210_H GRSTCLR_BKBLL: FFC6 4210_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40B.46 GRSTCLR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, always write 0.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF and SEDF flags in GR1STERSTR_BKA/BKB, ERROVF0 flag in GROVFSTR_BKA/BKB, and EADR[19:0] in GR1STEADR_BKA/BKB.

(4) GROVFSTR_BKA/BKB— Global RAM Error Count Overflow Status Register

GROVFSTR_BKA/BKB monitors occurrence of error overflow. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

The setting of this register is used for accesses through the respective access port.

Access: GROVFSTR_BKA and GROVFSTR_BKB are read-only registers that can be read in 32-bit units
GROVFSTR_BKAL and GROVFSTR_BKBL are read-only registers that can be read in 16-bit units
GROVFSTR_BKALL and GROVFSTR_BKBLL are read-only registers that can be read in 8-bit units

Address: GROVFSTR_BKA: FFC6 4014$_H$
GROVFSTR_BKAL: FFC6 4014$_H$
GROVFSTR_BKALL: FFC6 4014$_H$
GROVFSTR_BKB: FFC6 4214$_H$
GROVFSTR_BKBL: FFC6 4214$_H$
GROVFSTR_BKBLL: FFC6 4214$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.47 GROVFSTR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF and SEDF) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error.

(5) GR1STERSTR_BKA/BKB— Global RAM 1st Error Status Register

GR1STERSTR_BKA/BKB monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. However, the corresponding error flag is set if an ECC 2-bit error occurs only when SEDF bit is set. GR1STERSTR_BKA/BKB is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in GRSTCLR_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access: GR1STERSTR_BKA and GR1STERSTR_BKB are read-only registers that can be read in 32-bit units.
GR1STERSTR_BKAL and GR1STERSTR_BKBL are read-only registers that can be read in 16-bit units.
GR1STERSTR_BKALL and GR1STERSTR_BKBLL are read-only registers that can be read in 8-bit units.

Address: GR1STERSTR_BKA: FFC6 4018$_{\text{H}}$
GR1STERSTR_BKAL: FFC6 4018$_{\text{H}}$
GR1STERSTR_BKALL: FFC6 4018$_{\text{H}}$
GR1STERSTR_BKB: FFC6 4218$_{\text{H}}$
GR1STERSTR_BKBL: FFC6 4218$_{\text{H}}$
GR1STERSTR_BKBLL: FFC6 4218$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.48 GR1STERSTR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in GRSTCLR_BKA/BKB to 1. [Setting condition] ECC 2-bit error is detected when DEDF is 0.
0	SEDF	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the STCLR0 bit in GRSTCLR_BKA/BKB to 1. [Setting condition] ECC 1-bit error is detected when DEDF and SEDF are 0.

(6) GR1STEADR_BKA/BKB— Global RAM 1st Error Address Register

GR1STEADR_BKA/BKB holds the address at which an error has occurred.
The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR_BKA/BKB. The address

is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error occurs, the address is not updated.
GR1STEADR_BKA/BKB is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in GRSTCLR_BKA/BKB.

The setting of this register is used for accesses through the respective access port.

Access: GR1STEADR_BKA and GR1STEADR_BKB are read-only registers that can be read in 32-bit units.
GR1STEADR_BKAL, GR1STEADR_BKBL, GR1STEADR_BKAH and GR1STEADR_BKBH are read-only registers that can be read in 16-bit units.
GR1STEADR_BKALL, GR1STEADR_BKALH, GR1STEADR_BKAHL, GR1STEADR_BKBLL, GR1STEADR_BKBLH and GR1STEADR_BKBHL are read-only registers that can be read in 8-bit units.

Address: GR1STEADR_BKA: FFC6 401C$_H$
GR1STEADR_BKAL: FFC6 401C$_H$
GR1STEADR_BKALL: FFC6 401C$_H$
GR1STEADR_BKALH: FFC6 401D$_H$
GR1STEADR_BKAH: FFC6 401E$_H$
GR1STEADR_BKAHL: FFC6 401E$_H$
GR1STEADR_BKB: FFC6 421C$_H$
GR1STEADR_BKBL: FFC6 421C$_H$
GR1STEADR_BKBLL: FFC6 421C$_H$
GR1STEADR_BKBLH: FFC6 421D$_H$
GR1STEADR_BKBH: FFC6 421E$_H$
GR1STEADR_BKBHL: FFC6 421E$_H$

Value after reset: GR1STEADR_BKA: FEE0 0000$_H$, GR1STEADR_BKB: FEF0 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	EADR[19:16]			
Value after reset	1	1	1	1	1	1	1	0	1	1	1	0/1^{*1}	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.49 GR1STEADR_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	EADR[19:0]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR_BKA/BKB. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Note 1. The value after reset differs depending on bank A or bank B.

(7) GRTSTCTL_BKA/BKB — Global RAM Test Control Register

This register is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST to 1, any data can be written to the ECC bits.
Also, input and output by the ECC decoder in the global RAM controller can be controlled for testing (self-diagnosis).

Set PROT1 to 0 and PROT0 to 1 when writing to this register.

Access:	GRTSTCTL_BKA and GRTSTCTL_BKB can be read or written in 32-bit units GRTSTCTL_BKAL and GRTSTCTL_BKBL can be read or written in 16-bit units
Address:	GRTSTCTL_BKA: FFC6 4020_H GRTSTCTL_BKAL: FFC6 4020_H GRTSTCTL_BKB: FFC6 4220_H GRTSTCTL_BKBL: FFC6 4220_H
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40B.50 GRTSTCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15	PROT1	Enables or disables modification of the ECCTST. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to GRTSTCTL_BKA/BKB.
14	PROT0	
13 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ECCTST	ECC test bit. This bit permits to switch the ECC data for writing into RAM. Assert (PROT1, PROT0) = (0, 1) to write to this register. 0: Output of ECC encoder is used for ECC data to RAM 1: Register output is used for ECC data to RAM.

(8) GRDECINBF1_BKA/BKB— Global RAM ECC Decoder Input Data Buffer 1

7-bit ECC of global RAM is replaced to this register value by write access to global RAM during GRTSTCTL_BKA/BKB.ECCTST = 1 (test mode).

Access: GRDECINBF1_BKA and GRDECINBF1_BKB can be read or written in 32-bit units.
GRDECINBF1_BKAL and GRDECINBF1_BKBL can be read or written in 16-bit units.
GRDECINBF1_BKALL and GRDECINBF1_BKBLL can be read or written in 8-bit units.

Address: GRDECINBF1_BKA: FFC6 4024_H
GRDECINBF1_BKAL: FFC6 4024_H
GRDECINBF1_BKALL: FFC6 4024_H
GRDECINBF1_BKB: FFC6 4224_H
GRDECINBF1_BKBL: FFC6 4224_H
GRDECINBF1_BKBLL: FFC6 4224_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	GRDECINBF1[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40B.51 GRDECINBF1_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, these bits are always read as 0. When writing, always write 0.
6 to 0	GRDECINBF1[6:0]	This field is valid ECCTST = 1 (test mode). When the write operation is executed, the data in this register is used for writing to 7-bit ECC field RAM.

40B.2.6 ECC for Peripheral RAM

40B.2.6.1 Overview

This is an ECC module for the RAM of the following peripheral modules.

CSIH, RS-CANFD, FlexRay and Ethernet AVB.

Table 40B.52 List of the ECC Functions for the peripheral RAM

Item	Description
ECC error detection/correction	ECC error detection and correction can be enabled or disabled. Either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction ● 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. In the initial state, 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Error notification can be enabled or disabled when an ECC 2-bit error is detected. ● Error notification can be enabled or disabled when an ECC 1-bit error is detected. In the initial state of ECC controller, 2-bit error notification is enabled and 1-bit error notification is disabled. However, if an interrupt is masked by the FEINTFMSK register, an interrupt processing is not executed.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC1-bit errors is available. A bit for clearing the error status is provided.
Address capture	Only one address at which an ECC error has occurred can be captured. A signal is generated upon detection of ECC 2-bit or 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed about RS-CANFD or FlexRay, initialize the RAM area before it is used.

40B.2.6.2 Interrupt Requests

ECC of peripheral interrupt requests are listed below.

Table 40B.53 CSIHn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
INTECCCSIHn	CSIHn ECC 1-bit error or 2-bit error interrupt	INTECCCSIH0	—
		INTECCCSIH1	
		INTECCCSIH2	
		INTECCCSIH3	

Table 40B.54 RCFDCn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
ECCCNFDRAMFEIF	RCFDC0 ECC 1-bit error or 2-bit error interrupt	INTECCCNFDRAM	—

Table 40B.55 FlexRay ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	FLXA0 ECC 1-bit error or 2-bit error interrupt	INTECCFLRAM	—

Table 40B.56 Ethernet AVB ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ETNB0 ECC 1-bit error or 2-bit error interrupt	INTECCETH	—

40B.2.6.3 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 40B.57 List of ECC Modules

Peripheral Functions		Module Name	ECC Module Names and Register	
			Base Address Name	Base Address <base_addr>
CSIHn	Buffered I/O (CSIH RAM)	ECCCSIH0	<ECCCSIH0_base>	FFC7 0100_H
		ECCCSIH1	<ECCCSIH1_base>	FFC7 0200_H
		ECCCSIH2	<ECCCSIH2_base>	FFC7 0300_H
		ECCCSIH3	<ECCCSIH3_base>	FFC7 0400_H
RCFDC0	Message bufferRAM (MB RAM)	ECCCFD0MB	<ECCCFD0MB_base>	FFC7 1300_H
	Acceptance filter list RAM0 (AFL0 RAM)	ECCCFD0AFL0	<ECCCFD0AFL0_base>	FFC7 1400_H
	Acceptance filter list RAM1 (AFL1 RAM)	ECCCFD0AFL1	<ECCCFD0AFL1_base>	FFC7 1500_H
FLXA0	Message RAM (MRAM)	ECCFLXA0	<ECCFLXA0_base>	FFC7 3100_H
	Temporary buffer (TBFA)	ECCFLXA0T0	<ECCFLXA0T0_base>	FFC7 3200_H
	Temporary buffer (TBFB)	ECCFLXA0T1	<ECCFLXA0T1_base>	FFC7 3300_H
ETNB0	Transmit FIFO RAM.(TXRAM)	ECCETNB0TX	<ECCETNB0TX_base>	FFC7 4100_H
	Receive FIFO RAM.(RXRAM)	ECCETNB0RX	<ECCETNB0RX_base>	FFC7 4200_H

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 40B.58 List of Registers

Module Name	Register Name	Symbol	Address
See (1) List of ECC Modules.	ECC control register	<Module_Name>CTL	<base_addr> + 00_H
	ECC test mode control register	<Module_Name>TMC	<base_addr> + 04_H
	ECC encode/decode input/output replacement test register	<Module_Name>TED	<base_addr> + $0C_H$
	ECC redundant bit data control test register	<Module_Name>TRC	<base_addr> + 08_H
	ECC decode syndrome data register	<Module_Name>SYND	<base_addr> + $0B_H$
	ECC 7-bit redundant bit data hold test register	<Module_Name>HORD	<base_addr> + $0A_H$
	ECC encode test register	<Module_Name>ECRD	<base_addr> + 09_H
	ECC redundant bit input/output replacement register	<Module_Name>ERDB	<base_addr> + 08_H
	ECC error address register 0	<Module_Name>AD0	<base_addr> + 10_H
SL_READTEST	ECCREAD test select register	SELB_READTEST	FFC7 8000H

40B.2.6.4 Details of Registers

(1) <Module_Name>CTL — ECC Control Register

The <Module_Name>CTL register controls the mode of the ECC and the status for target peripheral modules.

Bits 7, 5, 4 and 3 should be set (written) while the target peripheral module’s operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: $001X_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	ECCOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 40B.59 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
15	EMCA1	Access Control Bits 1 and 0 to ECC Mode Selection These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ECCOVFF	By detecting an error while the error status is set and the new error has another address than the already latched (not cleared or reset is not issued), this bit is set. 0: Overflow is not occurred after reset of clearing ECER2F and ECER1F. 1: Error address register overflowed **NOTE:** This bit clear condition is as follows. (1) Reset (2) Writing ECER2C = 1 when ECER2F = 1 or ECER1C = 1 when ECER1F = 1 (3) Selecting through mode enable (ECTHM = 1)
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority. ECER2C = 1 also clears the ECCOVFF bit while ECER2F bit is set.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority. ECER1C = 1 also clears the ECCOVFF bit while ECER1F bit is set.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40B.59 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
7	ECTHM	ECC Function through Mode Selection Set this bit to select whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, write 01_B to the EMCA1 and EMCA0 bits at the same time. Set this bit to 1 to disable the ECC function. 0: Passing through mode is disabled (normal operation mode). 1: Passing through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 2-bit error is detected, a target peripheral module's interrupt will be generated.
3	EC1EDIC	1-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 1-bit error is detected, a target peripheral module's interrupt will be generated.
2	ECER2F	2-Bit Error Detection Flag This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (a target peripheral module's interrupt) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. When through mode is enable (ECTHM = 1), this bit is cleared. If 2-bit error is detected again while this bit is set, an interrupt will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
1	ECER1F	1-Bit Error Detection/Correction Flag This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. When through mode is enabled (ECTHM = 1), this bit is cleared. 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
0	ECEMF	ECC Error Message Flag This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. This bit is also cleared when through mode is enabled (ECTHM = 1) and there is no 1-bit error in decode circuit input data. 0: The current RAM output data does not have bit errors. 1: The current RAM output data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

(2) <Module_Name>TMC — ECC Test Mode Control Register

The <Module_Name>TMC register is used to switch to the test mode, and this register is for test mode control.

This register can be used when a target peripheral module is not accessed to RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 40B.60 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
15	ETMA1	Access Control Bits 1 and 0 to ECC Test Mode These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to test control bits of the test registers and this register. When writing to this bit, write 10_B to the ETMA1 and ETMA0 bits at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: <Module_Name>TED, <Module_Name>TRC, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, <Module_Name>ERDB Register test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the <Module_Name>TED and <Module_Name>ERDB registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the <Module_Name>TED register will be the write value of the <Module_Name>TED register. Read value of the <Module_Name>ERDB register will be the write value of the <Module_Name>ERDB register. 1: Read value of the <Module_Name>TED register can read RAM data. Read value of the <Module_Name>ERDB register will be the ECC Data to be written to RAM.
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies which is output to the ECC to be stored in RAM, the ECC data generated for write data or the value of the <Module_Name>ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated for write data is stored in RAM. 1: The value of <Module_Name>ERDB Register is stored in RAM.

Table 40B.60 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Selection This bit specifies data written to RAM or the value of the <Module_Name>TED register as the input to the ECC encoder. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the <Module_Name>TED.
1	ECDCS	ECC Decoder Input Selection This bit specifies which data is for generation of syndrome code and error detection, RAM data or the value of <Module_Name>TED. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from RAM Data. 1: Syndrome code generation and error detection are performed from <Module_Name>TED register value.
0	ECREIS	ECC Redundant Bit Input Data Selection This bit specifies which ECC data is for generation of syndrome code and error detection, ECC data stored in RAM or the value of the <Module_Name>ERDB. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from <Module_Name>ERDB register value.

(3) <Module_Name>TED — ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

This register value is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000$_H$ is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access: This register can be read or written in 32-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40B.61 <Module_Name>TED Register Contents

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When <Module_Name>TMC.ECENS = 1, the value of this register is used to generate ECC data and that is stored to RAM. When <Module_Name>TMC.ECDCS = 1, the value of this register is used to generate syndrome code that is stored in ECC decode syndrome data register (<Module_Name>SYND). In addition, when <Module_Name>TMC.ECTRRS = 1, RAM data [31:0] instead of written data is read for the value of this register.

(4) <Module_Name>TRC — ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and consists of four 8-bit registers, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, and <Module_Name>ERDB.

This register can be accessed when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access:	This register can be read or written in 32-bit units.
Address:	See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	<Module_Name>SYND (see **Section (6)**)								<Module_Name>HORD (see **Section (7)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<Module_Name>ECRD (see **Section (8)**)								<Module_Name>ERDB (see **Section (9)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(5) <Module_Name>AD0 — Target ECC Error Address Register 0

This is read only register to hold the ECC error occurred address.

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as trigger and it is hold as the error occurring address.

Access:	This register is a read-only register that can be read in 32-bit units.
Address:	See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ECEAD[30:16]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.62 <Module_Name>AD0 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 0	ECEAD[30:0]	ECEAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECEAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECEAD0

(6) <Module_Name>SYND — ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register is read-only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40B.63 <Module_Name>SYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

(7) <Module_Name>HORD — ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40B.64 <Module_Name>HORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. When <Module_Name>TMC.ECTRRS =1 and if <Module_Name>TED register is read, ECC code is stored.

(8) <Module_Name>ECRD — ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40B.65 <Module_Name>ECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing and can read ECC data for data written in the <Module_Name>TED register when <Module_Name>MC.ECENS =1.

(9) <Module_Name>ERDB — ECC Redundant Bit Input/Output Replacement Register

In ECC test mode, this register handles test data.

This register value can be handled as generated ECC data at the time of writing to RAM or as read ECC data at the time of reading RAM data.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Access:	This register can be read or written in 8-bit units.
Address:	See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40B.66 <Module_Name>ERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	These bits can store this register value as ECC data when <Module_Name>TMC.ECREOS = 1. When the register is read while <Module_Name>TMC.ECREIS =1, the value read from these bits is ECC data read from the RAM. When <Module_Name>TMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instated of written data.

(10) SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the target peripheral module's RAM ECC registers.

Setting 1 to the bit corresponding to each function will enable writing to the read-only bit.

Access:	This register can be read or written in 16-bit units.
Address:	See **Table 40B.57, List of ECC Modules** and **Table 40B.58, List of Registers**.
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCANFDE7A03	RTCANFDE7A02	RTCANFDE7A01	RTETHRXRAM	RTETHTXRAM	—	—	—	—	RTFLXAE7A0	RTFLXATRAM1	RTFLXATRAM0	RTCSIHE7A3	RTCSIHE7A2	RTCSIHE7A1	RTCSIHE7A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40B.67 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
15	RTCANFDE7A03	RCFDC0 (AFL Buffer 1) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
14	RTCANFDE7A02	RCFDC0 (AFL Buffer 0) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
13	RTCANFDE7A01	RCFDC0 (Message Buffer) ECC Register Write Access for Testing Purpose Enable /Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
12	RTETHRXRAM	ETNB0 (RXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB0 (RXRAM) ECC read-only bit can be written).
11	RTETHTXRAM	ETNB0 (TXRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (The ETNB0 (TXRAM) ECC read-only bit can be written).
10 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	RTFLXAE7A0	FLXA0 (MRAM) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (MRAM) ECC read-only bit can be written).
5	RTFLXATRAM1	FLXA0 (TBFB) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (TBFB) ECC read-only bit can be written).
4	RTFLXATRAM0	FLXA0 (TBFA) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (FLXA0 (TBFA) ECC read-only bit can be written).
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH3 ECC read-only bit can be written).
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH2 ECC read-only bit can be written).

Table 40B.67 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH0 ECC read-only bit can be written).

40B.3 Memory Protection

40B.3.1 Overview

This product incorporates the memory protection function to prevent erroneous accesses to data in memories and control registers of the peripheral circuits.

- MPU
  The CPU protects memory against illegal access by itself. The CPU does not generate the signals for access to addresses where access is prohibited by the MPU. For details, see *the RH850 Family User's Manual: Software*.
- Slave Guard
  A specific memory is protected against illegal accesses from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.
  - PEG
    The local RAM is protected against illegal accesses. However, accesses from the CPU incorporating the local RAM itself are excluded. For example, accesses from the CPU1 to local RAM in the CPU1 are not rejected by the PEG. For details, see **Section 3BC, CPU System of RH850/F1KM**.
  - IPG
    The CPU Peripheral is protected against illegal accesses.
    For details, see **Section 3BC, CPU System of RH850/F1KM**.
  - GRG
    The global RAM is protected against illegal accesses.
  - PBG/HBG
    The control registers in the peripheral circuits are protected against illegal accesses. For details, see **Section 40B.3.3, PBG/HBG**.
  - PBGC
    The CPU system has its dedicated PBG function which is called PBG for CPU system. For details, see **Section 40B.3.4, PBG for CPU System**.

40B.3.1.1 Identifiers for Slave Guard

For the slave guard function, the type of illegal accesses to be rejected can be designated using the following identifiers.

Table 40B.68 Identifiers for Slave Guard

Identifier	Function
UM	When the CPU makes an access, this indicates the operating mode of the CPU. 0: Supervisor mode 1: User mode When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 0.
SPID	When the CPU makes an access, this indicates the system protection identifier SPID that is assigned to the CPU. When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 00_B.
PEID	This indicates the access source bus master. 001_B: CPU1 100_B: Other bus master (H-Bus bus master) When the PDMA makes an access, the value of this identifier is the value in the channel master setting register.

40B.3.2 GRG (Global RAM Guard)

This product is provided with 4-channel GRG, which is described in detail in the following sections.

CAUTION

The retention RAM is a part of the global RAM. The Guard for the retention RAM is shared with the global RAM's. Therefore, use the same register as the global RAM's in case of the retention RAM.

40B.3.2.1 List of Registers

Table 40B.69 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
MGDGR	FFC4 9000$_H$	MGDGRPROT0_BKA	GRG protection setting register0 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9004$_H$	MGDGRBAD0_BKA	GRG compare base address register0 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9008$_H$	MGDGRADV0_BKA	GRG compare address valid register0 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9010$_H$	MGDGRPROT1_BKA	GRG protection setting register1 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9014$_H$	MGDGRBAD1_BKA	GRG compare base address register1 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9018$_H$	MGDGRADV1_BKA	GRG compare address valid register1 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9020$_H$	MGDGRPROT2_BKA	GRG protection setting register2 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9024$_H$	MGDGRBAD2_BKA	GRG compare base address register2 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9028$_H$	MGDGRADV2_BKA	GRG compare address valid register2 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9030$_H$	MGDGRPROT3_BKA	GRG protection setting register3 (Bank A)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9034$_H$	MGDGRBAD3_BKA	GRG compare base address register3 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9038$_H$	MGDGRADV3_BKA	GRG compare address valid register3 (Bank A)	R/W	0000 0000$_H$	8/16/32
	FFC4 9040$_H$	MGDGRSCTL_BKA	GRG control register (Bank A)	W	0000 0000$_H$	8/16/32
	FFC4 9044$_H$	MGDGRSSTAT_BKA	GRG error status register (Bank A)	R	0000 0000$_H$	8/16/32
	FFC4 9048$_H$	MGDGRSTYPE_BKA	GRG error access type register (Bank A)	R	0000 0000$_H$	8/16/32
	FFC4 904C$_H$	MGDGRSAD_BKA	GRG error address register (Bank A)	R	FEE0 0000$_H$	8/16/32
	FFC4 9200$_H$	MGDGRPROT0_BKB	GRG protection setting register0 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9204$_H$	MGDGRBAD0_BKB	GRG compare base address register0 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9208$_H$	MGDGRADV0_BKB	GRG compare address valid register0 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9210$_H$	MGDGRPROT1_BKB	GRG protection setting register1 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9214$_H$	MGDGRBAD1_BKB	GRG compare base address register1 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9218$_H$	MGDGRADV1_BKB	GRG compare address valid register1 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9220$_H$	MGDGRPROT2_BKB	GRG protection setting register2 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9224$_H$	MGDGRBAD2_BKB	GRG compare base address register2 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9228$_H$	MGDGRADV2_BKB	GRG compare address valid register2 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9230$_H$	MGDGRPROT3_BKB	GRG protection setting register3 (Bank B)	R/W	03FE 01F0$_H$	8/16/32
	FFC4 9234$_H$	MGDGRBAD3_BKB	GRG compare base address register3 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9238$_H$	MGDGRADV3_BKB	GRG compare address valid register3 (Bank B)	R/W	0000 0000$_H$	8/16/32
	FFC4 9240$_H$	MGDGRSCTL_BKB	GRG control register (Bank B)	W	0000 0000$_H$	8/16/32
	FFC4 9244$_H$	MGDGRSSTAT_BKB	GRG error status register (Bank B)	R	0000 0000$_H$	8/16/32
	FFC4 9248$_H$	MGDGRSTYPE_BKB	GRG error access type register (Bank B)	R	0000 0000$_H$	8/16/32
	FFC4 924C$_H$	MGDGRSAD_BKB	GRG error address register (Bank B)	R	FEF0 0000$_H$	8/16/32

Note: MGDGRPROTn_BKA/BKB, MGDGRBADn_BKA/BKB, and MGDGRADVn_BKA/BKB set the protection specifications for each channel (n = 0 to 3).
MGDGRSCTL_*, MGDGRSSTAT_*, MGDGRSTYPE_* and MGDGRSAD_* indicate error information on each bank.
“BKA” represents for Bank A and “BKB” represents for Bank B.

40B.3.2.2 Details of Registers

(1) MGDGRPROTn_BKA/BKB — GRG Protection Setting Register n (n = 0 to 3)

Access:	MGDGRPROTn_BKA and MGDGRPROTn_BKB can be read or written in 32-bit units MGDGRPROTn_BKAL, MGDGRPROTn_BKBL, MGDGRPROTn_BKAH and MGDGRPROTn_BKBH can be read or written in 16-bit units MGDGRPROTn_BKALL, MGDGRPROTn_BKAHL, MGDGRPROTn_BKAHH, MGDGRPROTn_BKBLL, MGDGRPROTn_BKBHL and MGDGRPROTn_BKBHH can be read or written in 8-bit units.
Address:	MGDGRPROTn_BKA: FFC4 9000_H + n × 10_H MGDGRPROTn_BKAL: FFC4 9000_H + n × 10_H MGDGRPROTn_BKALL: FFC4 9000_H + n × 10_H MGDGRPROTn_BKAH: FFC4 9002_H + n × 10_H MGDGRPROTn_BKAHL: FFC4 9002_H + n × 10_H MGDGRPROTn_BKAHH: FFC4 9003_H + n × 10_H MGDGRPROTn_BKB: FFC4 9200_H + n × 10_H MGDGRPROTn_BKBL: FFC4 9200_H + n × 10_H MGDGRPROTn_BKBLL: FFC4 9200_H + n × 10_H MGDGRPROTn_BKBH: FFC4 9202_H + n × 10_H MGDGRPROTn_BKBHL: FFC4 9202_H + n × 10_H MGDGRPROTn_BKBHH: FFC4 9203_H + n × 10_H
Value after reset:	03FE $01F0_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	EN	—	—	—	—	UM	—	—	—	PEID4	—	—	PEID1	—	—
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
R/W	R	R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SPID3	SPID2	SPID1	SPID0	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R

Table 40B.70 MGDGRPROTn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	EN	Protection Enable 0: Disables protection. 1: Enables protection. Only access permitted by this register is possible.
29 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	UM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20, 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.

Table 40B.70 MGDGRPROTn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SPID3	Access with SPID = 3*[2] 0: Disables access with SPID3. 1: Enables access with SPID3.
7	SPID2	Access with SPID = 2*[2] 0: Disables access with SPID2. 1: Enables access with SPID2.
6	SPID1	Access with SPID = 1 (CPU1*[3])*[2] 0: Disables access with SPID1. 1: Enables access with SPID1.
5	SPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*[2] 0: Disables access with SPID0. 1: Enables access with SPID0.
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. Access with PEID
PEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
SPID is a bit list with each bit corresponding to a SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

CAUTION

Global RAM Guard (GRG) only supports write accesses. The guard setting is not enabled for read accesses.

(2) MGDGRBADn_BKA/BKB — GRG Compare Base Address Register n (n = 0 to 3)

Access: MGDGRBADn_BKA and MGDGRBADn_BKB can be read or written in 32-bit units.
MGDGRBADn_BKAL, MGDGRBADn_BKAH, MGDGRBADn_BKBL and MGDGRBADn_BKBH can be read or written in 16-bit units.
MGDGRBADn_BKALH, MGDGRBADn_BKAHL, MGDGRBADn_BKBLH and MGDGRBADn_BKBHL can be read or written in 8-bit units.

Address:
MGDGRBADn_BKA: FFC4 9004_H + n × 10_H
MGDGRBADn_BKAL: FFC4 9004_H + n × 10_H
MGDGRBADn_BKALH: FFC4 9005_H + n × 10_H
MGDGRBADn_BKAH: FFC4 9006_H + n × 10_H
MGDGRBADn_BKAHL: FFC4 9006_H + n × 10_H
MGDGRBADn_BKB: FFC4 9204_H + n × 10_H
MGDGRBADn_BKBL: FFC4 9204_H + n × 10_H
MGDGRBADn_BKBLH: FFC4 9205_H + n × 10_H
MGDGRBADn_BKBH: FFC4 9206_H + n × 10_H
MGDGRBADn_BKBHL: FFC4 9206_H + n × 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	AD[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD[15:11]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 40B.71 MGDGRBADn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 11	AD[19:11]	Compare Base Address*[1] These bits set the base address of protection setting domain specified by the GRG protection setting register n (MGDGRPROTn_BKA/BKB).
10 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. For details, see **Section 40B.3.2.2(3), MGDGRADVn_BKA/BKB — GRG Compare Address Valid Register n (n = 0 to 3)**.

(3) MGDGRADVn_BKA/BKB — GRG Compare Address Valid Register n (n = 0 to 3)

Access: MGDGRADVn_BKA and MGDGRADVn_BKB can be read or written in 32-bit units.

MGDGRADVn_BKAL, MGDGRADVn_BKAH, MGDGRADVn_BKBL and MGDGRADVn_BKBH can be read or written in 16-bit units.

MGDGRADVn_BKALH, MGDGRADVn_BKAHL, MGDGRADVn_BKBLH and MGDGRADVn_BKBHL can be read or written in 8-bit units.

Address:
MGDGRADVn_BKA: FFC4 9008$_H$ + n × 10$_H$
MGDGRADVn_BKAL: FFC4 9008$_H$ + n × 10$_H$
MGDGRADVn_BKALH: FFC4 9009$_H$ + n × 10$_H$
MGDGRADVn_BKAH: FFC4 900A$_H$ + n × 10$_H$
MGDGRADVn_BKAHL: FFC4 900A$_H$ + n × 10$_H$
MGDGRADVn_BKB: FFC4 9208$_H$ + n × 10$_H$
MGDGRADVn_BKBL: FFC4 9208$_H$ + n × 10$_H$
MGDGRADVn_BKBLH: FFC4 9209$_H$ + n × 10$_H$
MGDGRADVn_BKBH: FFC4 920A$_H$ + n × 10$_H$
MGDGRADVn_BKBHL: FFC4 920A$_H$ + n × 10$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	ADV[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:11]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 40B.72 MGDGRADVn_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 11	ADV[19:11]	Valid Compare Address Setting MGDGRADVn_BKA/BKB[19:11] to 1 executes address compare. If all the bits of MGDGRADVn_BKA/BKB[19:11] are 1, 2 Kbytes, which is the minimum unit based on the address specified by MGDGRBADn_BKA/BKB, are protected. However, if all the bits of MGDGRADVn_BKA/BKB[19:11] are 0, all the areas of global RAM are protected.
10 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Setting example

If MGDGRBADn_BKA.AD[19:11] is 1D0$_H$ and MGDGRADVn_BKA.ADV[19:11] is 1FD$_H$, global RAM guard protection area n is from FEEE 8000$_H$ to FEEE 87FF$_H$ and FEEE 9000$_H$ to FEEE 97FF$_H$.

If MGDGRBADn_BKB.AD[19:11] is 1D0$_H$ and MGDGRADVn_BKB.ADV[19:11] is 1FD$_H$, global RAM guard protection area n is from FEFE 8000$_H$ to FEFE 87FF$_H$ and FEFE 9000$_H$ to FEFE 97FF$_H$.

Concept (e.g. bank A)

When MGDGRBADn_BKA.AD[19:11] is 1D0$_H$, the base address is FEEE 8000$_H$.
The settable range is shown by an underline as follows:

1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	E	8	0	0	0	H

The settable range is as follows when MGDGRADVn_BKA.ADV[19:11] is 1FD$_H$:

0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	1 1 1 0	1 0 0 0	0 0 0 0	0 0 0 0	B

The area to which protection applies is as shown below because the bits set to 0 and the lower eleven bits are not applicable:

0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0	1 0 0 x	0 x x x	x x x x	x x x x	B

Therefore the protection range is as follows:

1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	E	8	0	0	0	H

to

1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 0 0 0	0 1 1 1	1 1 1 1	1 1 1 1	B
F	E	E	E	8	7	F	F	H

and

1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0	B
F	E	E	E	9	0	0	0	H

to

1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 0 0 1	0 1 1 1	1 1 1 1	1 1 1 1	B
F	E	E	E	9	7	F	F	H

The following 2 Kbytes each (a total of 4 Kbytes) are protected (in hexadecimal):

- FEEE 8000$_H$ to FEEE 87FF$_H$
- FEEE 9000$_H$ to FEEE 97FF$_H$

(4) MGDGRSCTL_BKA/BKB — GRG Control Register

Access: MGDGRSCTL_BKA and MGDGRSCTL_BKB registers are write-only registers that can be written in 32-bit units.
MGDGRSCTL_BKAL and MGDGRSCTL_BKBL registers are write-only registers that can be written in 16-bit units.
MGDGRSCTL_BKALL and MGDGRSCTL_BKBLL registers are write-only registers that can be written in 8-bit units.

Address: MGDGRSCTL_BKA: FFC4 9040$_H$
MGDGRSCTL_BKAL: FFC4 9040$_H$
MGDGRSCTL_BKALL: FFC4 9040$_H$
MGDGRSCTL_BKB: FFC4 9240$_H$
MGDGRSCTL_BKBL: FFC4 9240$_H$
MGDGRSCTL_BKBLL: FFC4 9240$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERRCLO	ERRCLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40B.73 MGDGRSCTL_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	ERRCLO	Error Entry Overflow Flag Clear 0: No operation 1: Clears the overflow flag.
0	ERRCLE	GRG Protection Violation Flag Clear 0: No operation 1: Clears the error flag. Set this bit at the same time as ERRCLO as shown in **Table 40B.74, Combinations of ERRCLO and ERRCLE**.

Table 40B.74 Combinations of ERRCLO and ERRCLE

ERRCLO	ERRCLE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(5) MGDGRSSTAT_BKA/BKB— GRG Error Status Register

Access: MGDGRSSTAT_BKA and MGDGRSSTAT_BKB are read-only registers that can be read in 32-bit units.
MGDGRSSTAT_BKAL and MGDGRSSTAT_BKBL are read-only registers that can be read in 16-bit units.
MGDGRSSTAT_BKALL and MGDGRSSTAT_BKBLL are read-only registers that can be read in 8-bit units.

Address: MGDGRSSTAT_BKA: FFC4 9044_H
MGDGRSSTAT_BKAL: FFC4 9044_H
MGDGRSSTAT_BKALL: FFC4 9044_H
MGDGRSSTAT_BKB: FFC4 9244_H
MGDGRSSTAT_BKBL: FFC4 9244_H
MGDGRSSTAT_BKBLL: FFC4 9244_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.75 MGDGRSSTAT_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of GRG error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether overflow has occurred in INTGUARD. Check the value of this bit. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	GRG protection violation Error Flag 0: No GRG protection violation has occurred. 1: A GRG protection violation has occurred.

(6) MGDGRSTYPE_BKA/BKB — GRG Error Access Type Register

Access: MGDGRSTYPE_BKA and MGDGRSTYPE_BKB are read-only registers that can be read in 32-bit units.
MGDGRSTYPE_BKAL and MGDGRSTYPE_BKBL are read-only registers that can be read in 16-bit units.
MGDGRSTYPE_BKALL, MGDGRSTYPE_BKALH, MGDGRSTYPE_BKBLL and MGDGRSTYPE_BKBLH are read-only registers that can be read in 8-bit units.

Address: MGDGRSTYPE_BKA: FFC4 9048$_H$
MGDGRSTYPE_BKAL: FFC4 9048$_H$
MGDGRSTYPE_BKALL: FFC4 9048$_H$
MGDGRSTYPE_BKALH: FFC4 9049$_H$
MGDGRSTYPE_BKB: FFC4 9248$_H$
MGDGRSTYPE_BKBL: FFC4 9248$_H$
MGDGRSTYPE_BKBLL: FFC4 9248$_H$
MGDGRSTYPE_BKBLH: FFC4 9249$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.76 MGDGRSTYPE_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	The PEID of the originator of the access which led to a GRG protection violation
12 to 10	Reserved	When read, the value after reset is returned.
9 to 8	SPID[1:0]	The SPID of the originator of the access which led to a GRG protection violation
7	Reserved	When read, the value after reset is returned.
6	UM	The UM of the originator of the access which led to a GRG protection violation
5 to 0	Reserved	When read, the value after reset is returned.

(7) MGDGRSAD_BKA/BKB — GRG Error Address Register

Access: MGDGRSAD_BKA and MGDGRSAD_BKB are read-only registers that can be read in 32-bit units.

MGDGRSAD_BKAL, MGDGRSAD_BKAH, MGDGRSAD_BKBL and MGDGRSAD_BKBH are read-only registers that can be read in 16-bit units.

MGDGRSAD_BKALL, MGDGRSAD_BKALH, MGDGRSAD_BKAHL, MGDGRSAD_BKBLL, MGDGRSAD_BKBLH and MGDGRSAD_BKBHL are read-only registers that can be read in 8-bit units.

Address:
MGDGRSAD_BKA: FFC4 904C$_H$
MGDGRSAD_BKAL: FFC4 904C$_H$
MGDGRSAD_BKALL: FFC4 904C$_H$
MGDGRSAD_BKALH: FFC4 904D$_H$
MGDGRSAD_BKAH: FFC4 904E$_H$
MGDGRSAD_BKAHL: FFC4 904E$_H$
MGDGRSAD_BKB: FFC4 924C$_H$
MGDGRSAD_BKBL: FFC4 924C$_H$
MGDGRSAD_BKBLL: FFC4 924C$_H$
MGDGRSAD_BKBLH: FFC4 924D$_H$
MGDGRSAD_BKBH: FFC4 924E$_H$
MGDGRSAD_BKBHL: FFC4 924E$_H$

Value after reset: MGDGRSAD_BKA: FEE0 0000$_H$, MGDGRSAD_BKB: FEF0 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SAD[19:16]			
Value after reset	1	1	1	1	1	1	1	0	1	1	1	0/1^{*1}	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.77 MGDGRSAD_BKA/BKB Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	SAD[19:0]	Address at GRG violation was occurred.

Note 1. The value after reset differs depending on bank A or bank B.

40B.3.3 PBG/HBG

The PBG/HBG module is divided into several PBG/HBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG/HBG channel can designate the access against which a single peripheral circuit should be protected.

Each PBG/HBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG/HBG group names, and the PBG/HBG channel numbers.

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG10	00	0	ECON_NMI	WDTNMIF WDTNMIFC
		1	ECON_FEINT	FEINTF FEINTFMSK FEINTFC
		2	SL_INTC	SELB_INTC1
		3	ADCA0	All registers in ADCA0
		4	KR0	KR0KRM
		5	PORT (Group A_ISO)	Registers of P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, and AP1 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		6	PORT (Group A_AWO)	Registers of P0, P1, P2, P3, P8, AP0, and IP0 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		7	JTAG (Group A)	JP0 register (except JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)
		8	RLN30	All registers in RLIN30
		9	RLN31	All registers in RLIN31
		10	RLN32	All registers in RLIN32
		11	RLN33	All registers in RLIN33
		12	RLN34	All registers in RLIN34
		13	RLN35	All registers in RLIN35
		14	RLN36	All registers in RLIN36
		15	RLN37	All registers in RLIN37
PBG11	01	0	DNF (TAUD0)	DNFATAUD0ICTL DNFATAUD0IEN
		1	DNF (ADCA0)	DNFAADCTL0CTL DNFAADCTL0EN
		2	FCLA0 (ADCA0)	FCLA0CTLn_ADC0 (n = 0 to 2)
		3	FCLA0 (NMI)	FCLA0CTL0_NMI
		4	FCLA0 (INTPn)	FCLA0CTLn_INTPL (n = 0 to 7)
		5	FCLA0 (INTPn)	FCLA0CTLn_INTPH (n = 0 to 7)
		6	DNF (ENCA0)	DNFAENCA0ICTL DNFAENCA0IEN
		7	DNF (ADCA1)	DNFAADCTL1CTL DNFAADCTL1EN
		8	FCLA0 (ADCA1)	FCLA0CTLn_ADC1 (n=0 to 2)

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG11	01	9	DNF (TAUB0)	DNFATAUB0ICTL DNFATAUB0IEN
		10	DNF (TAUB1)	DNFATAUB1ICTL DNFATAUB1IEN
		11	Reserved area	—
		12	PORT (Group B_ISO)	Registers of P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, and AP1 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		13	PORT (Group B_AWO)	Registers of P0, P1, P2, P3, P8, AP0, and IP0 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		14	JTAG (Group B)	JP0 register (JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)
		15	FCLA0 (INTPn)	FCLA0CTLn_INTPU (n = 0 to 7)
PBG12	07	0	RLN240 (Global)	All global registers in RLN240
		1	RLN2400	All channel registers in RLN2400
		2	RLN2401	All channel registers in RLN2401
		3	RLN2402	All channel registers in RLN2402
		4	RLN2403	All channel registers in RLN2403
		5	RLN241 (Global)	All global registers in RLN241
		6	RLN2414	All channel registers in RLN2414
		7	RLN2415	All channel registers in RLN2415
		8	RLN2416	All channel registers in RLN2416
		9	RLN2417	All channel registers in RLN2417
		10	RLN242 (Global)	All global registers in RLN242
		11	RLN2428	All channel registers in RLN2428
		12	RLN2429	All channel registers in RLN2429
		13	RLN24210	All channel registers in RLN24210
		14	RLN24211	All channel registers in RLN24211
		15	DNF (RSENTn)	DNFASENTICTL DNFASENTIEN
PBG13	08	0	DCRA0	All registers in DCRA0
		1	DCRA1	All registers in DCRA1
		2	DCRA2	All registers in DCRA2
		3	DCRA3	All registers in DCRA3
		4	RIIC0	All registers in RIIC0
		5	SL_READTEST	SELB_READTEST
		6	SL_DMAC	DTFSEL_TAUD0/DTFSEL_TAUB0/DTFSEL_TAUB1
		7	GRZF	All registers in GRZF
		8	RIIC1	All registers in RIIC1
		9	Reserved area	—
		10	Reserved area	—
		11	Reserved area	—
		12	Reserved area	—
		13	RSENT0	All registers in RSENT0
		14	RSENT1	All registers in RSENT1

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG20	02	0	TAUD0	All registers in TAUD0 (except SELB_TAUD0I)
		1	SL_TAUD0	SELB_TAUD0I
		2	TAUJ0	All registers in TAUJ0 (except SELB_TAUJ0I)
		3	SL_TAUJ0	SELB_TAUJ0I
		4	RTCA0	All registers in RTCA0
		5	WDTA0	All registers in WDTA0
		6	WDTA1	All registers in WDTA1
		7	Reserved area	—
		8	PIC0	All registers in PIC0
		9	TAPA0	All registers in TAPA0
		10	ENCA0	All registers in ENCA0
		11	TAUJ1	All registers in TAUJ1
		12	TAUB0	All registers in TAUB0 (except SELB_TAUB0I)
		13	TAUB1	All registers in TAUB1 (except SELB_TAUB1I)
		14	PWBAn PWGAn PWSAn SLPWG PWGA_INTF	All registers in PWM-Diag
		15	SL_TAUJ2	SELB_TAUJ2I
PBG21	09	0	Reserved area	—
		1	Reserved area	—
		2	Reserved area	—
		3	Reserved area	—
		4	Reserved area	—
		5	SL_TAUB0	SELB_TAUB0I
		6	SL_TAUB1	SELB_TAUB1I
		7	Reserved area	—
		8	TAUJ2	All registers in TAUJ2 (except SELB_TAUJ2I)
		9	TAUJ3	All registers in TAUJ3

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG30	03	0	Reserved area	—
		1	Reserved area	—
		2	Reserved area	—
		3	Reserved area	—
		4	Reserved area	—
		5	Reserved area	—
		6	Reserved area	—
		7	Reserved area	—
		8	Reserved area	—
		9	Reserved area	—
		10	ETNB0	All registers in ETNB0 (except ETNB0IFCTL)
		11	Reserved area	—
		12	ADCA1	All registers in ADCA1
		13	ECCETNB0TX	All register in ECCETNB0TX group
		14	ECCETNB0RX	All register in ECCETNB0RX group
PBG31	04	0	OSTM0	All registers in OSTM0
		1	OSTM1-4	All registers in OSTM1, OSTM2, OSTM3, and OSTM4
		2	Reserved area	—
		3	Reserved area	—
		4	ECCCSIH0	All registers in ECC CSIH0
		5	ECCCSIH1	All registers in ECC CSIH1
		6	ECCCSIH2	All registers in ECC CSIH2
		7	ECCCSIH3	All registers in ECC CSIH3
		8	Reserved area	—
		9	Reserved area	—
		10	Reserved area	—
		11	Reserved area	—
		12	ECCFLXA0	All register in ECCFLXA0 group
		13	ECCFLXA0T0	All register in ECCFLXA0T0 group
		14	ECCFLXA0T1	All register in ECCFLXA0T1 group

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG32	05	0	CSIH0 (Group A)	CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU
		1	CSIH0 (Group B)	CSIH0 registers other than the above
		2	CSIH1 (Group A)	CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU
		3	CSIH1 (Group B)	CSIH1 registers other than the above
		4	CSIH2 (Group A)	CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU
		5	CSIH2 (Group B)	CSIH2 registers other than the above
		6	CSIH3 (Group A)	CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU
		7	CSIH3 (Group B)	CSIH3 registers other than the above
		8	CSIG0 (Group A)	CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU
		9	CSIG0 (Group B)	CSIG0 registers other than the above
		10	CSIG1 (Group A)	CSIG1CTL0-2, CSIG1STR0, CSIG1STCR0, CSIG1EMU
		11	CSIG1 (Group B)	CSIG1 registers other than the above
		12	CSIG2 (Group A)	CSIG2CTL0-2, CSIG2STR0, CSIG2STCR0, CSIG2EMU
		13	CSIG2 (Group B)	CSIG2 registers other than the above
		14	CSIG3 (Group A)	CSIG3CTL0-2, CSIG3STR0, CSIG3STCR0, CSIG3EMU
		15	CSIG3 (Group B)	CSIG3 registers other than the above
PBG40	10	0	Flash memory (DCIB)	EEPRDCYCL
		1	DFECC	DFECCCTL, DFERSTR, DFERSTC, DFOVFSTR, DFOVFSTC, DFERRINT, DFEADR, DFTSTCTL
PBG50	06	0	System control*1	All registers in Write-Protect Function, Reset Controller, Power Supply Circuit, Supply Voltage Monitor, Clock Controller, Clock Monitor, Stand-By Controller, and Low-Power Sampler*1 (except STBC0PSC, STBC0STPT, SWRESA, PROTCMD0, PROTS0, JPPCMD0, JPPROTS0, PPCMD0-3, PPCMD8-13, PPCMD18-22, PPROTS0-3, PPROTS8-13, PPROTS18-22, FLMDPCMD, FLMDPS)
		1	STBC0	STBC0PSC, STBC0STPT
		2	Reserved area	—
		3	Reserved area	—
		4	RESCTL	SWRESA
		5	Flash memory (Self Programming)	FLMD, *2
		6	Flash memory (Control)	—*2
		7	Flash memory (SCDS)	PRDNAME1-3, CHIPID1-2
		8	WPROTR	PROTCMD0 PROTS0

Table 40B.78 PBG/HBG Groups and Channels and Target Modules

PBG/HBG Group	Group No. (Register Name index)*3	PBG/HBG Channel Number	Protection Target Module	Target Register
PBG60	11	0	RCFDC0 (channel 0)	All registers in RCFDC0 Ch0 group*4
		1	RCFDC0 (channel 1)	All registers in RCFDC0 Ch1 group*4
		2	RCFDC0 (channel 2)	All registers in RCFDC0 Ch2 group*4
		3	RCFDC0 (channel 3)	All registers in RCFDC0 Ch3 group*4
		4	RCFDC0 (channel 4)	All registers in RCFDC0 Ch4 group*4
		5	RCFDC0 (channel 5)	All registers in RCFDC0 Ch5 group*4
		6	RCFDC0 (channel 6)	All registers in RCFDC0 Ch6 group*4
		7	RCFDC0 (channel 7)	All registers in RCFDC0 Ch7 group*4
		8	RCFDC0 (Global)	All registers in RCFDC0 Global group*4
		9	ECCCFD0MB	All registers in ECCCFD0MB
		10	ECCCFD0AFL0	All registers in ECCCFD0AFL0
		11	ECCCFD0AFL1	All registers in ECCCFD0AFL1
HBG00	00	0	MEMC0	All registers in MEMC0
		1	MEMC0_CS0	External Memory Area(CS0)
		2	MEMC0_CS1	External Memory Area(CS1)
		3	MEMC0_CS2	External Memory Area(CS2)
		4	MEMC0_CS3	External Memory Area(CS3)
HBG01	01	0	SFMA0	All registers in SFMA0
		1	SFMA0_MEM	External Serial Flash Memory Area
HBG02	02	0	FLXA0	All registers in FLXA0

Note 1. For details, see **Section 5, Write-Protected Registers**, **Section 9BC, Reset Controller of RH850/F1KM**, **Section 10B, Power Supply Circuit of RH850/F1KM-S4**, **Section 11BC, Supply Voltage Monitor of RH850/F1KM**, **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**, **Section 13, Clock Monitor (CLMA)**, **Section 14, Stand-By Controller (STBC)**, and **Section 15, Low-Power Sampler (LPS)**.

Note 2. Regarding the PBG registers for the flash memory, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Note 3. Regarding the PBG register addresses, see **Table 40B.79, List of PBG/HBG Protection Registers**.

Note 4. Regarding the RS-CANFD guard group, see **Section 24, CANFD Interface (RS-CANFD)**.

NOTE

Be sure to enable PBG/HBG before disabling register access clock of each clock domain.

40B.3.3.1 List of Registers

The following table lists the registers provided for each PBG/HBG group. And PBG/HBG group is equal to module name.

Table 40B.79 List of PBG/HBG Protection Registers

PBG/ HBG Group[*1]	Group No.[*2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG10	00	FSGD00PROT0	PBG00 protection register 0	R/W	066F FFF7$_H$	FFC4 0000$_H$	8/16/32	ISO
		FSGD00PROT1	PBG00 protection register 1	R/W	066F FFF7$_H$	FFC4 0004$_H$	8/16/32	
		FSGD00PROT2	PBG00 protection register 2	R/W	066F FFF7$_H$	FFC4 0008$_H$	8/16/32	
		FSGD00PROT3	PBG00 protection register 3	R/W	066F FFF7$_H$	FFC4 000C$_H$	8/16/32	
		FSGD00PROT4	PBG00 protection register 4	R/W	066F FFF7$_H$	FFC4 0010$_H$	8/16/32	
		FSGD00PROT5	PBG00 protection register 5	R/W	066F FFF7$_H$	FFC4 0014$_H$	8/16/32	
		FSGD00PROT6	PBG00 protection register 6	R/W	066F FFF7$_H$	FFC4 0018$_H$	8/16/32	
		FSGD00PROT7	PBG00 protection register 7	R/W	066F FFF7$_H$	FFC4 001C$_H$	8/16/32	
		FSGD00PROT8	PBG00 protection register 8	R/W	066F FFF7$_H$	FFC4 0020$_H$	8/16/32	
		FSGD00PROT9	PBG00 protection register 9	R/W	066F FFF7$_H$	FFC4 0024$_H$	8/16/32	
		FSGD00PROT10	PBG00 protection register 10	R/W	066F FFF7$_H$	FFC4 0028$_H$	8/16/32	
		FSGD00PROT11	PBG00 protection register 11	R/W	066F FFF7$_H$	FFC4 002C$_H$	8/16/32	
		FSGD00PROT12	PBG00 protection register 12	R/W	066F FFF7$_H$	FFC4 0030$_H$	8/16/32	
		FSGD00PROT13	PBG00 protection register 13	R/W	066F FFF7$_H$	FFC4 0034$_H$	8/16/32	
		FSGD00PROT14	PBG00 protection register 14	R/W	066F FFF7$_H$	FFC4 0038$_H$	8/16/32	
		FSGD00PROT15	PBG00 protection register 15	R/W	066F FFF7$_H$	FFC4 003C$_H$	8/16/32	
PBG11	01	FSGD01PROT0	PBG01 protection register 0	R/W	066F FFF7$_H$	FFC4 0100$_H$	8/16/32	ISO
		FSGD01PROT1	PBG01 protection register 1	R/W	066F FFF7$_H$	FFC4 0104$_H$	8/16/32	
		FSGD01PROT2	PBG01 protection register 2	R/W	066F FFF7$_H$	FFC4 0108$_H$	8/16/32	
		FSGD01PROT3	PBG01 protection register 3	R/W	066F FFF7$_H$	FFC4 010C$_H$	8/16/32	
		FSGD01PROT4	PBG01 protection register 4	R/W	066F FFF7$_H$	FFC4 0110$_H$	8/16/32	
		FSGD01PROT5	PBG01 protection register 5	R/W	066F FFF7$_H$	FFC4 0114$_H$	8/16/32	
		FSGD01PROT6	PBG01 protection register 6	R/W	066F FFF7$_H$	FFC4 0118$_H$	8/16/32	
		FSGD01PROT7	PBG01 protection register 7	R/W	066F FFF7$_H$	FFC4 011C$_H$	8/16/32	
		FSGD01PROT8	PBG01 protection register 8	R/W	066F FFF7$_H$	FFC4 0120$_H$	8/16/32	
		FSGD01PROT9	PBG01 protection register 9	R/W	066F FFF7$_H$	FFC4 0124$_H$	8/16/32	
		FSGD01PROT10	PBG01 protection register 10	R/W	066F FFF7$_H$	FFC4 0128$_H$	8/16/32	
		—	Reserved	—	—	FFC4 012C$_H$	—	
		FSGD01PROT12	PBG01 protection register 12	R/W	066F FFF7$_H$	FFC4 0130$_H$	8/16/32	
		FSGD01PROT13	PBG01 protection register 13	R/W	066F FFF7$_H$	FFC4 0134$_H$	8/16/32	
		FSGD01PROT14	PBG01 protection register 14	R/W	066F FFF7$_H$	FFC4 0138$_H$	8/16/32	
		FSGD01PROT15	PBG01 protection register 15	R/W	066F FFF7$_H$	FFC4 013C$_H$	8/16/32	

Table 40B.79 List of PBG/HBG Protection Registers

PBG/ HBG Group*[1]	Group No.*[2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG12	07	FSGD07PROT0	PBG07 protection register 0	R/W	066F FFF7$_H$	FFC4 0400$_H$	8/16/32	ISO
		FSGD07PROT1	PBG07 protection register 1	R/W	066F FFF7$_H$	FFC4 0404$_H$	8/16/32	
		FSGD07PROT2	PBG07 protection register 2	R/W	066F FFF7$_H$	FFC4 0408$_H$	8/16/32	
		FSGD07PROT3	PBG07 protection register 3	R/W	066F FFF7$_H$	FFC4 040C$_H$	8/16/32	
		FSGD07PROT4	PBG07 protection register 4	R/W	066F FFF7$_H$	FFC4 0410$_H$	8/16/32	
		FSGD07PROT5	PBG07 protection register 5	R/W	066F FFF7$_H$	FFC4 0414$_H$	8/16/32	
		FSGD07PROT6	PBG07 protection register 6	R/W	066F FFF7$_H$	FFC4 0418$_H$	8/16/32	
		FSGD07PROT7	PBG07 protection register 7	R/W	066F FFF7$_H$	FFC4 041C$_H$	8/16/32	
		FSGD07PROT8	PBG07 protection register 8	R/W	066F FFF7$_H$	FFC4 0420$_H$	8/16/32	
		FSGD07PROT9	PBG07 protection register 9	R/W	066F FFF7$_H$	FFC4 0424$_H$	8/16/32	
		FSGD07PROT10	PBG07 protection register 10	R/W	066F FFF7$_H$	FFC4 0428$_H$	8/16/32	
		FSGD07PROT11	PBG07 protection register 11	R/W	066F FFF7$_H$	FFC4 042C$_H$	8/16/32	
		FSGD07PROT12	PBG07 protection register 12	R/W	066F FFF7$_H$	FFC4 0430$_H$	8/16/32	
		FSGD07PROT13	PBG07 protection register 13	R/W	066F FFF7$_H$	FFC4 0434$_H$	8/16/32	
		FSGD07PROT14	PBG07 protection register 14	R/W	066F FFF7$_H$	FFC4 0438$_H$	8/16/32	
		FSGD07PROT15	PBG07 protection register 15	R/W	066F FFF7$_H$	FFC4 043C$_H$	8/16/32	
PBG13	08	FSGD08PROT0	PBG08 protection register 0	R/W	066F FFF7$_H$	FFC4 0500$_H$	8/16/32	ISO
		FSGD08PROT1	PBG08 protection register 1	R/W	066F FFF7$_H$	FFC4 0504$_H$	8/16/32	
		FSGD08PROT2	PBG08 protection register 2	R/W	066F FFF7$_H$	FFC4 0508$_H$	8/16/32	
		FSGD08PROT3	PBG08 protection register 3	R/W	066F FFF7$_H$	FFC4 050C$_H$	8/16/32	
		FSGD08PROT4	PBG08 protection register 4	R/W	066F FFF7$_H$	FFC4 0510$_H$	8/16/32	
		FSGD08PROT5	PBG08 protection register 5	R/W	066F FFF7$_H$	FFC4 0514$_H$	8/16/32	
		FSGD08PROT6	PBG08 protection register 6	R/W	066F FFF7$_H$	FFC4 0518$_H$	8/16/32	
		FSGD08PROT7	PBG08 protection register 7	R/W	066F FFF7$_H$	FFC4 051C$_H$	8/16/32	
		FSGD08PROT8	PBG08 protection register 8	R/W	066F FFF7$_H$	FFC4 0520$_H$	8/16/32	
		—	Reserved	—	—	FFC4 0524$_H$	—	
		—	Reserved	—	—	FFC4 0528$_H$	—	
		—	Reserved	—	—	FFC4 052C$_H$	—	
		—	Reserved	—	—	FFC4 0530$_H$	—	
		FSGD08PROT13	PBG08 protection register 13	R/W	066F FFF7$_H$	FFC4 0534$_H$	8/16/32	
		FSGD08PROT14	PBG08 protection register 14	R/W	066F FFF7$_H$	FFC4 0538$_H$	8/16/32	

Table 40B.79 List of PBG/HBG Protection Registers

PBG/ HBG Group*[1]	Group No.*[2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG20	02	FSGD02PROT0	PBG02 protection register 0	R/W	066F FFF7$_H$	FFDD D000$_H$	8/16/32	ISO
		FSGD02PROT1	PBG02 protection register 1	R/W	066F FFF7$_H$	FFDD D004$_H$	8/16/32	
		FSGD02PROT2	PBG02 protection register 2	R/W	066F FFF7$_H$	FFDD D008$_H$	8/16/32	
		FSGD02PROT3	PBG02 protection register 3	R/W	066F FFF7$_H$	FFDD D00C$_H$	8/16/32	
		FSGD02PROT4	PBG02 protection register 4	R/W	066F FFF7$_H$	FFDD D010$_H$	8/16/32	
		FSGD02PROT5	PBG02 protection register 5	R/W	066F FFF7$_H$	FFDD D014$_H$	8/16/32	
		FSGD02PROT6	PBG02 protection register 6	R/W	0607 FE77$_H$	FFDD D018$_H$	8/16/32	
		—	Reserved	—	—	FFDD D01C$_H$	—	
		FSGD02PROT8	PBG02 protection register 8	R/W	066F FFF7$_H$	FFDD D020$_H$	8/16/32	
		FSGD02PROT9	PBG02 protection register 9	R/W	066F FFF7$_H$	FFDD D024$_H$	8/16/32	
		FSGD02PROT10	PBG02 protection register 10	R/W	066F FFF7$_H$	FFDD D028$_H$	8/16/32	
		FSGD02PROT11	PBG02 protection register 11	R/W	066F FFF7$_H$	FFDD D02C$_H$	8/16/32	
		FSGD02PROT12	PBG02 protection register 12	R/W	066F FFF7$_H$	FFDD D030$_H$	8/16/32	
		FSGD02PROT13	PBG02 protection register 13	R/W	066F FFF7$_H$	FFDD D034$_H$	8/16/32	
		FSGD02PROT14	PBG02 protection register 14	R/W	066F FFF7$_H$	FFDD D038$_H$	8/16/32	
		FSGD02PROT15	PBG02 protection register 15	R/W	066F FFF7$_H$	FFDD D03C$_H$	8/16/32	
PBG21	09	—	Reserved	—	—	FFDD D100$_H$	—	ISO
		—	Reserved	—	—	FFDD D104$_H$	—	
		—	Reserved	—	—	FFDD D108$_H$	—	
		—	Reserved	—	—	FFDD D10C$_H$	—	
		—	Reserved	—	—	FFDD D110$_H$	—	
		FSGD09PROT5	PBG09 protection register 5	R/W	066F FFF7$_H$	FFDD D114$_H$	8/16/32	
		FSGD09PROT6	PBG09 protection register 6	R/W	066F FFF7$_H$	FFDD D118$_H$	8/16/32	
		—	Reserved	—	—	FFDD D11C$_H$	—	
		FSGD09PROT8	PBG09 protection register 8	R/W	066F FFF7$_H$	FFDD D120$_H$	8/16/32	
		FSGD09PROT9	PBG09 protection register 9	R/W	066F FFF7$_H$	FFDD D124$_H$	8/16/32	
PBG30	03	—	Reserved	—	—	FFF9 4000$_H$	—	ISO
		—	Reserved	—	—	FFF9 4004$_H$	—	
		—	Reserved	—	—	FFF9 4008$_H$	—	
		—	Reserved	—	—	FFF9 400C$_H$	—	
		—	Reserved	—	—	FFF9 4010$_H$	—	
		—	Reserved	—	—	FFF9 4014$_H$	—	
		—	Reserved	—	—	FFF9 4018$_H$	—	
		—	Reserved	—	—	FFF9 401C$_H$	—	
		—	Reserved	—	—	FFF9 4020$_H$	—	
		—	Reserved	—	—	FFF9 4024$_H$	—	
		FSGD03PROT10	PBG03 protection register 10	R/W	066F FFF7$_H$	FFF9 4028$_H$	8/16/32	
		—	Reserved	—	—	FFF9 402C$_H$	—	
		FSGD03PROT12	PBG03 protection register 12	R/W	066F FFF7$_H$	FFF9 4030$_H$	8/16/32	
		FSGD03PROT13	PBG03 protection register 13	R/W	066F FFF7$_H$	FFF9 4034$_H$	8/16/32	
		FSGD03PROT14	PBG03 protection register 14	R/W	066F FFF7$_H$	FFF9 4038$_H$	8/16/32	

Table 40B.79 List of PBG/HBG Protection Registers

PBG/ HBG Group*1	Group No.*2	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG31	04	FSGD04PROT0	PBG04 protection register 0	R/W	066F FFF7$_H$	FFF9 4100$_H$	8/16/32	ISO
		FSGD04PROT1	PBG04 protection register 1	R/W	0607 FE77$_H$	FFF9 4104$_H$	8/16/32	
		—	Reserved	—	—	FFF9 4108$_H$	—	
		—	Reserved	—	—	FFF9 410C$_H$	—	
		FSGD04PROT4	PBG04 protection register 4	R/W	066F FFF7$_H$	FFF9 4110$_H$	8/16/32	
		FSGD04PROT5	PBG04 protection register 5	R/W	066F FFF7$_H$	FFF9 4114$_H$	8/16/32	
		FSGD04PROT6	PBG04 protection register 6	R/W	066F FFF7$_H$	FFF9 4118$_H$	8/16/32	
		FSGD04PROT7	PBG04 protection register 7	R/W	066F FFF7$_H$	FFF9 411C$_H$	8/16/32	
		—	Reserved	—	—	FFF9 4120$_H$	—	
		—	Reserved	—	—	FFF9 4124$_H$	—	
		—	Reserved	—	—	FFF9 4128$_H$	—	
		—	Reserved	—	—	FFF9 412C$_H$	—	
		FSGD04PROT12	PBG04 protection register 12	R/W	066F FFF7$_H$	FFF9 4130$_H$	8/16/32	
		FSGD04PROT13	PBG04 protection register 13	R/W	066F FFF7$_H$	FFF9 4134$_H$	8/16/32	
		FSGD04PROT14	PBG04 protection register 14	R/W	066F FFF7$_H$	FFF9 4138$_H$	8/16/32	
PBG32	05	FSGD05PROT0	PBG05 protection register 0	R/W	066F FFF7$_H$	FFF9 4200$_H$	8/16/32	ISO
		FSGD05PROT1	PBG05 protection register 1	R/W	066F FFF7$_H$	FFF9 4204$_H$	8/16/32	
		FSGD05PROT2	PBG05 protection register 2	R/W	066F FFF7$_H$	FFF9 4208$_H$	8/16/32	
		FSGD05PROT3	PBG05 protection register 3	R/W	066F FFF7$_H$	FFF9 420C$_H$	8/16/32	
		FSGD05PROT4	PBG05 protection register 4	R/W	066F FFF7$_H$	FFF9 4210$_H$	8/16/32	
		FSGD05PROT5	PBG05 protection register 5	R/W	066F FFF7$_H$	FFF9 4214$_H$	8/16/32	
		FSGD05PROT6	PBG05 protection register 6	R/W	066F FFF7$_H$	FFF9 4218$_H$	8/16/32	
		FSGD05PROT7	PBG05 protection register 7	R/W	066F FFF7$_H$	FFF9 421C$_H$	8/16/32	
		FSGD05PROT8	PBG05 protection register 8	R/W	066F FFF7$_H$	FFF9 4220$_H$	8/16/32	
		FSGD05PROT9	PBG05 protection register 9	R/W	066F FFF7$_H$	FFF9 4224$_H$	8/16/32	
		FSGD05PROT10	PBG05 protection register 10	R/W	066F FFF7$_H$	FFF9 4228$_H$	8/16/32	
		FSGD05PROT11	PBG05 protection register 11	R/W	066F FFF7$_H$	FFF9 422C$_H$	8/16/32	
		FSGD05PROT12	PBG05 protection register 12	R/W	066F FFF7$_H$	FFF9 4230$_H$	8/16/32	
		FSGD05PROT13	PBG05 protection register 13	R/W	066F FFF7$_H$	FFF9 4234$_H$	8/16/32	
		FSGD05PROT14	PBG05 protection register 14	R/W	066F FFF7$_H$	FFF9 4238$_H$	8/16/32	
		FSGD05PROT15	PBG05 protection register 15	R/W	066F FFF7$_H$	FFF9 423C$_H$	8/16/32	
PBG40	10	FSGD10PROT0	PBG10 protection register 0	R/W	066F FFF7$_H$	FFC5 9C00$_H$	8/16/32	ISO
		FSGD10PROT1	PBG10 protection register 1	R/W	066F FFF7$_H$	FFC5 9C04$_H$	8/16/32	
PBG50	06	FSGD06PROT0	PBG06 protection register 0	R/W	066F FFF7$_H$	FFF9 0000$_H$	8/16/32	ISO
		FSGD06PROT1	PBG06 protection register 1	R/W	0647 FF77$_H$	FFF9 0004$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0008$_H$	—	
		—	Reserved	—	—	FFF9 000C$_H$	—	
		FSGD06PROT4	PBG06 protection register 4	R/W	066F FFF7$_H$	FFF9 0010$_H$	8/16/32	
		FSGD06PROT5	PBG06 protection register 5	R/W	066F FFF7$_H$	FFF9 0014$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0018$_H$	—	
		FSGD06PROT7	PBG06 protection register 7	R/W	066F FFF7$_H$	FFF9 001C$_H$	8/16/32	
		FSGD06PROT8	PBG06 protection register 8	R/W	066F FFF7$_H$	FFF9 0020$_H$	8/16/32	

Table 40B.79 List of PBG/HBG Protection Registers

PBG/ HBG Group*[1]	Group No.*[2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG60	11	FSGD11PROT0	PBG11 protection register 0	R/W	066F FFF7$_H$	FFC7 1800$_H$	8/16/32	ISO
		FSGD11PROT1	PBG11 protection register 1	R/W	066F FFF7$_H$	FFC7 1804$_H$	8/16/32	
		FSGD11PROT2	PBG11 protection register 2	R/W	066F FFF7$_H$	FFC7 1808$_H$	8/16/32	
		FSGD11PROT3	PBG11 protection register 3	R/W	066F FFF7$_H$	FFC7 180C$_H$	8/16/32	
		FSGD11PROT4	PBG11 protection register 4	R/W	066F FFF7$_H$	FFC7 1810$_H$	8/16/32	
		FSGD11PROT5	PBG11 protection register 5	R/W	066F FFF7$_H$	FFC7 1814$_H$	8/16/32	
		FSGD11PROT6	PBG11 protection register 6	R/W	066F FFF7$_H$	FFC7 1818$_H$	8/16/32	
		FSGD11PROT7	PBG11 protection register 7	R/W	066F FFF7$_H$	FFC7 181C$_H$	8/16/32	
		FSGD11PROT8	PBG11 protection register 8	R/W	066F FFF7$_H$	FFC7 1820$_H$	8/16/32	
		FSGD11PROT9	PBG11 protection register 9	R/W	066F FFF7$_H$	FFC7 1824$_H$	8/16/32	
		FSGD11PROT10	PBG11 protection register 10	R/W	066F FFF7$_H$	FFC7 1828$_H$	8/16/32	
		FSGD11PROT11	PBG11 protection register 11	R/W	066F FFF7$_H$	FFC7 182C$_H$	8/16/32	
HBG00	00	HFSGD00PROT0	HBG00 protection register 0	R/W	066F FFF7$_H$	FFF9 C000$_H$	8/16/32	ISO
		HFSGD00PROT1	HBG00 protection register 1	R/W	066F FFF7$_H$	FFF9 C004$_H$	8/16/32	
		HFSGD00PROT2	HBG00 protection register 2	R/W	066F FFF7$_H$	FFF9 C008$_H$	8/16/32	
		HFSGD00PROT3	HBG00 protection register 3	R/W	066F FFF7$_H$	FFF9 C00C$_H$	8/16/32	
		HFSGD00PROT4	HBG00 protection register 4	R/W	066F FFF7$_H$	FFF9 C010$_H$	8/16/32	
HBG01	01	HFSGD01PROT0	HBG01 protection register 0	R/W	066F FFF7$_H$	FFF9 C100$_H$	8/16/32	ISO
		HFSGD01PROT1	HBG01 protection register 1	R/W	066F FFF7$_H$	FFF9 C104$_H$	8/16/32	
HBG02	02	HFSGD02PROT0	HBG02 protection register 0	R/W	066F FFF7$_H$	FFF9 C200$_H$	8/16/32	ISO

Note 1. The Group indicates the module name.

Note 2. Register name index.

The following table lists the registers provided for each PBG/HBG group.

Table 40B.80 List of PBG/HBG Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGxx	ERRSLVxxCTL	PBGxx error control register	W	0000 0000_H	<base_addr0> + 0_H	8/16/32
	ERRSLVxxSTAT	PBGxx error status register	R	0000 0000_H	<base_addr0> + 4_H	8/16/32
	ERRSLVxxADDR	PBGxx error address register	R	0000 0000_H	<base_addr0> + 8_H	32
	ERRSLVxxTYPE	PBGxx error type register	R	0000 0000_H	<base_addr0> + C_H	16/32
HBGxx	HERRSLVxxCTL	HBGxx error control register	W	0000 0000_H	<base_addr1> + 0_H	8/16/32
	HERRSLVxxSTAT	HBGxx error status register	R	0000 0000_H	<base_addr1> + 4_H	8/16/32
	HERRSLVxxADDR	HBGxx error address register	R	0000 0000_H	<base_addr1> + 8_H	32
	HERRSLVxxTYPE	HBGxx error type register	R	0000 0000_H	<base_addr1> + C_H	16/32

In the above table, “xx” in the register names and symbols represents the PBG group numbers. The table below shows the base address values <base_addr0>, which correspond to each of the PBG group numbers and channel numbers as well as base address values <base_addr1>, which correspond to each of the HBG group numbers and channel numbers.

Table 40B.81 PBG Group Numbers and Error Base Addresses

PBG Group	PBG Group Number	<base_addr0>
PBG10	00	FFC4 0040_H
PBG11	01	FFC4 0140_H
PBG12	07	FFC4 0440_H
PBG13	08	FFC4 0540_H
PBG20	02	FFDD $D040_H$
PBG21	09	FFDD $D140_H$
PBG30	03	FFF9 4040_H
PBG31	04	FFF9 4140_H
PBG32	05	FFF9 4240_H
PBG40	10	FFC5 $9C40_H$
PBG50	06	FFF9 0040_H
PBG60	11	FFC7 1840_H

Table 40B.82 HBG Group Numbers and Error Base Addresses

HBG Group	HBG Group Number	<base_addr1>
HBG00	00	FFF9 $C040_H$
HBG01	01	FFF9 $C140_H$
HBG02	02	FFF9 $C240_H$

40B.3.3.2 Details of Registers

(1) FSGDxxPROTn — PBGxx Protection Register n
HFSGDxxPROTn — HBGxx Protection Register n

FSGDxxPROTn and HFSGDxxPROTn specify the access to be rejected for protecting the target peripheral circuit control registers. Any access that is disabled with any of the identifiers is rejected as an illegal access. "n" in the register names and symbols represents the PBG/HBG channel number.

Access:	FSGDxxPROTn and HFSGDxxPROTn can be read or written in 32-bit units. FSGDxxPROTnL, FSGDxxPROTnH, HFSGDxxPROTnL, and HFSGDxxPROTnH can be read or written in 16-bit units. FSGDxxPROTnLL, FSGDxxPROTnHL, FSGDxxPROTnHH, HFSGDxxPROTnLL, HFSGDxxPROTnHL, and HFSGDxxPROTnHH can be read or written in 8-bit units.
Address:	See **Table 40B.79, List of PBG/HBG Protection Registers**.
Value after reset:	See **Table 40B.79, List of PBG/HBG Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	—	—	—	PROTPEID4	—	—	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	*1	*1	*1	*1	*1	*1	*1	*1	1
R/W	R/W	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	*1	*1	*1	*1	1	0	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Note 1. It varies depending on each register. See **Section 40B.3.3.1, List of Registers**.

Table 40B.83 FSGDxxPROTn/HFSGxxPROTn Register Contents

Bit Position	Bit Name	Function
31	PROTLOCK	Register Lock 0: Enables FSGDxxPROTn/HFSGDxxPROTn rewrite. 1: Disables FSGDxxPROTn/HFSGDxxPROTn rewrite. When PROTLOCK is set to 1, the value is held until reset is asserted.
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PROTPEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20,19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PROTPEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PROTSPID3	Access with SPID = 3*2 0: Disables access with SPID3. 1: Enables access with SPID3.

Table 40B.83 FSGDxxPROTn/HFSGxxPROTn Register Contents

Bit Position	Bit Name	Function
7	PROTSPID2	Access with SPID = 2*[2] 0: Disables access with SPID2. 1: Enables access with SPID2.
6	PROTSPID1	Access with SPID = 1 (CPU1*[3])*[2] 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*[2] 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVxxCTL — PBGxx Error Control Register
HERRSLVxxCTL — HBGxx Error Control Register

ERRSLVxxCTL and HERRSLVxxCTL clear the status in the error status register with the PBGxx/HBGxx.

Access: ERRSLVxxCTL and HERRSLVxxCTL are write-only registers that can be written in 32-bit units.
ERRSLVxxCTLL and HERRSLVxxCTLL are write-only registers that can be written in 16-bit units.
ERRSLVxxCTLLL and HERRSLVxxCTLLL are write-only registers that can be written in 8-bit units.

Address: ERRSLVxxCTL: <base_addr0> + 0_H
ERRSLVxxCTLL: <base_addr0> + 0_H
ERRSLVxxCTLLL: <base_addr0> + 0_H
HERRSLVxxCTL: <base_addr1> + 0_H
HERRSLVxxCTLL: <base_addr1> + 0_H
HERRSLVxxCTLLL: <base_addr1> + 0_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40B.84 ERRSLVxxCTL/HERRSLVxxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40B.85 CLRO and CLRE in ERRSLVxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVxxSTAT — PBGxx Error Status Register HERRSLVxxSTAT — HBGxx Error Status Register

ERRSLVxxSTAT and HERRSLVxxSTAT hold the status of the illegal access rejected with the PBGxx/HBGxx.

Access: ERRSLVxxSTAT and HERRSLVxxSTAT are read-only registers that can be read in 32-bit units.
ERRSLVxxSTATL and HERRSLVxxSTATL are read-only registers that can be read in 16-bit units.
ERRSLVxxSTATLL and HERRSLVxxSTATLL are read-only registers that can be read in 8-bit units.

Address: ERRSLVxxSTAT: <base_addr0> + 4_H
ERRSLVxxSTATL: <base_addr0> + 4_H
ERRSLVxxSTATLL: <base_addr0> + 4_H
HERRSLVxxSTAT: <base_addr1> + 4_H
HERRSLVxxSTATL: <base_addr1> + 4_H
HERRSLVxxSTATLL: <base_addr1> + 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.86 ERRSLVxxSTAT/HERRSLVxxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBG/HBG error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBG/HBG protection violation 1: PBG/HBG protection violation occurred.

CAUTION

After writing HERRSLVxxCTL, the value of HERRSLVxxSTAT takes the following procedure for a reflection:

- Execute SYNCM
- Wait for 24 × CPU clock*[1] cycles

Note 1. CPU clock: Clock selected by CKSC_CPUCLKS_CTRL and CKSC_CPUCLKD_CTL

(4) ERRSLVxxADDR — PBGxx Error Address Register
HERRSLVxxADDR — HBGxx Error Address Register

ERRSLVxxADDR holds the address of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.
HERRSLVxxADDR holds the address of the illegal access rejected with the HBGxx. The register is not updated when corresponding HERRSLVxxSTAT.ERR is 1.

Access: ERRSLVxxADDR and HERRSLVxxADDR are read-only registers that can be read in 32-bit units.

Address: ERRSLVxxADDR: <base_addr0> + 8_H
HERRSLVxxADDR: <base_addr1> + 8_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.87 ERRSLVxxADDR/HERRSLVxxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBG/HBG protection violation is generated.
1, 0	Reserved	ERRSLVxxADDR: When read, the value after reset is returned. HERRSLVxxADDR: When read, an undefined value is returned.

(5) ERRSLVxxTYPE — PBGxx Error Type Register HERRSLVxxTYPE — HBGxx Error Type Register

ERRSLVxxTYPE holds the type of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.

HERRSLVxxTYPE holds the type of the illegal access rejected with the HBGxx. The register is not updated when corresponding HERRSLVxxSTAT.ERR is 1.

Access:	ERRSLVxxTYPE and HERRSLVxxTYPE are read-only registers that can be read in 32-bit units. ERRSLVxxTYPEL and HERRSLVxxTYPEL are read-only registers that can be read in 16-bit units.
Address:	ERRSLVxxTYPE: <base_addr0> + C$_H$ ERRSLVxxTYPEL: <base_addr0> + C$_H$ HERRSLVxxTYPE: <base_addr1> + C$_H$ HERRSLVxxTYPEL: <base_addr1> + C$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.88 ERRSLVxxTYPE/HERRSLVxxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBG/HBG protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBG/HBG protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBG/HBG protection violation is generated.
5	Reserved	When read, the value after reset is returned.
4 to 1	Reserved	These bits are read as an undefined value.
0	WRITE	This bit is set to 1 when the access that has generated the PBG/HBG protection violation is a write access.

40B.3.4 PBG for CPU System

The PBGC module is divided into two PBGC groups, PBGC0 and PBGC1. PBGC0 group contains protection registers for INTC2 and DMA functions. PBGC1 group contains protection registers for ECC control function etc. Each PBGC group holds the information of the access that has been rejected.

The following table lists the target registers to be protected and the corresponding PBGC group names.

Table 40B.89 Target Registers of PBG for CPU Subsystem

PBGC Group	Group No.	Channel Number	Protection Target Module	Target Register
PBGC0	C0 (PBC group 0 for CPU system)	0	INTC2	ICxxx (xxx = 32 to 376) IMRm (m = 1 to 11) (described in **Section 7BC, Exception/Interrupt of RH850/F1KM**)
		1	PDMA0	All registers inside DMA controller (described in **Section 8, DMA Controller**)

Table 40B.89 Target Registers of PBG for CPU Subsystem

PBGC Group	Group No.	Channel Number	Protection Target Module	Target Register
PBGC1	C1 (PBC group 1 for CPU system)	0	Flash memory (Programing function)	BFASELR*[1]
		1	Code flash ECC control register (VCI)	CFECCCTL_VCI CFERRINT_VCI CFSTCLR_VCI CFOVFSTR_VCI CF1STERSTR_VCI CF1STEADR0_VCI CFSTSTCTL_VCI
		2	Code flash ECC control register (PE1)	CFECCCTL_PE1 CFERRINT_PE1 CFSTCLR_PE1 CFOVFSTR_PE1 CF1STERSTR_PE1 CF1STEADR0_PE1
		3	Local RAM ECC control register (PE1)	LRTSTCTL_PE1 LRTDATBF0_PE1 LRECCCTL_PE1 LRERRINT_PE1 LRSTCLR_PE1 LROVFSTR_PE1 LR1STERSTR_PE1 LR1STEADR0_PE1
		4	On-Chip Debug module	EPC (described in **Section 43, On-Chip Debug Unit (OCD)**)
		5	Buffer controller	FBUFCCTL (described in **Section 3BC, CPU System of RH850/F1KM**)
		6	Global RAM ECC Control register (bank A)	GRECCCTL_BKA GRERRINT_BKA GRSTCLR_BKA GROVFSTR_BKA GR1STERSTR_BKA GR1STEADR_BKA GRTSTCTL_BKA GRDECINBF1_BKA
		7	Global RAM ECC Control register (bank B)	GRECCCTL_BKB GRERRINT_BKB GRSTCLR_BKB GROVFSTR_BKB GR1STERSTR_BKB GR1STEADR_BKB GRTSTCTL_BKB GRDECINBF1_BKB

Note 1. Regarding the PBGC registers for the flash memory, see *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

40B.3.4.1 List of Registers

The following table lists the registers provided for each PBGC group. And PBG group is equal to module name.

Table 40B.90 List of PBGC Protection Registers

PBG Group*[1]	Group No.	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBGC0	C0	FSGDC0PROT0	PBGC0 protection register 0	R/W	07FF FFFF$_H$	FFC4 C000$_H$	8/16/32	ISO
		FSGDC0PROT1	PBGC0 protection register 1	R/W	07FF FFFF$_H$	FFC4 C004$_H$	8/16/32	
PBGC1	C1	FSGDC1PROT0	PBGC1 protection register 0	R/W	07FF FFFF$_H$	FFC4 C120$_H$	8/16/32	ISO
		FSGDC1PROT1	PBGC1 protection register 1	R/W	07FF FFFF$_H$	FFC4 C124$_H$	8/16/32	
		FSGDC1PROT2	PBGC1 protection register 2	R/W	07FF FFFF$_H$	FFC4 C128$_H$	8/16/32	
		FSGDC1PROT3	PBGC1 protection register 3	R/W	07FF FFFF$_H$	FFC4 C12C$_H$	8/16/32	
		FSGDC1PROT4	PBGC1 protection register 4	R/W	07FF FFFF$_H$	FFC4 C130$_H$	8/16/32	
		FSGDC1PROT5	PBGC1 protection register 5	R/W	07FF FFFF$_H$	FFC4 C134$_H$	8/16/32	
		FSGDC1PROT6	PBGC1 protection register 6	R/W	07FF FFFF$_H$	FFC4 C118$_H$	8/16/32	
		FSGDC1PROT7	PBGC1 protection register 7	R/W	07FF FFFF$_H$	FFC4 C11C$_H$	8/16/32	

Note 1. The Group indicates the module name.

Table 40B.91 List of PBGC Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGC0	ERRSLVC0CTL	PBGC0 error control register	W	0000 0000$_H$	FFC4 C800$_H$	8/16/32
	ERRSLVC0STAT	PBGC0 error status register	R	0000 0000$_H$	FFC4 C804$_H$	8/16/32
	ERRSLVC0ADDR	PBGC0 error address register	R	0000 0000$_H$	FFC4 C808$_H$	32
	ERRSLVC0TYPE	PBGC0 error type register	R	0000 0000$_H$	FFC4 C80C$_H$	16/32
PBGC1	ERRSLVC1CTL	PBGC1 error control register	W	0000 0000$_H$	FFC4 C900$_H$	8/16/32
	ERRSLVC1STAT	PBGC1 error status register	R	0000 0000$_H$	FFC4 C904$_H$	8/16/32
	ERRSLVC1ADDR	PBGC1 error address register	R	0000 0000$_H$	FFC4 C908$_H$	32
	ERRSLVC1TYPE	PBGC1 error type register	R	0000 0000$_H$	FFC4 C90C$_H$	16/32

40B.3.4.2 Details of Registers

(1) FSGDCxPROTn — PBGCx Protection Register n (x = 0, 1)

FSGDCxPROTn specifies the access to be rejected for protecting the target registers. Any access that is disabled with any of the identifiers is rejected as an illegal access.

"n" in the register names and symbols represents the PBGC channel number.

Access:	FSGDCxPROTn register can be read or written in 32-bit units. FSGDCxPROTnL and FSGDCxPROTnH registers can be read or written in 16-bit units. FSGDCxPROTnLL, FSGDCxPROTnHL, and FSGDCxPROTnHH registers can be read or written in 8-bit units.
Address:	See **Table 40B.90, List of PBGC Protection Registers**.
Value after reset:	See **Table 40B.90, List of PBGC Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	—	—	—	PROTPEID4	—	—	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 40B.92 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
31	PROTLOCK	Lock of write to this register 0: Register can be re-written. 1: Any further write to this register is ignored. This bit can be cleared by RESET.
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PROTPEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20, 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PROTPEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PROTSPID3	Access with SPID = 3*2 0: Disables access with SPID3. 1: Enables access with SPID3.
7	PROTSPID2	Access with SPID = 2*2 0: Disables access with SPID2. 1: Enables access with SPID2.

Table 40B.92 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
6	PROTSPID1	Access with SPID = 1 (CPU1*3)*2 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*2 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVCxCTL — PBGCx Error Control Register (x = 0, 1)

ERRSLVCxCTL clears the status in error status register PBGCx.

Access: ERRSLVCxCTL register is a write-only register that can be written in 32-bit units.
ERRSLVCxCTLL register is a write-only register that can be written in 16-bit units.
ERRSLVCxCTLLL register is a write-only register that can be written in 8-bit units.

Address: ERRSLVCxCTL: FFC4 C800$_H$ + (100$_H$ × x)
ERRSLVCxCTLL: FFC4 C800$_H$ + (100$_H$ × x)
ERRSLVCxCTLLL: FFC4 C800$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40B.93 ERRSLVCxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40B.94 CLRO and CLRE in ERRSLVCxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVCxSTAT — PBGCx Error Status Register (x = 0, 1)

ERRSLVCxSTAT holds the status of the illegal access rejected with the PBGCx.

Access:	ERRSLVCxSTAT register is a read-only register that can be read in 32-bit units. ERRSLVCxSTATL register is a read-only register that can be read in 16-bit units. ERRSLVCxSTATLL register is a read-only register that can be read in 8-bit units.
Address:	ERRSLVCxSTAT: FFC4 C804$_H$ + (100$_H$ × x) ERRSLVCxSTATL: FFC4 C804$_H$ + (100$_H$ × x) ERRSLVCxSTATLL: FFC4 C804$_H$ + (100$_H$ × x)
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.95 ERRSLVCxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBGC error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBGC protection violation 1: A PBGC protection violation occurred.

(4) ERRSLVCxADDR — PBGCx Error Address Register (x = 0, 1)

ERRSLVCxADDR holds the address of the illegal access rejected with the PBGCx.

The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access: ERRSLVCxADDR register is a read-only register that can be read in 32-bit units.

Address: FFC4 C808$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.96 ERRSLVCxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBGC protection violation is generated.
1, 0	Reserved	When read, the value after reset is returned.

(5) ERRSLVCxTYPE — PBGCx Error Type Register (x = 0, 1)

ERRSLVCxTYPE holds the type of the illegal access rejected with the PBGCx.

The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access:	ERRSLVCxTYPE register is a read-only register that can be read in 32-bit units. ERRSLVCxTYPEL register is a read-only register that can be read in 16-bit units.
Address:	ERRSLVCxTYPE: FFC4 C80C$_H$ + (100$_H$ × x) ERRSLVCxTYPEL: FFC4 C80C$_H$ + (100$_H$ × x)
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40B.97 ERRSLVCxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBGC protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBGC protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBGC protection violation is generated.
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	This bit is set to 1 when an access that has generated the PBGC protection violation is the write.

Section 40C Functional Safety of RH850/F1KM-S1

This section provides an overview of the safety mechanisms included in the RH850/F1KM Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance with ISO26262.

For more information about the development process and safety mechanisms, please contact our sales office.

The following are the failure detection functions provided by this microcontroller.

40C.1 Overview

ECC

Detects failures of memories and data transfer paths and corrects some types of failures.

Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements from erroneous access.

Clock Monitor

Monitors the clock operation to detect abnormal operations. For details, see **Section 13, Clock Monitor (CLMA)**.

Data CRC

Generates CRC to detect data errors.

For details, see **Section 41, Data CRC (DCRA)**.

Write-Protected Registers

The write-protected registers are protected from inadvertent write access due to erroneous program execution.

For details, see **Section 5, Write-Protected Registers**.

40C.2 ECC

40C.2.1 Overview

This product incorporates an ECC for the following memories. The ECC enables detection and correction of errors of the data retained in the memories. The ECC also enables detection and correction of errors produced between the ECC encoder and memories and between memories and ECC decoder.

Table 40C.1 ECC Overview

Applicable Memory		Applicable Data Width [bits]	Operation upon Error Detection					Failure Insertion
			Detection/ Correction	SYSERR	Interrupt Notice	Error Status	Address Capture	
Code flash		128	SEC-DED	SED/DED*2	SEC-DED	Possible	Possible	Possible
Data flash		32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
Local RAM (CPU1/Self) Retention RAM (CPU1/Self)		32	SEC-DED	SED/DED*2	SEC-DED	Possible	Possible	Possible
Peripheral RAM*1	CSIH	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible
	RS-CANFD	32	SEC-DED	—	SEC-DED	Possible	Possible	Possible

Note 1. For details of ECC for each peripheral IP, see **Section 40C.2.5, ECC for Peripheral RAM**.

Note 2. For details, see **Section 40C.2.2.2, Interrupt Requests** and **Section 40C.2.4.2, Interrupt Requests**.

Applicable Data Width

This is the data width to be ECC encoded.

To write data with a smaller data width than shown, the following processing is required. ECC is also performed for a read in (1).

(1) Reading data to be ECC-encoded including data to be rewritten

(2) Replacing data to be rewritten

(3) Writing back data generated in (2)

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected. SED-DED: 1-bit errors and 2-bit errors can only be detected.

SYSERR

SYSERR can be generated upon error detection.

Interrupt Notice

An interrupt can be generated upon error detection.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

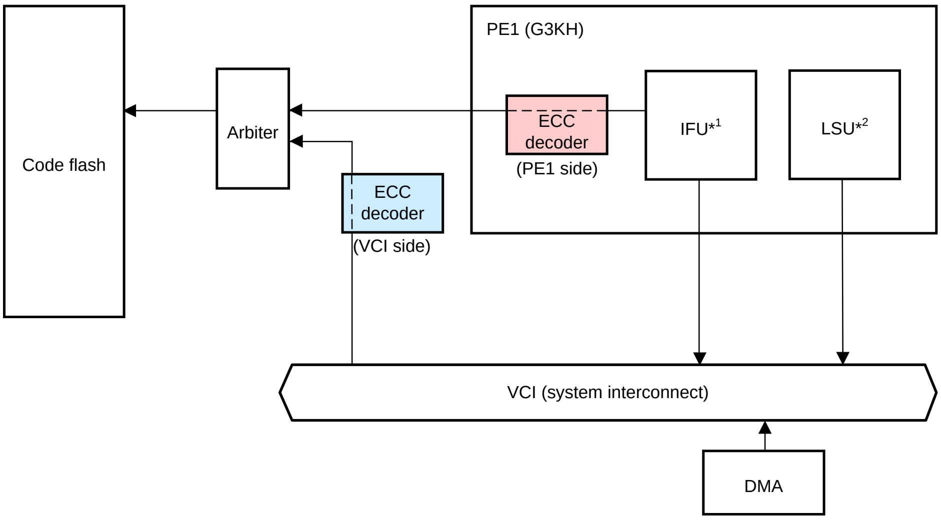
Self-diagnosis of the ECC decoder error notification function can be performed by using an intentionally generated ECC error.

40C.2.2 Code Flash ECC

40C.2.2.1 Overview

RH850/F1KM has two code flash ECC decoder circuits implemented, one inside the Processor Element (PE1) and one on the VCI (system interconnect).

Figure 40C.1, Block Diagram of Code Flash ECC shows the location of two ECC decoders for the code flash.



Note 1. Instruction Fetch Unit

Note 2. Load Store Unit

Figure 40C.1 Block Diagram of Code Flash ECC

The code flash ECC is summarized in the table below.

Table 40C.2 Summary of Code Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. (This is not supported in CFERRINT_PE1 register.) Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3BC, CPU System of RH850/F1KM**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.

Table 40C.2 Summary of Code Flash ECC

Item	Description
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The ECC 1-bit error status is set only when no error status has been set. The ECC 2-bit error status is set even when the ECC 1-bit error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error is also captured.
Self-diagnosis	The ECC bit can be read directly.
Inhibiting instruction execution	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching avoids the execution of incorrect instructions.

40C.2.2.2 Interrupt Requests

Interrupt requests for code flash ECC are listed below.

Table 40C.3 Code Flash ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0*[1] (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR	—

Note 1. ECCFLI0FEIF flag set can be read (not jump to handler address of FEINT) in SYSERR processing because return from SYSERR is not possible.

Table 40C.4 Code Flash ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	SYSERR, INTECCFLI0 (SED & SECDIS = 1)	—
		INTECCFLI0 (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of code flash	SYSERR, INTECCFLI0	—

Table 40C.5 Code Flash ECC Interrupt Requests (During Bus Master Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	INTECCFLI0	—
—	ECC 2-bit error interrupt of code flash	INTECCFLI0	—

40C.2.2.3 List of Registers

Table 40C.6 List of Registers

Module Name	Register Name	Symbol*[1]	Address
CFECC_VCI	Code flash ECC control register (VCI)	CFECCCTL_VCI	FFC6 2200_H
	Code flash error information control register (VCI)	CFERRINT_VCI	FFC6 2204_H
	Code flash status clear register (VCI)	CFSTCLR_VCI	FFC6 2208_H
	Code flash error count overflow status register (VCI)	CFOVFSTR_VCI	FFC6 $220C_H$
	Code flash 1st error status register (VCI)	CF1STERSTR_VCI	FFC6 2210_H
	Code flash 1st error address register (VCI)	CF1STEADR0_VCI	FFC6 2250_H
CFECC_CPU1	Code flash ECC control register (PE1)	CFECCCTL_PE1	FFC6 2400_H
	Code flash error information control register (PE1)	CFERRINT_PE1	FFC6 2404_H
	Code flash status clear register (PE1)	CFSTCLR_PE1	FFC6 2408_H
	Code flash error count overflow status register (PE1)	CFOVFSTR_PE1	FFC6 $240C_H$
	Code flash 1st error status register (PE1)	CF1STERSTR_PE1	FFC6 2410_H
	Code flash 1st error address register (PE1)	CF1STEADR0_PE1	FFC6 2450_H
CFECC_VCI	Code flash sub-test control register (VCI)	CFSTSTCTL_VCI	FFC6 2350_H

Note 1. The registers suffixed with symbols "_VCI" and "_PE1" are provided to ECC controllers corresponding to each access port: registers with "_VCI" are provided to the ECC controller for data access from the system interconnect to the code flash and registers with "_PE1" are provided to the ECC controller for fetch access from the CPU1.

40C.2.2.4 Details of Registers

(1) CFECCCTL_VCI/PE1 — Code Flash ECC Control Register

CFECCCTL_VCI/PE1 enables or disables ECC error detection and correction and 1-bit error correction. When writing to CFECCCTL_VCI/PE1, PROT1 and PROT0 need to be 01_{B}.

Access:	CFECCCTL_VCI and CFECCCTL_PE1 can be read or written in 32-bit units. CFECCCTL_VCIL and CFECCCTL_PE1L can be read or written in 16-bit units.
Address:	CFECCCTL_VCI: FFC6 2200_{H} CFECCCTL_VCIL: FFC6 2200_{H} CFECCCTL_PE1: FFC6 2400_{H} CFECCCTL_PE1L: FFC6 2400_{H}
Value after reset:	0000 0000_{H}

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.7 CFECCCTL_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFECCCTL_VCI/PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) CFERRINT_VCI/PE1 — Code Flash Error Information Control Register

CFERRINT_VCI/PE1 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access:	CFERRINT_VCI and CFERRINT_PE1 can be read or written in 32-bit units. CFERRINT_VCIL and CFERRINT_PE1L can be read or written in 16-bit units. CFERRINT_VCILL and CFERRINT_PE1LL can be read or written in 8-bit units.
Address:	CFERRINT_VCI: FFC6 2204$_H$ CFERRINT_VCIL: FFC6 2204$_H$ CFERRINT_VCILL: FFC6 2204$_H$ CFERRINT_PE1: FFC6 2404$_H$ CFERRINT_PE1L: FFC6 2404$_H$ CFERRINT_PE1LL: FFC6 2404$_H$
Value after reset:	0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE*[1]	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.8 CFERRINT_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE*[1]	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

Note 1. This bit is not supported in CFERRINT_PE1. When writing to this bit in CFERRINT_PE1, always write 1.

Note 2. Regarding the interrupt request, see **Section 40C.2.2.2, Interrupt Requests**.

(3) CFSTCLR_VCI/PE1 — Code Flash Status Clear Register

CFSTCLR_VCI/PE1 clears the error flags in the error status register (CF1STERSTR_VCI/PE1), the overflow flag in the error overflow status register (CFOVFSTR_VCI/PE1), and the error address register (CF1STEADR0_VCI/PE1).

Access: CFSTCLR_VCI and CFSTCLR_PE1 are write-only registers that can be written in 32-bit units.
CFSTCLR_VCIL and CFSTCLR_PE1L are write-only registers that can be written in 16-bit units.
CFSTCLR_VCILL and CFSTCLR_PE1LL are write-only registers that can be written in 8-bit units.

Address: CFSTCLR_VCI: FFC6 2208$_H$
CFSTCLR_VCIL: FFC6 2208$_H$
CFSTCLR_VCILL: FFC6 2208$_H$

CFSTCLR_PE1: FFC6 2408$_H$
CFSTCLR_PE1L: FFC6 2408$_H$
CFSTCLR_PE1LL: FFC6 2408$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40C.9 CFSTCLR_VCI/_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Clear 0: No effect (Setting 0 does not affect the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1; ERROVF0 flag in CFOVFSTR_VCI/PE1; and CF1STEADR0_VCI/PE1.) 1: Writing 1 to this bit clears the DEDF0 and SEDF0 flags in CF1STERSTR_VCI/PE1; ERROVF0 flag in CFOVFSTR_VCI/PE1; and CF1STEADR0_VCI/PE1.

(4) CFOVFSTR_VCI/PE1 — Code Flash Error Count Overflow Status Register

CFOVFSTR_VCI/PE1 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

Access: CFOVFSTR_VCI and CFOVFSTR_PE1 are read-only registers that can be read in 32-bit units.
CFOVFSTR_VCIL and CFOVFSTR_PE1L are read-only registers that can be read in 16-bit units.
CFOVFSTR_VCILL and CFOVFSTR_PE1LL are read-only registers that can be read in 8-bit units.

Address: CFOVFSTR_VCI: FFC6 220C$_H$
CFOVFSTR_VCIL: FFC6 220C$_H$
CFOVFSTR_VCILL: FFC6 220C$_H$

CFOVFSTR_PE1: FFC6 240C$_H$
CFOVFSTR_PE1L: FFC6 240C$_H$
CFOVFSTR_PE1LL: FFC6 240C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.10 CFOVFSTR_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 shows whether a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register has occurred, except when both of the error address and source of the second error are the same as those of the first error. 0: Did not occur. 1: Occurred. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1.

(5) CF1STERSTR_VCI/PE1 — Code Flash 1st Error Status Register

CF1STERSTR_VCI/PE1 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0. If a 2-bit ECC error occurs while the 1-bit ECC error flag is set, the 2-bit ECC error flag is set while retaining the 1-bit ECC error flag.

Access:	CF1STERSTR_VCI and CF1STERSTR_PE1 are read-only registers that can be read in 32-bit units. CF1STERSTR_VCIL and CF1STERSTR_PE1L are read-only registers that can be read in 16-bit units. CF1STERSTR_VCILL and CF1STERSTR_PE1LL are read-only registers that can be read in 8-bit units.
Address:	CF1STERSTR_VCI: FFC6 2210$_H$ CF1STERSTR_VCIL: FFC6 2210$_H$ CF1STERSTR_VCILL: FFC6 2210$_H$ CF1STERSTR_PE1: FFC6 2410$_H$ CF1STERSTR_PE1L: FFC6 2410$_H$ CF1STERSTR_PE1LL: FFC6 2410$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.11 CF1STERSTR_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1. [Setting condition] ECC 2-bit error is detected when DEDF0 is 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected when DEDF0 flag is 0. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1. [Setting condition] ECC 1-bit error is detected when DEDF0, SEDF0 are 0.

(6) CF1STEADR0_VCI/PE1 — Code Flash 1st Error Address Register

CF1STEADR0_VCI/PE1 holds the address at which an error has occurred.

The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated.

In addition, the EADR[24:4] bits in this register correspond to the [24:4] bits of the real address. The real address can be calculated by adding the upper address [31:25] bits, to which code flash is mapped, as a base address.

The CF1STEADR0_VCI/PE1 register is cleared by an internal reset, the external reset, or by setting the STCLR bit in the CFSTCLR_VCI/PE1 register to 1.

Access: CF1STEADR0_VCI and CF1STEADR0_PE1 are read-only registers that can be read in 32-bit units.
CF1STEADR0_VCIL, CF1STEADR0_VCIH, CF1STEADR0_PE1L and CF1STEADR0_PE1H are read-only registers that can be read in 16-bit units.
CF1STEADR0_VCILL, CF1STEADR0_VCILH, CF1STEADR0_VCIHL, CF1STEADR0_VCIHH, CF1STEADR0_PE1LL, CF1STEADR0_PE1LH, CF1STEADR0_PE1HL and CF1STEADR0_PE1HH are read-only registers that can be read in 8-bit units.

Address: CF1STEADR0_VCI: FFC6 2250$_H$
CF1STEADR0_VCIL: FFC6 2250$_H$
CF1STEADR0_VCIH: FFC6 2252$_H$
CF1STEADR0_VCILL: FFC6 2250$_H$
CF1STEADR0_VCILH: FFC6 2251$_H$
CF1STEADR0_VCIHL: FFC6 2252$_H$
CF1STEADR0_VCIHH: FFC6 2253$_H$
CF1STEADR0_PE1: FFC6 2450$_H$
CF1STEADR0_PE1L: FFC6 2450$_H$
CF1STEADR0_PE1H: FFC6 2452$_H$
CF1STEADR0_PE1LL: FFC6 2450$_H$
CF1STEADR0_PE1LH: FFC6 2451$_H$
CF1STEADR0_PE1HL: FFC6 2452$_H$
CF1STEADR0_PE1HH: FFC6 2453$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	EADR[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.12 CF1STEADR0_VCI/PE1 Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	EADR[24:4]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR_VCI/PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error has been already occurred, the address is not updated. [Clearing condition] Set the STCLR0 bit in CFSTCLR_VCI/PE1 to 1.
3 to 0	Reserved	When read, the value after reset is returned.

(7) CFSTSTCTL_VCI — Code Flash Sub-Test Control Register

CFSTSTCTL_VCI is used for the ECC test (self-diagnosis). This register is dedicated for code flash. After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read directly.

When writing to CFSTSTCTL_VCI, PROT1 and PROT0 need to be 01_B.

Access: CFSTSTCTL_VCI can be read or written in 32-bit units.
CFSTSTCTL_VCIL can be read or written in 16-bit units.

Address: CFSTSTCTL_VCI: FFC6 2350_H
CFSTSTCTL_VCIL: FFC6 2350_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40C.13 CFSTSTCTL_VCI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to CFSTSTCTL_VCI.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, ECC bits can be read directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3BC, CPU System of RH850/F1KM**.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n address. The results of reading code flash are as follows:

Table 40C.14 Results of Reading Code Flash

Bit Number	Meaning	Bit Position	Description
bit[31:10]	all-0	31 to 10	These bits are always 0.
bit[9]	reserved	9	Unknown
bit[8:0]	ECC bits	8 to 0	ECC bits

40C.2.3 Data Flash ECC

40C.2.3.1 Overview

The data flash ECC is summarized in the table below.

Table 40C.15 Summary of Data Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection / correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit errors is selectable. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit error is selectable. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the Interrupt Controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no ECC error status has been set, the address at which the first ECC error occurred is captured. ECC error address is captured when both SEDF and DEDF of DFERSTR are 0.
Self-diagnosis	ROM data and the ECC bit can be read directly. Arbitrary data can be written to ROM data and the ECC bit.

40C.2.3.2 Interrupt Requests

The interrupt requests for data flash ECC are shown below.

Table 40C.16 Data Flash ECC Interrupt Requests (During Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of data flash	INTECCEEP0	—
—	ECC 2-bit error interrupt of data flash	INTECCEEP0	—

40C.2.3.3 List of Registers

Table 40C.17 List of Registers

Module Name	Register Name	Symbol	Address
DFECC	Data flash ECC control register	DFECCCTL	FFC6 2A00$_H$
	Data flash error status register	DFERSTR	FFC6 2A04$_H$
	Data flash error status clear register	DFERSTC	FFC6 2A08$_H$
	Data flash error overflow status register	DFOVFSTR	FFC6 2A0C$_H$
	Data flash error overflow status clear register	DFOVFSTC	FFC6 2A10$_H$
	Data flash error notification control register	DFERRINT	FFC6 2A14$_H$
	Data flash 1st error address register	DFEADR	FFC6 2A18$_H$
	Data flash test control register	DFTSTCTL	FFC6 2A1C$_H$

40C.2.3.4 Details of Registers

(1) DFECCCTL — Data Flash ECC Control Register

DFECCCTL enables or disables ECC error detection and correction and 1-bit error correction. When writing to DFECCCTL, PROT1 and PROT0 need to be 01_B.

Access: DFECCCTL can be read or written in 16-bit units.

Address: DFECCCTL: FFC6 $2A00_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.18 DFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCCTL.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(2) DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection and correction are enabled, and the DEDF bit is set if an ECC 2-bit error is detected.

Access: DFERSTR is a read-only register that can be read in 8-bit units.

Address: DFERSTR: FFC6 2A04$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40C.19 DFERSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 2-bit error is detected when both SEDF and DEDF are 0.
0	SEDF	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the ERRCLR bit in DFERSTC to 1. [Setting condition] ECC 1-bit error is detected when both SEDF and DEDF are 0.

(3) DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the data flash error status register.

Access: DFERSTC is a write-only register that can be written in 8-bit units.

Address: DFERSTC: FFC6 $2A08_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERRCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40C.20 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

(4) DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of data flash error overflow.

Access: DFOVFSTR is a read-only register that can be read in 8-bit units.

Address: DFOVFSTR: FFC6 $2A0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40C.21 DFOVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag ERROVF is set if an ECC error occurs while the error address register is full. 0: Did not occurred. 1: Occurred. [Clearing condition] Set the ERROVFCLR bit is set in data flash error overflow status clear register.

(5) DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the data flash error overflow flag.

Access: DFOVFSTC is a write-only register that can be written in 8-bit units.

Address: DFOVFSTC: FFC6 2A10$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVFCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40C.22 DFOVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the ERROVF flag in DFOVFSTR.) 1: The ERROVF flag in DFOVFSTR is cleared.

(6) DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: DFERRINT can be read or written in 8-bit units.

Address: DFERRINT: FFC6 2A14$_H$

Value after reset: 02$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 40C.23 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(7) DFEADR — Data Flash 1st Error Address Register

DFEADR holds the address at which an ECC error has occurred while both of the SEDF and DEDF bits in the data flash error status register are 0.

Access: DFEADR is a read-only register that can be read in 32-bit units.

Address: DFEADR: FFC6 2A18$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	DFEADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DFEADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.24 DFEADR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 2	DFEADR[20:2]	ECC Error Address DFEADR is read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address FF20 0000$_H$.
1, 0	Reserved	When read, the value after reset is returned.

(8) DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for the ECC test.

After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read.

When writing to DFTSTCTL, PROT1 and PROT0 need to be 01_B.

Access:	DFTSTCTL can be read or written in 16-bit units.
Address:	DFTSTCTL: FFC6 $2A1C_H$
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40C.25 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFTSTCTL.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Sets ECC test mode. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).

40C.2.4 Local RAM (Including the Retention RAM) ECC

40C.2.4.1 Overview

CAUTION

The retention RAM is a part of the local RAM. The ECC for the retention RAM is shared with the local RAM. Therefore, use for the retention RAM the same register as that for the local RAM.

The local RAM ECC of CPU1 is summarized in the table below.

Table 40C.26 Summary of Local RAM ECC (CPU1)

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction. ● 2-bit error detection and 1-bit error detection. When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors detected.
Error notification	A notification is sent when an ECC error occurs. ● Enabling or disabling of error notification in the case of detection of ECC 2-bit error is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit error during data access and instruction fetch. ● Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 1-bit error (SECDIS = 1) during data access and instruction fetch. For details of the SYSERR, see **Section 3BC, CPU System of RH850/F1KM**. In the initial state of ECC controller, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. However, if an interrupt is masked by the FEINTFMSK register of the interrupt controller, an interrupt processing is not executed.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Address capture	When no error status has been set, the address at which the first error occurred is captured. In addition, when the retained address source is an ECC 1-bit error or ECC 2-bit error, the address is also captured.
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data and the ECC bit can be read directly.
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

CAUTION

When ECC error detection/correction for the local RAM is enabled for access, initialize the RAM with the 32-bit length of RAM access before the RAM is used. If the RAM before initialization is read, an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

Moreover, if the RAM is not initialized with the 32-bit length (for example, initialized with 8- or 16-bit length of access), an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

40C.2.4.2 Interrupt Requests

The local RAM ECC interrupt requests are listed below.

Table 40C.27 Local RAM ECC Interrupt Requests (During CPU Fetch Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40C.28 Local RAM ECC Interrupt Requests (During CPU Data Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	SYSERR, INTECCRAM (SED & SECDIS = 1)	—
		INTECCRAM (SED & SECDIS = 0)	—
—	ECC 2-bit error interrupt of RAM	SYSERR, INTECCRAM	—

Table 40C.29 Local RAM ECC Interrupt Requests (During Data Access except CPU Access)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	INTECCRAM	—

40C.2.4.3 List of Registers

Table 40C.30 List of Registers

Module Name	Address	Symbol	Register Name	R/W	Value after Reset	Access Size
LRTST	FFC6 5004_H	LRTSTCTL_PE1	Local RAM test control register (PE1)	R/W	0000 0000_H	16/32
	FFC6 5008_H	LRTDATBF0_PE1	Local RAM test data read buffer 0 (PE1)	R	0000 0000_H	32
LRECC	FFC6 5400_H	LRECCCTL_PE1	Local RAM ECC control register (PE1)	R/W	0000 0000_H	16/32
	FFC6 5404_H	LRERRINT_PE1	Local RAM error information control register (PE1)	R/W	0000 0003_H	8/16/32
	FFC6 5408_H	LRSTCLR_PE1	Local RAM status clear register (PE1)	W	0000 0000_H	8/16/32
	FFC6 $540C_H$	LROVFSTR_PE1	Local RAM error count overflow status register (PE1)	R	0000 0000_H	8/16/32
	FFC6 5410_H	LR1STERSTR_PE1	Local RAM 1st error status register (PE1)	R	0000 0000_H	8/16/32
	FFC6 5450_H	LR1STEADR0_PE1	Local RAM 1st error address register 0 (PE1)	R	0000 0000_H	8/16/32

40C.2.4.4 Details of Registers

(1) LRTSTCTL_PE1 — Local RAM Test Control Register

LRTSTCTL_PE1 is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST = 1, any data can be written to the ECC bits. The DATSEL bit is used to select RAM data or the ECC bits. When writing to LRTSTCTL_PE1, PROT1 and PROT0 need to be 01_B.

Access: LRTSTCTL_PE1 register can be read or written in 32-bit units.
LRTSTCTL_PE1L register can be read or written in 16-bit units.

Address: LRTSTCTL_PE1: FFC6 5004_H
LRTSTCTL_PE1L: FFC6 5004_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST	DATSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.31 LRTSTCTL_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST and DATSEL bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRTSTCTL_PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST = 1, the ECC bits can be read by reading LRTDATBF0_PE1 register or written directly. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1).
0	DATSEL	Data Select This bit is valid when ECCTST = 1. This bit selects the RAM bit which can be accessed when writing. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: RAM data is selected. 1: The ECC bits are selected.

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

(2) LRTDATBF0_PE1 — Local RAM Test Data Read Buffer 0

In ECC test mode (self-diagnosis), the ECC bits can be read. If the local RAM is read while ECCTST = 1 in the local RAM test control register LRTSTCTL_PE1, reading from the local RAM reads out the ECC bits, and these bits are stored in this buffer.

Access: LRTDATBF0_PE1 register is a read-only register that can be read in 32-bit units.

Address: LRTDATBF0_PE1: FFC6 5008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	LRTDATBF[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.32 LRTDATBF0_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	LRTDATBF[6:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the local RAM, the ECC bits for the local RAM are stored in LRTDATBF[6:0].

(3) LRECCCTL_PE1 — Local RAM ECC Control Register

LRECCCTL_PE1 enables or disables ECC error detection and correction and 1-bit error correction.

When writing to LRECCCTL_PE1, PROT1 and PROT0 need to be 01_B.

Access: LRECCCTL_PE1 register can be read or written in 32-bit units.
LRECCCTL_PE1L register can be read or written in 16-bit units.

Address: LRECCCTL_PE1: FFC6 5400_H
LRECCCTL_PE1L: FFC6 5400_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.33 LRECCCTL_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRECCCTL_PE1.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables correction of a 1-bit error upon detection. 1: Disables correction of a 1-bit error upon detection.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with the setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(4) LRERRINT_PE1 — Local RAM Error Information Control Register

LRERRINT_PE1 enables or disables generation of the error notification signal to the interrupt controller upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access:	LRERRINT_PE1 register can be read or written in 32-bit units. LRERRINT_PE1L register can be read or written in 16-bit units. LRERRINT_PE1LL register can be read or written in 8-bit units.
Address:	LRERRINT_PE1: FFC6 5404$_H$ LRERRINT_PE1L: FFC6 5404$_H$ LRERRINT_PE1LL: FFC6 5404$_H$
Value after reset:	0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 40C.34 LRERRINT_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(5) LRSTCLR_PE1 — Local RAM Status Clear Register

LRSTCLR_PE1 clears the error flags in the error status register (LR1STERSTR_PE1), the overflow flag in the error overflow status register (LROVFSTR_PE1), and the error address register (LR1STEADR0_PE1). LRSTCLR_PE1 is a write-only register and is always read as 0.

Access: LRSTCLR_PE1 register is a write-only register that can be written in 32-bit units.
LRSTCLR_PE1L register is a write-only register that can be written in 16-bit units.
LRSTCLR_PE1LL register is a write-only register that can be written in 8-bit units.

Address: LRSTCLR_PE1: FFC6 5408$_H$
LRSTCLR_PE1L: FFC6 5408$_H$
LRSTCLR_PE1LL: FFC6 5408$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 40C.35 LRSTCLR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF0 and SEDF0 flags in LR1STERSTR_PE1; ERROVF0 flag in LROVFSTR_PE1; and LR1STEADR0_PE1.

(6) LROVFSTR_PE1 — Local RAM Error Count Overflow Status Register

LROVFSTR_PE1 monitors occurrence of error overflow. If a second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF0 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1.

Access: LROVFSTR_PE1 register is a read-only register that can be read in 32-bit units.
LROVFSTR_PE1L register is a read-only register that can be read in 16-bit units.
LROVFSTR_PE1LL register is a read-only register that can be read in 8-bit units.

Address: LROVFSTR_PE1: FFC6 540C$_{\text{H}}$
LROVFSTR_PE1L: FFC6 540C$_{\text{H}}$
LROVFSTR_PE1LL: FFC6 540C$_{\text{H}}$

Value after reset: 0000 0000$_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.36 LROVFSTR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

(7) LR1STERSTR_PE1 — Local RAM 1st Error Status Register

LR1STERSTR_PE1 monitors occurrence of the first error when the ECC error detection/correction is enabled. The error status is set if an error occurs while the error flag is 0.

If more than one error occurs simultaneously, all the corresponding error flags are set. LR1STERSTR_PE1 is cleared by an internal reset, the external reset, or setting 1 to the STCLR0 bit in LRSTCLR_PE1.

Access:	LR1STERSTR_PE1 register is a read-only register that can be read in 32-bit units. LR1STERSTR_PE1L register is a read-only register that can be read in 16-bit units. LR1STERSTR_PE1LL register is a read-only register that can be read in 8-bit units.
Address:	LR1STERSTR_PE1: FFC6 5410$_H$ LR1STERSTR_PE1L: FFC6 5410$_H$ LR1STERSTR_PE1LL: FFC6 5410$_H$
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.37 LR1STERSTR_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-Bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1 to 1. [Setting condition] ECC 2-bit error is detected with DEDF0 being 0.
0	SEDF0	ECC 1-Bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. [Clearing condition] Set the STCLR0 bit in LRSTCLR_PE1 to 1. [Setting condition] ECC 1-bit error is detected with both SEDF0 and DEDF0 being 0.

(8) LR1STEADR0_PE1 — Local RAM 1st Error Address Register 0

LR1STEADR0_PE1 holds the address at which an error has occurred.

The error address is set if an error occurs while error flags are 0 in LR1STERSTR_PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Since this register holds the internal address, add the base address[31:19] bits of the associated memory to transform the internal address to the real address. LR1STEADR0_PE1 is cleared by an internal reset, the external reset, or setting the STCLR0 bit to 1 in LRSTCLR_PE1.

Access: LR1STEADR0_PE1 register is a read-only register that can be read in 32-bit units.
LR1STEADR0_PE1L and LR1STEADR0_PE1H registers are read-only registers that can be read in 16-bit units.
LR1STEADR0_PE1LL, LR1STEADR0_PE1LH, and LR1STEADR0_PE1HL registers are read-only registers that can be read in 8-bit units.

Address: LR1STEADR0_PE1: FFC6 5450_H
LR1STEADR0_PE1L: FFC6 5450_H
LR1STEADR0_PE1H: FFC6 5452_H
LR1STEADR0_PE1LL: FFC6 5450_H
LR1STEADR0_PE1LH: FFC6 5451_H
LR1STEADR0_PE1HL: FFC6 5452_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	EADR[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.38 LR1STEADR0_PE1 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 2	EADR[18:2]	1st Error Address Monitors the address of the first error. The error address is held if an error occurs while all the error flags are 0 in LR1STERSTR_PE1. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.
1, 0	Reserved	When read, the value after reset is returned.

40C.2.5 ECC for Peripheral RAM

40C.2.5.1 Overview

This is an ECC module for the RAM of the following peripheral modules.

CSIH and RS-CANFD.

Table 40C.39 List of the ECC Functions for the peripheral RAM

Item	Description
ECC error detection/correction	ECC error detection and correction can be enabled or disabled. Either of the following settings can be selected. ● 2-bit error detection and 1-bit error detection/correction ● 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. In the initial state, 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. ● Error notification can be enabled or disabled when an ECC 2-bit error is detected. ● Error notification can be enabled or disabled when an ECC 1-bit error is detected. In the initial state of ECC controller, 2-bit error notification is enabled and 1-bit error notification is disabled. However, if an interrupt is masked by the FEINTFMSK register, an interrupt processing is not executed.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC1-bit errors is available. A bit for clearing the error status is provided.
Address capture	Only one address at which an ECC error has occurred can be captured. A signal is generated upon detection of ECC 2-bit or 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed about RS-CANFD, initialize the RAM area before it is used.

40C.2.5.2 Interrupt Requests

ECC of peripheral interrupt requests are listed below.

Table 40C.40 CSIHn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
INTECCCSIHn	CSIHn ECC 1-bit error or 2-bit error interrupt	INTECCCSIH0	—
		INTECCCSIH1	
		INTECCCSIH2	
		INTECCCSIH3	

Table 40C.41 RCFDCn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal	Description	Name	DMA Trigger Number
ECCCNFDRAMFEIF	RCFDC0 ECC 1-bit error or 2-bit error interrupt	INTECCCNFDRAM	—

40C.2.5.3 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 40C.42 List of ECC Modules

Peripheral Functions		Module Name	ECC Module Names and Register	
			Base Address Name	Base Address <base_addr>
CSIHn	Buffered I/O (CSIH RAM)	ECCCSIH0	<ECCCSIH0_base>	FFC7 0100$_H$
		ECCCSIH1	<ECCCSIH1_base>	FFC7 0200$_H$
		ECCCSIH2	<ECCCSIH2_base>	FFC7 0300$_H$
		ECCCSIH3	<ECCCSIH3_base>	FFC7 0400$_H$
RCFDC0	Message bufferRAM (MB RAM)	ECCCFD0MB	<ECCCFD0MB_base>	FFC7 1300$_H$
	Acceptance filter list RAM0 (AFL0 RAM)	ECCCFD0AFL0	<ECCCFD0AFL0_base>	FFC7 1400$_H$
	Acceptance filter list RAM1 (AFL1 RAM)	ECCCFD0AFL1	<ECCCFD0AFL1_base>	FFC7 1500$_H$

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 40C.43 List of Registers

Module Name	Register Name	Symbol	Address
See (1) List of ECC Modules.	ECC control register	<Module_Name>CTL	<base_addr> + 00$_H$
	ECC test mode control register	<Module_Name>TMC	<base_addr> + 04$_H$
	ECC encode/decode input/output replacement test register	<Module_Name>TED	<base_addr> + 0C$_H$
	ECC redundant bit data control test register	<Module_Name>TRC	<base_addr> + 08$_H$
	ECC decode syndrome data register	<Module_Name>SYND	<base_addr> + 0B$_H$
	ECC 7-bit redundant bit data hold test register	<Module_Name>HORD	<base_addr> + 0A$_H$
	ECC encode test register	<Module_Name>ECRD	<base_addr> + 09$_H$
	ECC redundant bit input/output replacement register	<Module_Name>ERDB	<base_addr> + 08$_H$
	ECC error address register 0	<Module_Name>AD0	<base_addr> + 10$_H$
SL_READTEST	ECCREAD test select register	SELB_READTEST	FFC7 8000$_H$

40C.2.5.4 Details of Registers

(1) <Module_Name>CTL — ECC Control Register

The <Module_Name>CTL register controls the mode of the ECC and the status for target peripheral modules.

Bits 7, 5, 4 and 3 should be set (written) while the target peripheral module’s operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access:	This register can be read or written in 16-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	$001X_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	ECCOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 40C.44 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
15	EMCA1	Access Control Bits 1 and 0 to ECC Mode Selection These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ECCOVFF	By detecting an error while the error status is set and the new error has another address than the already latched (not cleared or reset is not issued), this bit is set. 0: Overflow is not occurred after reset of clearing ECER2F and ECER1F. 1: Error address register overflowed **NOTE:** This bit clear condition is as follows. (1) Reset (2) Writing ECER2C = 1 when ECER2F = 1 or ECER1C = 1 when ECER1F = 1 (3) Selecting through mode enable (ECTHM = 1)
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority. ECER2C = 1 also clears the ECCOVFF bit while ECER2F bit is set.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority. ECER1C = 1 also clears the ECCOVFF bit while ECER1F bit is set.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40C.44 <Module_Name>CTL Register Contents

Bit Position	Bit Name	Function
7	ECTHM	ECC Function through Mode Selection Set this bit to select whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, write 01_B to the EMCA1 and EMCA0 bits at the same time. Set this bit to 1 to disable the ECC function. 0: Passing through mode is disabled (normal operation mode). 1: Passing through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 2-bit error is detected, a target peripheral module's interrupt will be generated.
3	EC1EDIC	1-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a target peripheral module's interrupt will not be generated. 1: When 1-bit error is detected, a target peripheral module's interrupt will be generated.
2	ECER2F	2-Bit Error Detection Flag This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (a target peripheral module's interrupt) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. When through mode is enable (ECTHM = 1), this bit is cleared. If 2-bit error is detected again while this bit is set, an interrupt will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
1	ECER1F	1-Bit Error Detection/Correction Flag This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. When through mode is enabled (ECTHM = 1), this bit is cleared. 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
0	ECEMF	ECC Error Message Flag This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. This bit is also cleared when through mode is enabled (ECTHM = 1) and there is no 1-bit error in decode circuit input data. 0: The current RAM output data does not have bit errors. 1: The current RAM output data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

(2) <Module_Name>TMC — ECC Test Mode Control Register

The <Module_Name>TMC register is used to switch to the test mode, and this register is for test mode control.

This register can be used when a target peripheral module is not accessed to RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access:	This register can be read or written in 16-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 40C.45 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
15	ETMA1	Access Control Bits 1 and 0 to ECC Test Mode These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to test control bits of the test registers and this register. When writing to this bit, write 10_B to the ETMA1 and ETMA0 bits at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: <Module_Name>TED, <Module_Name>TRC, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, <Module_Name>ERDB Register test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the <Module_Name>TED and <Module_Name>ERDB registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the <Module_Name>TED register will be the write value of the <Module_Name>TED register. Read value of the <Module_Name>ERDB register will be the write value of the <Module_Name>ERDB register. 1: Read value of the <Module_Name>TED register can read RAM data. Read value of the <Module_Name>ERDB register will be the ECC Data to be written to RAM.
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies which is output to the ECC to be stored in RAM, the ECC data generated for write data or the value of the <Module_Name>ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated for write data is stored in RAM. 1: The value of <Module_Name>ERDB Register is stored in RAM.

Table 40C.45 <Module_Name>TMC Register Contents

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Selection This bit specifies data written to RAM or the value of the <Module_Name>TED register as the input to the ECC encoder. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the <Module_Name>TED.
1	ECDCS	ECC Decoder Input Selection This bit specifies which data is for generation of syndrome code and error detection, RAM data or the value of <Module_Name>TED. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from RAM Data. 1: Syndrome code generation and error detection are performed from <Module_Name>TED register value.
0	ECREIS	ECC Redundant Bit Input Data Selection This bit specifies which ECC data is for generation of syndrome code and error detection, ECC data stored in RAM or the value of the <Module_Name>ERDB. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from <Module_Name>ERDB register value.

(3) <Module_Name>TED — ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

This register value is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access: This register can be read or written in 32-bit units.

Address: See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40C.46 <Module_Name>TED Register Contents

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When <Module_Name>TMC.ECENS = 1, the value of this register is used to generate ECC data and that is stored to RAM. When <Module_Name>TMC.ECDCS = 1, the value of this register is used to generate syndrome code that is stored in ECC decode syndrome data register (<Module_Name>SYND). In addition, when <Module_Name>TMC.ECTRRS = 1, RAM data [31:0] instead of written data is read for the value of this register.

(4) <Module_Name>TRC — ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and consists of four 8-bit registers, <Module_Name>SYND, <Module_Name>HORD, <Module_Name>ECRD, and <Module_Name>ERDB.

This register can be accessed when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when a target peripheral module is not accessed to RAM.

Access:	This register can be read or written in 32-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	<Module_Name>SYND (see **Section (6)**)								<Module_Name>HORD (see **Section (7)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<Module_Name>ECRD (see **Section (8)**)								<Module_Name>ERDB (see **Section (9)**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(5) <Module_Name>AD0 — Target ECC Error Address Register 0

This is read only register to hold the ECC error occurred address.

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as trigger and it is hold as the error occurring address.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ECEAD[30:16]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.47 <Module_Name>AD0 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 0	ECEAD[30:0]	ECEAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECEAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECEAD0

(6) <Module_Name>SYND — ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register is read-only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40C.48 <Module_Name>SYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

(7) <Module_Name>HORD — ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40C.49 <Module_Name>HORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. When <Module_Name>TMC.ECTRRS =1 and if <Module_Name>TED register is read, ECC code is stored.

(8) <Module_Name>ECRD — ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (<Module_Name>TMC.ECTMCE = 1). When ECC test mode is disabled (<Module_Name>TMC.ECTMCE = 0), 00_H is read.

Access:	This register is a read-only register that can be read in 8-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40C.50 <Module_Name>ECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing and can read ECC data for data written in the <Module_Name>TED register when <Module_Name>MC.ECENS =1.

(9) <Module_Name>ERDB — ECC Redundant Bit Input/Output Replacement Register

In ECC test mode, this register handles test data.

This register value can be handled as generated ECC data at the time of writing to RAM or as read ECC data at the time of reading RAM data.

This register can be accessed when ECC test mode in enabled (<Module_Name>TMC.ECTMCE = 1). When <Module_Name>TMC.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Access:	This register can be read or written in 8-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40C.51 <Module_Name>ERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	These bits can store this register value as ECC data when <Module_Name>TMC.ECREOS = 1. When the register is read while <Module_Name>TMC.ECREIS =1, the value read from these bits is ECC data read from the RAM. When <Module_Name>TMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instated of written data.

(10) SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the target peripheral module's RAM ECC registers.

Setting 1 to the bit corresponding to each function will enable writing to the read-only bit.

Access:	This register can be read or written in 16-bit units.
Address:	See **Table 40C.42, List of ECC Modules** and **Table 40C.43, List of Registers**.
Value after reset:	0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCANFDE7A03	RTCANFDE7A02	RTCANFDE7A01	—	—	—	—	—	—	—	—	—	RTCSIHE7A3	RTCSIHE7A2	RTCSIHE7A1	RTCSIHE7A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 40C.52 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
15	RTCANFDE7A03	RCFDC0 (AFL Buffer 1) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
14	RTCANFDE7A02	RCFDC0 (AFL Buffer 0) ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
13	RTCANFDE7A01	RCFDC0 (Message Buffer) ECC Register Write Access for Testing Purpose Enable /Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RCFDC0 ECC read-only bit can be written).
12 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH3 ECC read-only bit can be written).
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH2 ECC read-only bit can be written).
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH0 ECC read-only bit can be written).

40C.3 Memory Protection

40C.3.1 Overview

This product incorporates the memory protection function to prevent erroneous accesses to data in memories and control registers of the peripheral circuits.

- MPU

  The CPU protects memory against illegal access by itself. The CPU does not generate the signals for access to addresses where access is prohibited by the MPU. For details, see the *RH850 Family User's Manual: Software*.

- Slave Guard

  A specific memory is protected against illegal accesses from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.

  - PEG
    The local RAM is protected against illegal accesses. However, accesses from the CPU incorporating the local RAM itself are excluded. For example, accesses from the CPU1 to local RAM in the CPU1 are not rejected by the PEG. For details, see **Section 3BC, CPU System of RH850/F1KM**.
  - IPG
    The CPU Peripheral is protected against illegal accesses. For details, see **Section 3BC, CPU System of RH850/F1KM**.
  - PBG
    The control registers in the peripheral circuits are protected against illegal accesses. For details, see **Section 40C.3.2, PBG**.
  - PBGC
    The CPU system has its dedicated PBG function which is called a PBG for CPU system. For details, see **Section 40C.3.3, PBG for CPU System**.

40C.3.1.1 Identifiers for Slave Guard

For the slave guard function, the type of illegal accesses to be rejected can be designated using the following identifiers.

Table 40C.53 Identifiers for Slave Guard

Identifier	Function
UM	When the CPU makes an access, this indicates the operating mode of the CPU. 0: Supervisor mode 1: User mode When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 0.
SPID	When the CPU makes an access, this indicates the system protection identifier SPID that is assigned to the CPU. When the PDMA makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 00_B.
PEID	This indicates the access source bus master. 001_B: CPU1 When the PDMA makes an access, the value of this identifier is the value in the channel master setting register.

40C.3.2 PBG

The PBG module is divided into several PBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG channel can designate the access against which a single peripheral circuit should be protected.

Each PBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG group names, and the PBG channel numbers.

Table 40C.54 PBG Groups and Channels and Target Modules

PBG Group	Group No.*3	PBG Channel Number	Protection Target Module	Target Register
PBG10	00	0	ECON_NMI	WDTNMIF WDTNMIFC
		1	ECON_FEINT	FEINTF FEINTFMSK FEINTFC
		2	SL_INTC	SELB_INTC1
		3	ADCA0	All registers in ADCA0
		4	KR0	KR0KRM
		5	PORT (Group A_ISO)	Registers of P9, P10, and P11 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		6	PORT (Group A_AWO)	Registers of P0, P8, and AP0 (except Pn, PSRn, PPRn, PNOTn, and PIBCn)
		7	JTAG (Group A)	JP0 register (except JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)
		8	RLN30	All registers in RLIN30
		9	RLN31	All registers in RLIN31
		10	RLN32	All registers in RLIN32
		11	RLN33	All registers in RLIN33
PBG11	01	0	DNF (TAUD0)	DNFATAUD0ICTL DNFATAUD0IEN
		1	DNF (ADCA0)	DNFAADCTL0CTL DNFAADCTL0EN
		2	FCLA0 (ADCA0)	FCLA0CTLn_ADC0 (n = 0 to 2)
		3	FCLA0 (NMI)	FCLA0CTL0_NMI
		4	FCLA0 (INTPn)	FCLA0CTLn_INTPL (n = 0 to 7)
		5	FCLA0 (INTPn)	FCLA0CTLn_INTPH (n = 0, 2 to 5)
		6	DNF (ENCA0)	DNFAENCA0ICTL DNFAENCA0IEN
		9	DNF (TAUB0)	DNFATAUB0ICTL DNFATAUB0IEN
		12	PORT (Group B_ISO)	Registers of P9, P10, and P11 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		13	PORT (Group B_AWO)	Registers of P0, P8, and AP0 (Pn, PSRn, PPRn, PNOTn, and PIBCn)
		14	JTAG (Group B)	JP0 register (JP0, JPSR0, JPPR0, JPNOT0, and JPIBC0)

Table 40C.54 PBG Groups and Channels and Target Modules

PBG Group	Group No.*3	PBG Channel Number	Protection Target Module	Target Register
PBG12	07	0	RLN240 (Global)	All global registers in RLN240
		1	RLN2400	All channel registers in RLN2400
		2	RLN2401	All channel registers in RLN2401
		3	RLN2402	All channel registers in RLN2402
		15	DNF (RSENTn)	DNFASENTICTL DNFASENTIEN
PBG13	08	0	DCRA0	All registers in DCRA0
		1	DCRA1	All registers in DCRA1
		2	DCRA2	All registers in DCRA2
		3	DCRA3	All registers in DCRA3
		4	RIIC0	All registers in RIIC0
		5	SL_READTEST	SELB_READTEST
		6	SL_DMAC	DTFSEL_TAUD0/DTFSEL_TAUB0
		8	RIIC1	All registers in RIIC1
		9	Reserved area	—
		10	Reserved area	—
		11	Reserved area	—
		13	RSENT0	All registers in RSENT0
		14	RSENT1	All registers in RSENT1
PBG20	02	0	TAUD0	All registers in TAUD0 (except SELB_TAUD0I)
		1	SL_TAUD0	SELB_TAUD0I
		2	TAUJ0	All registers in TAUJ0 (except SELB_TAUJ0I)
		3	SL_TAUJ0	SELB_TAUJ0I
		4	RTCA0	All registers in RTCA0
		5	WDTA0	All registers in WDTA0
		6	WDTA1	All registers in WDTA1
		8	PIC0	All registers in PIC0
		9	TAPA0	All registers in TAPA0
		10	ENCA0	All registers in ENCA0
		11	TAUJ1	All registers in TAUJ1

Table 40C.54 PBG Groups and Channels and Target Modules

PBG Group	Group No.*3	PBG Channel Number	Protection Target Module	Target Register
PBG20	02	12	TAUB0	All registers in TAUB0 (except SELB_TAUB0I)
		13	Reserved area	—
		14	PWBAn PWGAn PWSAn SLPWG PWGA_INTF	All registers in PWM-Diag
		15	SL_TAUJ2	SELB_TAUJ2I
PBG21	09	0	Flash memory (DCIB)	EEPRDCYCL
		1	DFECC	DFECCCTL, DFERSTR, DFERSTC, DFOVFSTR, DFOVFSTC, DFERRINT, DFEADR, DFTSTCTL
		5	SL_TAUB0	SELB_TAUB0I
		8	TAUJ2	All registers in TAUJ2 (except SELB_TAUJ2I)
		9	TAUJ3	All registers in TAUJ3
PBG30	03	0	RCFDC0 (channel 0)	All registers in RCFDC0 Ch0 group*4
		1	RCFDC0 (channel 1)	All registers in RCFDC0 Ch1 group*4
		2	RCFDC0 (channel 2)	All registers in RCFDC0 Ch2 group*4
		3	RCFDC0 (channel 3)	All registers in RCFDC0 Ch3 group*4
		4	RCFDC0 (channel 4)	All registers in RCFDC0 Ch4 group*4
		5	RCFDC0 (channel 5)	All registers in RCFDC0 Ch5 group*4
		8	RCFDC0 (Global)	All registers in RCFDC0 Global group*4
PBG31	04	0	OSTM0	All registers in OSTM0
		4	ECCCSIH0	All registers in ECC CSIH0
		5	ECCCSIH1	All registers in ECC CSIH1
		6	ECCCSIH2	All registers in ECC CSIH2
		7	ECCCSIH3	All registers in ECC CSIH3
		8	ECCCFD0MB	All registers in ECCCFD0MB
		9	ECCCFD0AFL0	All registers in ECCCFD0AFL0
		10	ECCCFD0AFL1	All registers in ECCCFD0AFL1

Table 40C.54 PBG Groups and Channels and Target Modules

PBG Group	Group No.*3	PBG Channel Number	Protection Target Module	Target Register
PBG32	05	0	CSIH0 (Group A)	CSIH0CTL0-2, CSIH0STR0, CSIH0STCR0, CSIH0EMU
		1	CSIH0 (Group B)	CSIH0 registers other than the above
		2	CSIH1 (Group A)	CSIH1CTL0-2, CSIH1STR0, CSIH1STCR0, CSIH1EMU
		3	CSIH1 (Group B)	CSIH1 registers other than the above
		4	CSIH2 (Group A)	CSIH2CTL0-2, CSIH2STR0, CSIH2STCR0, CSIH2EMU
		5	CSIH2 (Group B)	CSIH2 registers other than the above
		6	CSIH3 (Group A)	CSIH3CTL0-2, CSIH3STR0, CSIH3STCR0, CSIH3EMU
		7	CSIH3 (Group B)	CSIH3 registers other than the above
		8	CSIG0 (Group A)	CSIG0CTL0-2, CSIG0STR0, CSIG0STCR0, CSIG0EMU
		9	CSIG0 (Group B)	CSIG0 registers other than the above
PBG50	06	0	System control*1	All registers in Write-Protect Function, Reset Controller, Power Supply Circuit, Supply Voltage Monitor, Clock Controller, Clock Monitor, Stand-By Controller, and Low-Power Sampler*1 (except STBC0PSC, STBC0STPT, SWRESA, PROTCMD0, PROTS0, JPPCMD0, JPPROTS0, PPCMD0, PPCMD8-11, PPROTS0, PPROTS8-11, FLMDPCMD, FLMDPS)
		1	STBC0	STBC0PSC, STBC0STPT
		4	RESCTL	SWRESA
		5	Flash memory (Self Programming)	FLMD, *2
		6	Flash memory (Control)	— *2
		7	Flash memory (SCDS)	PRDNAME1-3, CHIPID1-2
		8	WPROTR	PROTCMD0 PROTS0

Note 1. For details, see **Section 5, Write-Protected Registers**, **Section 9BC, Reset Controller of RH850/F1KM**, **Section 10C, Power Supply Circuit of RH850/F1KM-S1**, **Section 11BC, Supply Voltage Monitor of RH850/F1KM**, **Section 12C, Clock Controller of RH850/F1KM-S1**, **Section 13, Clock Monitor (CLMA)**, **Section 14, Stand-By Controller (STBC)**, and **Section 15, Low-Power Sampler (LPS)**.

Note 2. Regarding the PBG registers for the flash memory, refer to *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

Note 3. Regarding the PBG register addresses, see the **Table 40C.55, List of PBG Protection Registers**.

Note 4. Regarding the RS-CANFD guard group, see **Section 24, CANFD Interface (RS-CANFD)**.

NOTE

Be sure to enable PBG before disabling register access clock of each clock domain.

40C.3.2.1 List of Registers

The following table lists the registers provided for each PBG group. And PBG group is equal to module name.

Table 40C.55 List of PBG Protection Registers

PBG Group*1	Group No.*2	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG10	00	FSGD00PROT0	PBG00 protection register 0	R/W	066F FFF7$_H$	FFC4 0000$_H$	8/16/32	ISO
		FSGD00PROT1	PBG00 protection register 1	R/W	066F FFF7$_H$	FFC4 0004$_H$	8/16/32	
		FSGD00PROT2	PBG00 protection register 2	R/W	066F FFF7$_H$	FFC4 0008$_H$	8/16/32	
		FSGD00PROT3	PBG00 protection register 3	R/W	066F FFF7$_H$	FFC4 000C$_H$	8/16/32	
		FSGD00PROT4	PBG00 protection register 4	R/W	066F FFF7$_H$	FFC4 0010$_H$	8/16/32	
		FSGD00PROT5	PBG00 protection register 5	R/W	066F FFF7$_H$	FFC4 0014$_H$	8/16/32	
		FSGD00PROT6	PBG00 protection register 6	R/W	066F FFF7$_H$	FFC4 0018$_H$	8/16/32	
		FSGD00PROT7	PBG00 protection register 7	R/W	066F FFF7$_H$	FFC4 001C$_H$	8/16/32	
		FSGD00PROT8	PBG00 protection register 8	R/W	066F FFF7$_H$	FFC4 0020$_H$	8/16/32	
		FSGD00PROT9	PBG00 protection register 9	R/W	066F FFF7$_H$	FFC4 0024$_H$	8/16/32	
		FSGD00PROT10	PBG00 protection register 10	R/W	066F FFF7$_H$	FFC4 0028$_H$	8/16/32	
		FSGD00PROT11	PBG00 protection register 11	R/W	066F FFF7$_H$	FFC4 002C$_H$	8/16/32	
PBG11	01	FSGD01PROT0	PBG01 protection register 0	R/W	066F FFF7$_H$	FFC4 0100$_H$	8/16/32	ISO
		FSGD01PROT1	PBG01 protection register 1	R/W	066F FFF7$_H$	FFC4 0104$_H$	8/16/32	
		FSGD01PROT2	PBG01 protection register 2	R/W	066F FFF7$_H$	FFC4 0108$_H$	8/16/32	
		FSGD01PROT3	PBG01 protection register 3	R/W	066F FFF7$_H$	FFC4 010C$_H$	8/16/32	
		FSGD01PROT4	PBG01 protection register 4	R/W	066F FFF7$_H$	FFC4 0110$_H$	8/16/32	
		FSGD01PROT5	PBG01 protection register 5	R/W	066F FFF7$_H$	FFC4 0114$_H$	8/16/32	
		FSGD01PROT6	PBG01 protection register 6	R/W	066F FFF7$_H$	FFC4 0118$_H$	8/16/32	
		—	Reserved	—	—	FFC4 011C$_H$	—	
		—	Reserved	—	—	FFC4 0120$_H$	—	
		FSGD01PROT9	PBG01 protection register 9	R/W	066F FFF7$_H$	FFC4 0124$_H$	8/16/32	
		—	Reserved	—	—	FFC4 0128$_H$	—	
		—	Reserved	—	—	FFC4 012C$_H$	—	
		FSGD01PROT12	PBG01 protection register 12	R/W	066F FFF7$_H$	FFC4 0130$_H$	8/16/32	
		FSGD01PROT13	PBG01 protection register 13	R/W	066F FFF7$_H$	FFC4 0134$_H$	8/16/32	
		FSGD01PROT14	PBG01 protection register 14	R/W	066F FFF7$_H$	FFC4 0138$_H$	8/16/32	
PBG12	07	FSGD07PROT0	PBG07 protection register 0	R/W	066F FFF7$_H$	FFC4 0400$_H$	8/16/32	ISO
		FSGD07PROT1	PBG07 protection register 1	R/W	066F FFF7$_H$	FFC4 0404$_H$	8/16/32	
		FSGD07PROT2	PBG07 protection register 2	R/W	066F FFF7$_H$	FFC4 0408$_H$	8/16/32	
		FSGD07PROT3	PBG07 protection register 3	R/W	066F FFF7$_H$	FFC4 040C$_H$	8/16/32	
		—	Reserved	—	—	FFC4 0410$_H$	—	
		—	Reserved	—	—	FFC4 0414$_H$	—	
		—	Reserved	—	—	FFC4 0418$_H$	—	
		—	Reserved	—	—	FFC4 041C$_H$	—	
		—	Reserved	—	—	FFC4 0420$_H$	—	
		—	Reserved	—	—	FFC4 0424$_H$	—	
		—	Reserved	—	—	FFC4 0428$_H$	—	
		—	Reserved	—	—	FFC4 042C$_H$	—	
		—	Reserved	—	—	FFC4 0430$_H$	—	
		FSGD07PROT15	PBG07 protection register 15	R/W	066F FFF7$_H$	FFC4 043C$_H$	8/16/32	

Table 40C.55 List of PBG Protection Registers

PBG Group[*1]	Group No.[*2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG13	08	FSGD08PROT0	PBG08 protection register 0	R/W	066F FFF7$_H$	FFC4 0500$_H$	8/16/32	ISO
		FSGD08PROT1	PBG08 protection register 1	R/W	066F FFF7$_H$	FFC4 0504$_H$	8/16/32	
		FSGD08PROT2	PBG08 protection register 2	R/W	066F FFF7$_H$	FFC4 0508$_H$	8/16/32	
		FSGD08PROT3	PBG08 protection register 3	R/W	066F FFF7$_H$	FFC4 050C$_H$	8/16/32	
		FSGD08PROT4	PBG08 protection register 4	R/W	066F FFF7$_H$	FFC4 0510$_H$	8/16/32	
		FSGD08PROT5	PBG08 protection register 5	R/W	066F FFF7$_H$	FFC4 0514$_H$	8/16/32	
		FSGD08PROT6	PBG08 protection register 6	R/W	066F FFF7$_H$	FFC4 0518$_H$	8/16/32	
		—	Reserved	—	—	FFC4 051C$_H$	—	
		FSGD08PROT8	PBG08 protection register 8	R/W	066F FFF7$_H$	FFC4 0520$_H$	8/16/32	
		—	Reserved	—	—	FFC4 0524$_H$	—	
		—	Reserved	—	—	FFC4 0528$_H$	—	
		—	Reserved	—	—	FFC4 052C$_H$	—	
		—	Reserved	—	—	FFC4 0530$_H$	—	
		FSGD08PROT13	PBG08 protection register 13	R/W	066F FFF7$_H$	FFC4 0534$_H$	8/16/32	
		FSGD08PROT14	PBG08 protection register 14	R/W	066F FFF7$_H$	FFC4 0538$_H$	8/16/32	
PBG20	02	FSGD02PROT0	PBG02 protection register 0	R/W	066F FFF7$_H$	FFDD D000$_H$	8/16/32	ISO
		FSGD02PROT1	PBG02 protection register 1	R/W	066F FFF7$_H$	FFDD D004$_H$	8/16/32	
		FSGD02PROT2	PBG02 protection register 2	R/W	066F FFF7$_H$	FFDD D008$_H$	8/16/32	
		FSGD02PROT3	PBG02 protection register 3	R/W	066F FFF7$_H$	FFDD D00C$_H$	8/16/32	
		FSGD02PROT4	PBG02 protection register 4	R/W	066F FFF7$_H$	FFDD D010$_H$	8/16/32	
		FSGD02PROT5	PBG02 protection register 5	R/W	066F FFF7$_H$	FFDD D014$_H$	8/16/32	
		FSGD02PROT6	PBG02 protection register 6	R/W	0607 FE77$_H$	FFDD D018$_H$	8/16/32	
		—	Reserved	—	—	FFDD D01C$_H$	—	
		FSGD02PROT8	PBG02 protection register 8	R/W	066F FFF7$_H$	FFDD D020$_H$	8/16/32	
		FSGD02PROT9	PBG02 protection register 9	R/W	066F FFF7$_H$	FFDD D024$_H$	8/16/32	
		FSGD02PROT10	PBG02 protection register 10	R/W	066F FFF7$_H$	FFDD D028$_H$	8/16/32	
		FSGD02PROT11	PBG02 protection register 11	R/W	066F FFF7$_H$	FFDD D02C$_H$	8/16/32	
		FSGD02PROT12	PBG02 protection register 12	R/W	066F FFF7$_H$	FFDD D030$_H$	8/16/32	
		—	Reserved	—	—	FFDD D034$_H$	—	
		FSGD02PROT14	PBG02 protection register 14	R/W	066F FFF7$_H$	FFDD D038$_H$	8/16/32	
		FSGD02PROT15	PBG02 protection register 15	R/W	066F FFF7$_H$	FFDD D03C$_H$	8/16/32	
PBG21	09	FSGD09PROT0	PBG09 protection register 0	R/W	066F FFF7$_H$	FFDD D100$_H$	8/16/32	ISO
		FSGD09PROT1	PBG09 protection register 1	R/W	066F FFF7$_H$	FFDD D104$_H$	8/16/32	
		—	Reserved	—	—	FFDD D108$_H$	—	
		—	Reserved	—	—	FFDD D10C$_H$	—	
		—	Reserved	—	—	FFDD D110$_H$	—	
		FSGD09PROT5	PBG09 protection register 5	R/W	066F FFF7$_H$	FFDD D114$_H$	8/16/32	
		—	Reserved	—	—	FFDD D118$_H$	—	
		—	Reserved	—	—	FFDD D11C$_H$	—	
		FSGD09PROT8	PBG09 protection register 8	R/W	066F FFF7$_H$	FFDD D120$_H$	8/16/32	
		FSGD09PROT9	PBG09 protection register 9	R/W	066F FFF7$_H$	FFDD D124$_H$	8/16/32	

Table 40C.55 List of PBG Protection Registers

PBG Group*[1]	Group No.*[2]	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBG30	03	FSGD03PROT0	PBG03 protection register 0	R/W	066F FFF7$_H$	FFF9 4000$_H$	8/16/32	ISO
		FSGD03PROT1	PBG03 protection register 1	R/W	066F FFF7$_H$	FFF9 4004$_H$	8/16/32	
		FSGD03PROT2	PBG03 protection register 2	R/W	066F FFF7$_H$	FFF9 4008$_H$	8/16/32	
		FSGD03PROT3	PBG03 protection register 3	R/W	066F FFF7$_H$	FFF9 400C$_H$	8/16/32	
		FSGD03PROT4	PBG03 protection register 4	R/W	066F FFF7$_H$	FFF9 4010$_H$	8/16/32	
		FSGD03PROT5	PBG03 protection register 5	R/W	066F FFF7$_H$	FFF9 4014$_H$	8/16/32	
		—	Reserved	—	—	FFF9 4018$_H$	—	
		—	Reserved	—	—	FFF9 401C$_H$	—	
		FSGD03PROT8	PBG03 protection register 8	R/W	066F FFF7$_H$	FFF9 4020$_H$	8/16/32	
PBG31	04	FSGD04PROT0	PBG04 protection register 0	R/W	066F FFF7$_H$	FFF9 4100$_H$	8/16/32	ISO
		—	Reserved	—	—	FFF9 4104$_H$	—	
		—	Reserved	—	—	FFF9 4108$_H$	—	
		—	Reserved	—	—	FFF9 410C$_H$	—	
		FSGD04PROT4	PBG04 protection register 4	R/W	066F FFF7$_H$	FFF9 4110$_H$	8/16/32	
		FSGD04PROT5	PBG04 protection register 5	R/W	066F FFF7$_H$	FFF9 4114$_H$	8/16/32	
		FSGD04PROT6	PBG04 protection register 6	R/W	066F FFF7$_H$	FFF9 4118$_H$	8/16/32	
		FSGD04PROT7	PBG04 protection register 7	R/W	066F FFF7$_H$	FFF9 411C$_H$	8/16/32	
		FSGD04PROT8	PBG04 protection register 8	R/W	066F FFF7$_H$	FFF9 4120$_H$	8/16/32	
		FSGD04PROT9	PBG04 protection register 9	R/W	066F FFF7$_H$	FFF9 4124$_H$	8/16/32	
		FSGD04PROT10	PBG04 protection register 10	R/W	066F FFF7$_H$	FFF9 4128$_H$	8/16/32	
PBG32	05	FSGD05PROT0	PBG05 protection register 0	R/W	066F FFF7$_H$	FFF9 4200$_H$	8/16/32	ISO
		FSGD05PROT1	PBG05 protection register 1	R/W	066F FFF7$_H$	FFF9 4204$_H$	8/16/32	
		FSGD05PROT2	PBG05 protection register 2	R/W	066F FFF7$_H$	FFF9 4208$_H$	8/16/32	
		FSGD05PROT3	PBG05 protection register 3	R/W	066F FFF7$_H$	FFF9 420C$_H$	8/16/32	
		FSGD05PROT4	PBG05 protection register 4	R/W	066F FFF7$_H$	FFF9 4210$_H$	8/16/32	
		FSGD05PROT5	PBG05 protection register 5	R/W	066F FFF7$_H$	FFF9 4214$_H$	8/16/32	
		FSGD05PROT6	PBG05 protection register 6	R/W	066F FFF7$_H$	FFF9 4218$_H$	8/16/32	
		FSGD05PROT7	PBG05 protection register 7	R/W	066F FFF7$_H$	FFF9 421C$_H$	8/16/32	
		FSGD05PROT8	PBG05 protection register 8	R/W	066F FFF7$_H$	FFF9 4220$_H$	8/16/32	
		FSGD05PROT9	PBG05 protection register 9	R/W	066F FFF7$_H$	FFF9 4224$_H$	8/16/32	
PBG50	06	FSGD06PROT0	PBG06 protection register 0	R/W	066F FFF7$_H$	FFF9 0000$_H$	8/16/32	ISO
		FSGD06PROT1	PBG06 protection register 1	R/W	0647 FF77$_H$	FFF9 0004$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0008$_H$	—	
		—	Reserved	—	—	FFF9 000C$_H$	—	
		FSGD06PROT4	PBG06 protection register 4	R/W	066F FFF7$_H$	FFF9 0010$_H$	8/16/32	
		FSGD06PROT5	PBG06 protection register 5	R/W	066F FFF7$_H$	FFF9 0014$_H$	8/16/32	
		—	Reserved	—	—	FFF9 0018$_H$	—	
		FSGD06PROT7	PBG06 protection register 7	R/W	066F FFF7$_H$	FFF9 001C$_H$	8/16/32	
		FSGD06PROT8	PBG06 protection register 8	R/W	066F FFF7$_H$	FFF9 0020$_H$	8/16/32	

Note 1. The Group indicates the module name.

Note 2. Register name index.

The following table lists the registers provided for each PBG group.

Table 40C.56 List of PBG Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGxx	ERRSLVxxCTL	PBGxx error control register	W	0000 0000_H	<base_addr0> + 0_H	8/16/32
	ERRSLVxxSTAT	PBGxx error status register	R	0000 0000_H	<base_addr0> + 4_H	8/16/32
	ERRSLVxxADDR	PBGxx error address register	R	0000 0000_H	<base_addr0> + 8_H	32
	ERRSLVxxTYPE	PBGxx error type register	R	0000 0000_H	<base_addr0> + C_H	16/32

In the above table, "xx" in the register names and symbols represents the PBG group numbers. The table below shows the base address values <base_addr0>, which correspond to each of the PBG group numbers.

Table 40C.57 PBG Group Numbers and Error Base Addresses

PBG Group	PBG Group Number	<base_addr0>
PBG10	00	FFC4 0040_H
PBG11	01	FFC4 0140_H
PBG12	07	FFC4 0440_H
PBG13	08	FFC4 0540_H
PBG20	02	FFDD $D040_H$
PBG21	09	FFDD $D140_H$
PBG30	03	FFF9 4040_H
PBG31	04	FFF9 4140_H
PBG32	05	FFF9 4240_H
PBG50	06	FFF9 0040_H

40C.3.2.2 Details of Registers

(1) FSGDxxPROTn — PBGxx Protection Register n

FSGDxxPROTn specifies the access to be rejected for protecting the target peripheral circuit control registers. Any access that is disabled with any of the identifiers is rejected as an illegal access. “n” in the register names and symbols represents the PBG channel number.

Access: FSGDxxPROTn register can be read or written in 32-bit units.
FSGDxxPROTnL and FSGDxxPROTnH registers can be read or written in 16-bit units.
FSGDxxPROTnLL, FSGDxxPROTnHL and FSGDxxPROTnHH registers can be read or written in 8-bit units.

Address: See **Table 40C.55, List of PBG Protection Registers**.

Value after reset: See **Table 40C.55, List of PBG Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	—	—	—	—	—	—	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	*1	*1	*1	*1	*1	*1	*1	*1	1
R/W	R/W	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	*1	*1	*1	*1	1	0	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Note 1. It varies depending on each register. See **Section 40C.3.2.1, List of Registers**.

Table 40C.58 FSGDxxPROTn Register Contents

Bit Position	Bit Name	Function
31	PROTLOCK	Register Lock 0: Enables FSGDxxPROTn rewrite. 1: Disables FSGDxxPROTn rewrite. When PROTLOCK is set to 1, the value is held until reset is asserted.
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PROTPEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PROTSPID3	Access with SPID = 3*2 0: Disables access with SPID3. 1: Enables access with SPID3.
7	PROTSPID2	Access with SPID = 2*2 0: Disables access with SPID2. 1: Enables access with SPID2.

Table 40C.58 FSGDxxPROTn Register Contents

Bit Position	Bit Name	Function
6	PROTSPID1	Access with SPID = 1 (CPU1*3)*2 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0*2 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVxxCTL — PBGxx Error Control Register

ERRSLVxxCTL clears the status in the error status register with the PBGxx.

Access: ERRSLVxxCTL is write-only register that can be written in 32-bit units.
ERRSLVxxCTLL is write-only register that can be written in 16-bit units.
ERRSLVxxCTLLL is write-only register that can be written in 8-bit units.

Address: ERRSLVxxCTL: <base_addr0> + 0_H
ERRSLVxxCTLL: <base_addr0> + 0_H
ERRSLVxxCTLLL: <base_addr0> + 0_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40C.59 ERRSLVxxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40C.60 CLRO and CLRE in ERRSLVxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVxxSTAT — PBGxx Error Status Register

ERRSLVxxSTAT holds the status of the illegal access rejected with the PBGxx.

Access: ERRSLVxxSTAT is read-only register that can be read in 32-bit units.
ERRSLVxxSTATL is read-only register that can be read in 16-bit units.
ERRSLVxxSTATLL is read-only register that can be read in 8-bit units.

Address: ERRSLVxxSTAT: <base_addr0> + 4_H
ERRSLVxxSTATL: <base_addr0> + 4_H
ERRSLVxxSTATLL: <base_addr0> + 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.61 ERRSLVxxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBG error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBG protection violation 1: A PBG protection violation occurred.

(4) ERRSLVxxADDR — PBGxx Error Address Register

ERRSLVxxADDR holds the address of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.

Access: ERRSLVxxADDR is read-only register that can be read in 32-bit units.

Address: ERRSLVxxADDR: <base_addr0> + 8_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.62 ERRSLVxxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBG protection violation is generated.
1, 0	Reserved	When read, the value after reset is returned.

(5) ERRSLVxxTYPE — PBGxx Error Type Register

ERRSLVxxTYPE holds the type of the illegal access rejected with the PBGxx. The register is not updated when corresponding ERRSLVxxSTAT.ERR is 1.

Access: ERRSLVxxTYPE is read-only register that can be read in 32-bit units.
ERRSLVxxTYPEL is read-only register that can be read in 16-bit units.

Address: ERRSLVxxTYPE: <base_addr0> + C_H
ERRSLVxxTYPEL: <base_addr0> + C_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.63 ERRSLVxxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBG protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBG protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBG protection violation is generated.
5	Reserved	When read, the value after reset is returned.
4 to 1	Reserved	These bits are read as an undefined value.
0	WRITE	This bit is set to 1 when the access that has generated the PBG protection violation is a write access.

40C.3.3 PBG for CPU System

The PBGC module is divided into two PBGC groups, PBGC0 and PBGC1. PBGC0 group contains protection registers for INTC2 and DMA functions. PBGC1 group contains protection registers for ECC control function etc. Each PBGC group holds the information of the access that has been rejected.

The following table lists the target registers to be protected and the corresponding PBGC group names.

Table 40C.64 Target Registers of PBG for CPU Subsystem

PBGC Group	Group No.	Channel Number	Protection Target Module	Target Register
PBGC0	C0 (PBC group 0 for CPU system)	0	INTC2	ICxxx (xxx=32 to 357) IMRm (m=1 to 11) (described in **Section 7BC, Exception/Interrupts of RH850/F1KM**)
		1	PDMA0	All registers inside DMA controller (described in **Section 8, DMA Controller**)
PBGC1	C1 (PBC group 1 for CPU system)	0	Flash memory (Programing function)	BFASELR*[1]
		1	Code flash ECC control register (VCI)	CFECCCTL_VCI CFERRINT_VCI CFSTCLR_VCI CFOVFSTR_VCI CF1STERSTR_VCI CF1STEADR0_VCI CFSTSTCTL_VCI
		2	Code flash ECC control register (PE1)	CFECCCTL_PE1 CFERRINT_PE1 CFSTCLR_PE1 CFOVFSTR_PE1 CF1STERSTR_PE1 CF1STEADR0_PE1
		3	Local RAM ECC control register (PE1)	LRTSTCTL_PE1 LRTDATBF0_PE1 LRECCCTL_PE1 LRERRINT_PE1 LRSTCLR_PE1 LROVFSTR_PE1 LR1STERSTR_PE1 LR1STEADR0_PE1
		4	On-Chip Debug module	EPC (described in **Section 43, On-Chip Debug Unit (OCD)**)
		5	Buffer controller	FBUFCCTL (described in **Section 3BC, CPU System of RH850/F1KM**)

Note 1. Regarding the PBGC registers for the flash memory, refer to *the RH850/F1KH, F1KM, F1K Flash Memory User's Manual: Hardware Interface*.

40C.3.3.1 List of Registers

The following table lists the registers provided for each PBGC group. And PBG group is equal to module name.

Table 40C.65 List of PBGC Protection Registers

PBG Group*[1]	Group No.	Symbol	Register Name	R/W	Value after Reset	Address	Access Size	Power Domain
PBGC0	C0	FSGDC0PROT0	PBGC0 protection register 0	R/W	07FF $FFFF_H$	FFC4 $C000_H$	8/16/32	ISO
		FSGDC0PROT1	PBGC0 protection register 1	R/W	07FF $FFFF_H$	FFC4 $C004_H$	8/16/32	
PBGC1	C1	FSGDC1PROT0	PBGC1 protection register 0	R/W	07FF $FFFF_H$	FFC4 $C120_H$	8/16/32	ISO
		FSGDC1PROT1	PBGC1 protection register 1	R/W	07FF $FFFF_H$	FFC4 $C124_H$	8/16/32	
		FSGDC1PROT2	PBGC1 protection register 2	R/W	07FF $FFFF_H$	FFC4 $C128_H$	8/16/32	
		FSGDC1PROT3	PBGC1 protection register 3	R/W	07FF $FFFF_H$	FFC4 $C12C_H$	8/16/32	
		FSGDC1PROT4	PBGC1 protection register 4	R/W	07FF $FFFF_H$	FFC4 $C130_H$	8/16/32	
		FSGDC1PROT5	PBGC1 protection register 5	R/W	07FF $FFFF_H$	FFC4 $C134_H$	8/16/32	

Note 1. The Group indicates the module name.

Table 40C.66 List of PBGC Error Registers

Module Name	Symbol	Register Name	R/W	Value after Reset	Address	Access Size
PBGC0	ERRSLVC0CTL	PBGC0 error control register	W	0000 0000_H	FFC4 $C800_H$	8/16/32
	ERRSLVC0STAT	PBGC0 error status register	R	0000 0000_H	FFC4 $C804_H$	8/16/32
	ERRSLVC0ADDR	PBGC0 error address register	R	0000 0000_H	FFC4 $C808_H$	32
	ERRSLVC0TYPE	PBGC0 error type register	R	0000 0000_H	FFC4 $C80C_H$	16/32
PBGC1	ERRSLVC1CTL	PBGC1 error control register	W	0000 0000_H	FFC4 $C900_H$	8/16/32
	ERRSLVC1STAT	PBGC1 error status register	R	0000 0000_H	FFC4 $C904_H$	8/16/32
	ERRSLVC1ADDR	PBGC1 error address register	R	0000 0000_H	FFC4 $C908_H$	32
	ERRSLVC1TYPE	PBGC1 error type register	R	0000 0000_H	FFC4 $C90C_H$	16/32

40C.3.3.2 Details of Registers

(1) FSGDCxPROTn — PBGCx Protection Register n (x = 0, 1)

FSGDCxPROTn specifies the access to be rejected for protecting the target registers. Any access that is disabled with any of the identifiers is rejected as an illegal access.

"n" in the register names and symbols represents the PBGC channel number.

Access:	FSGDCxPROTn register can be read or written in 32-bit units. FSGDCxPROTnL and FSGDCxPROTnH registers can be read or written in 16-bit units. FSGDCxPROTnLL, FSGDCxPROTnHL and FSGDCxPROTnHH registers can be read or written in 8-bit units.
Address:	See **Table 40C.65, List of PBGC Protection Registers**.
Value after reset:	See **Table 40C.65, List of PBGC Protection Registers**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PROTUM	—	—	—	—	—	—	PROTPEID1	—	—
Value after reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PROTSPID3	PROTSPID2	PROTSPID1	PROTSPID0	—	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 40C.67 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PROTPEID1	Access with PEID = 1 (CPU1)[*1] 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PROTSPID3	Access with SPID = 3[*2] 0: Disables access with SPID3. 1: Enables access with SPID3.
7	PROTSPID2	Access with SPID = 2[*2] 0: Disables access with SPID2. 1: Enables access with SPID2.
6	PROTSPID1	Access with SPID = 1 (CPU1[*3])[*2] 0: Disables access with SPID1. 1: Enables access with SPID1.
5	PROTSPID0	Access with SPID = 0[*2] 0: Disables access with SPID0. 1: Enables access with SPID0.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40C.67 FSGDCxPROTn Register Contents

Bit Position	Bit Name	Function
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master regardless of other setting in this register. 1: The setting of PROTRD is effective.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master regardless of other setting in this register. 1: The setting of PROTWR is effective.
1	PROTRD	Read Permission 0: Disables reading by a bus master subject to access filtering. 1: Enables reading by a bus master subject to access filtering.
0	PROTWR	Write Permission 0: Disables writing by a bus master subject to access filtering. 1: Enables writing by a bus master subject to access filtering.

Note 1. Access with PEID
PROTPEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
PROTSPID is a bit list with each bit representing an SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

Note 3. Setting value of MCFG0.SPID

(2) ERRSLVCxCTL — PBGCx Error Control Register (x = 0, 1)

ERRSLVCxCTL clears the status in error status register PBGCx.

Access: ERRSLVCxCTL is a write-only register that can be written in 32-bit units.
ERRSLVCxCTLL is a write-only register that can be written in 16-bit units.
ERRSLVCxCTLLL is a write-only register that can be written in 8-bit units.

Address: ERRSLVCxCTL: FFC4 C800$_H$ + (100$_H$ × x)
ERRSLVCxCTLL: FFC4 C800$_H$ + (100$_H$ × x)
ERRSLVCxCTLLL: FFC4 C800$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 40C.68 ERRSLVCxCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears the overflow flag. 0: Does not clear the overflow flag. 1: Clears the overflow flag.
0	CLRE	Clears the error flag. 0: Does not clear the error flag. 1: Clears the error flag.

Table 40C.69 CLRO and CLRE in ERRSLVCxxCTL Register

CLRO	CLRE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(3) ERRSLVCxSTAT — PBGCx Error Status Register (x = 0, 1)

ERRSLVCxSTAT holds the status of the illegal access rejected with the PBGCx.

Access: ERRSLVCxSTAT is a read-only register that can be read in 32-bit units.
ERRSLVCxSTATL is a read-only register that can be read in 16-bit units.
ERRSLVCxSTATLL is a read-only register that can be read in 8-bit units.

Address: ERRSLVCxSTAT: FFC4 $C804_H$ + (100_H × x)
ERRSLVCxSTATL: FFC4 $C804_H$ + (100_H × x)
ERRSLVCxSTATLL: FFC4 $C804_H$ + (100_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.70 ERRSLVCxSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of PBGC error entry stages is 1. Note that this overflow is notified to INTGUARD. In addition, it is not possible to determine whether an overflow has occurred by INTGUARD. For an overflow check the OVF bit should be checked. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	Error Status Flag 0: No PBGC protection violation 1: A PBGC protection violation occurred.

(4) ERRSLVCxADDR — PBGCx Error Address Register (x = 0, 1)

ERRSLVCxADDR holds the address of the illegal access rejected with the PBGCx. The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access: ERRSLVCxADDR is read-only register that can be read in 32-bit units.

Address: FFC4 C808$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.71 ERRSLVCxADDR Register Contents

Bit Position	Bit Name	Function
31 to 2	ADDR[31:2]	Address in which the PBGC protection violation is generated.
1,0	Reserved	When read, the value after reset is returned.

(5) ERRSLVCxTYPE — PBGCx Error Type Register (x = 0, 1)

ERRSLVCxTYPE holds the type of the illegal access rejected with the PBGCx. The register is not updated when corresponding ERRSLVCxSTAT.ERR is 1.

Access: ERRSLVCxTYPE is read-only register that can be read in 32-bit units.
ERRSLVCxTYPEL is read-only register that can be read in 16-bit units.

Address: ERRSLVCxTYPE: FFC4 C80C$_H$ + (100$_H$ × x)
ERRSLVCxTYPEL: FFC4 C80C$_H$ + (100$_H$ × x)

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40C.72 ERRSLVCxTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 13	PEID[2:0]	PEID of the access source from which the PBGC protection violation is generated.
12 to 10	Reserved	When read, the value after reset is returned.
9, 8	SPID[1:0]	SPID of the access source from which the PBGC protection violation is generated.
7	Reserved	When read, the value after reset is returned.
6	UM	UM of the access source from which the PBGC protection violation is generated.
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	This bit is set to 1 when an access that has generated the PBGC protection violation is the write.

Section 41 Data CRC (DCRA)

This section contains a generic description of the data CRC function A (DCRA).

The first part of this section describes the features specific to RH850/F1KH, RH850/F1KM, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of DCRA.

41.1 Features of RH850/F1KH, RH850/F1KM DCRA

41.1.1 Number of Units

This microcontroller has the following number of DCRA units.

Each DCRA unit has single channel interface.

Table 41.1 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	4	4	4
Name	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)

Table 41.2 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	4	4	4	4	4
Name	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)

Table 41.3 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	1	1	4	4
Name	DCRAn (n = 0)	DCRAn (n = 0)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)

Table 41.4 Index (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Index	Description
n	Throughout this section, the individual data CRC function A units are identified by the index “n”; for example, DCRAnCTL (n = 0 to 3) indicates the DCRAn control register.

41.1.2 Register Base Addresses

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses.

Table 41.5 Register Base Addresses (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Base Address Name	Base Address
<DCRA0_base>	FFF7 0000_H
<DCRA1_base>	FFF7 1000_H
<DCRA2_base>	FFF7 2000_H
<DCRA3_base>	FFF7 3000_H

41.1.3 Clock Supply

The DCRAn clock supply is shown in the following table.

Table 41.6 Clock Supply (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Unit Clock Name	Supply Clock Name	Description
DCRAn	PCLK	CPUCLK_UL	Module clock
	Register access clock	CPUCLK_UL	Bus clock

41.1.4 Reset Sources

DCRAn reset sources are listed in the following table. DCRAn is initialized by these reset sources.

Table 41.7 Reset Sources (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Unit Name	Reset Source
DCRAn	All reset sources (ISORES)

41.2 Overview

41.2.1 Functional Overview

The data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC
  (X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2 +X1+1)
- 16-bit CCITT CRC
  (X16+X12+X5+1)
- CRC of an arbitrary data block length can be generated.
- After initialization of the CRC data register, every write access to the CRC input register generates a new CRC according to the selected polynomial, and the result is stored in the CRC data register.

41.2.2 Block Diagram

The following picture shows the block diagram of the data CRC function A.

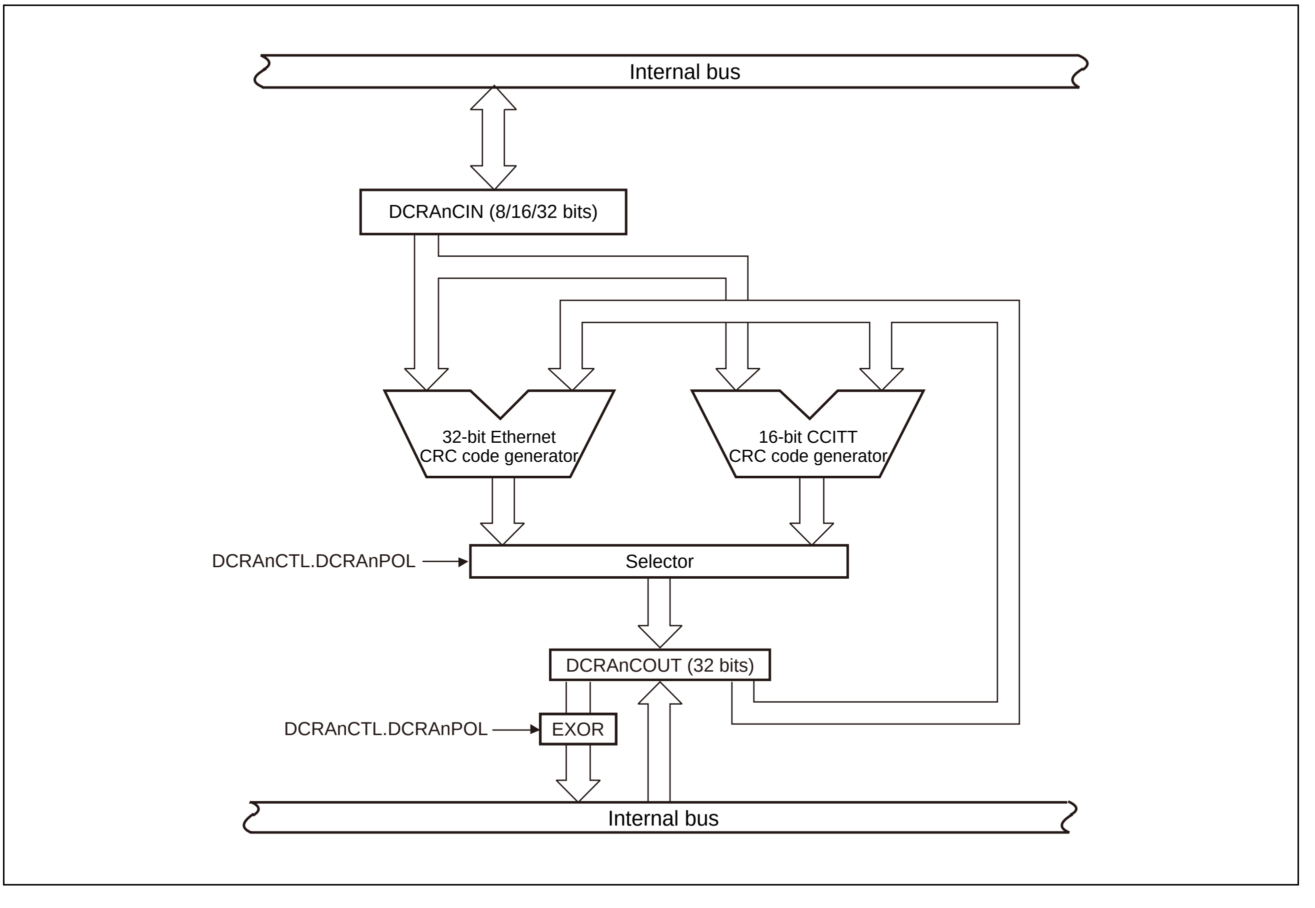
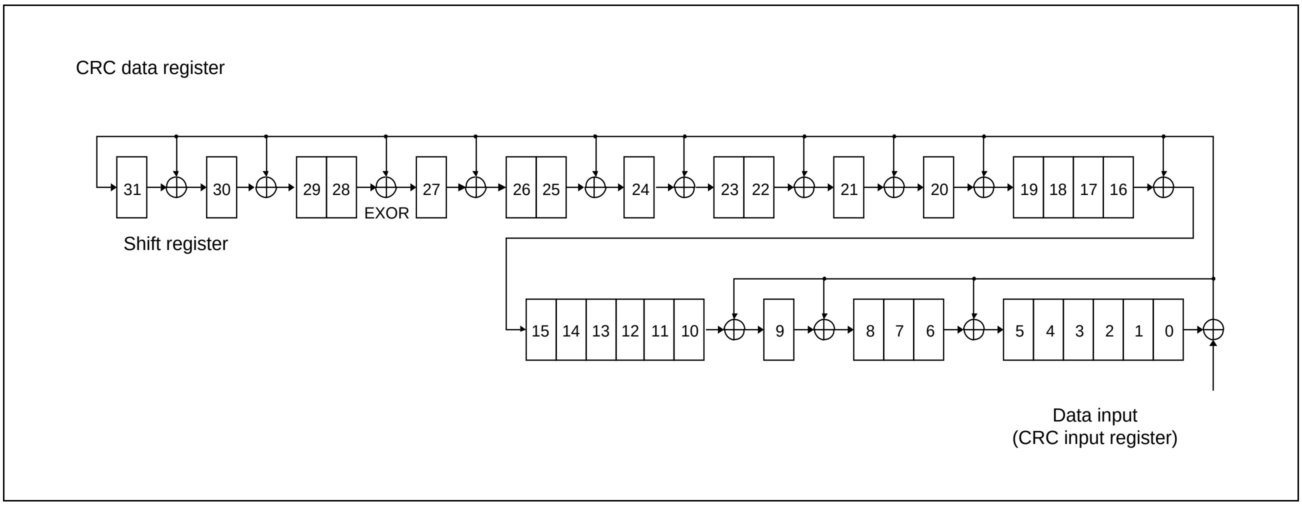


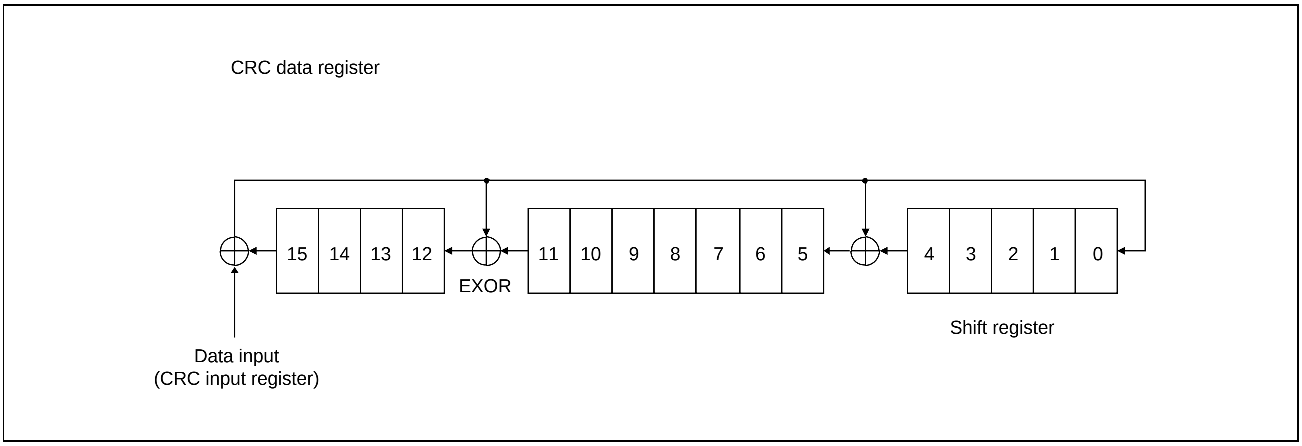
Figure 41.1 Block Diagram of Data CRC Function A

41.2.3 Operational Circuit

- **32-bit Ethernet**



- **16-bit CCITT**



41.3 Registers

41.3.1 List of Registers

DCRA registers are listed in the following table.

For details about <DCRAn_base>, **Section 41.1.2, Register Base Addresses**.

Table 41.8 List of Registers

Module Name	Register Name	Symbol	Address
DCRAn	CRC input register	DCRAnCIN	<DCRAn_base> + 00_H
	CRC data register	DCRAnCOUT	<DCRAn_base> + 04_H
	CRC control register	DCRAnCTL	<DCRAn_base> + 20_H

41.3.2 DCRAnCIN — CRC Input Register

This register holds the input data for CRC calculation. The effective bit width used for CRC calculation must be set by DCRAnCTL.DCRAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAnCIN register is written. The DCRAnCOUT register must be initialized with the initial starting value, before the first data of the data block is written to DCRAnCIN register.

Access: This register can be read or written in 32-bit units

Address: <DCRAn_base>

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCIN[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCIN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.9 DCRAnCIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCIN[31:0]	Input Data for CRC Calculation The valid bits are: ● For 32 bit effective bit width: DCRAnCIN[31:0] ● For 16 bit effective bit width: DCRAnCIN[15:0] ● For 8 bit effective bit width: DCRAnCIN[7:0]

41.3.3 DCRAnCOUT — CRC Data Register

This register stores the result of the CRC code generated by the 32-bit Ethernet polynomial or the 16-bit CCITT polynomial.

Access:	This register can be read or written in 32-bit units.
Address:	<DCRAn_base> + 4_H
Value after reset:	0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCOUT[31:16]															
Value after reset *1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCOUT[15:0]															
Value after reset *1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The read value after reset is 0000 0000$_H$ since the 32-bit Ethernet CRC polynomial is selected as the CRC generating function after reset.

Table 41.10 DCRAnCOUT Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCOUT	Result of the CRC Code Generation When the 16-bit CCITT polynomial is enabled, the bits 15 to 0 show the CRC result. The bits 31 to 16 are undefined. The read value of this register is a value obtained by performing EXOR calculation for the following value: • For 32-bit Ethernet polynomial: FFFF FFFF$_H$ • For 16-bit CCITT polynomial: 0000$_H$ For example, when DCRAnCOUT = 5555 5555$_H$ for the 32-bit Ethernet polynomial, AAAA AAAA$_H$ is read.

CAUTION

This register must be initialized by setting the initial start value before the first data of the data block is written to DCRAnCIN register.

41.3.4 DCRAnCTL — CRC Control Register

This register controls the CRC generation process.

Access: This register can be read or written in 8-bit units.

Address: <DCRAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	DCRAnISZ[1:0]		DCRAnPOL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 41.11 DCRAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2, 1	DCRAnISZ[1:0]	Specify the CRC input bit width. 00: 32 bits (DCRAnCIN[31:0]) 01: 16 bits (DCRAnCIN[15:0]) 10: 8 bits (DCRAnCIN[7:0]) 11: Setting prohibited
0	DCRAnPOL	Specifies the CRC generation method. 0: 32-bit Ethernet CRC polynomial generation. The byte order of the DCRAnCIN register is LSB (least significant bit) first. This means that, if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 0 (LSB) is the start bit of the input data. 1: 16-bit CCITT CRC polynomial generation. The byte order of the DCRAnCIN register is MSB (most significant bit) first. This means that, if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 7 (MSB) is the start bit of the input data.

CAUTION

- If the CRC generation method (DCRAnCTL.DCRAnPOL) is changed, the DCRAnCOUT register must be initialized by setting the initial start value.
- The CRC input bit width (DCRAnCTL.DCRAnISZ[1:0]) must be set according to the data block bit width. Changing the CRC input bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or one word). After the final CRC result is read from DCRAnCOUT register, the bit width can be changed. In that case, the DCRAnCOUT register must be initialized with the initial start value.

41.4 Operation

The data CRC function A generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the data CRC function in 8-, 16- or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT. The initial starting value must be set at the DCRAnCOUT register before the first write access to the CRC input register (DCRAnCIN) is performed.

The flowchart below shows the CRC generating procedure.

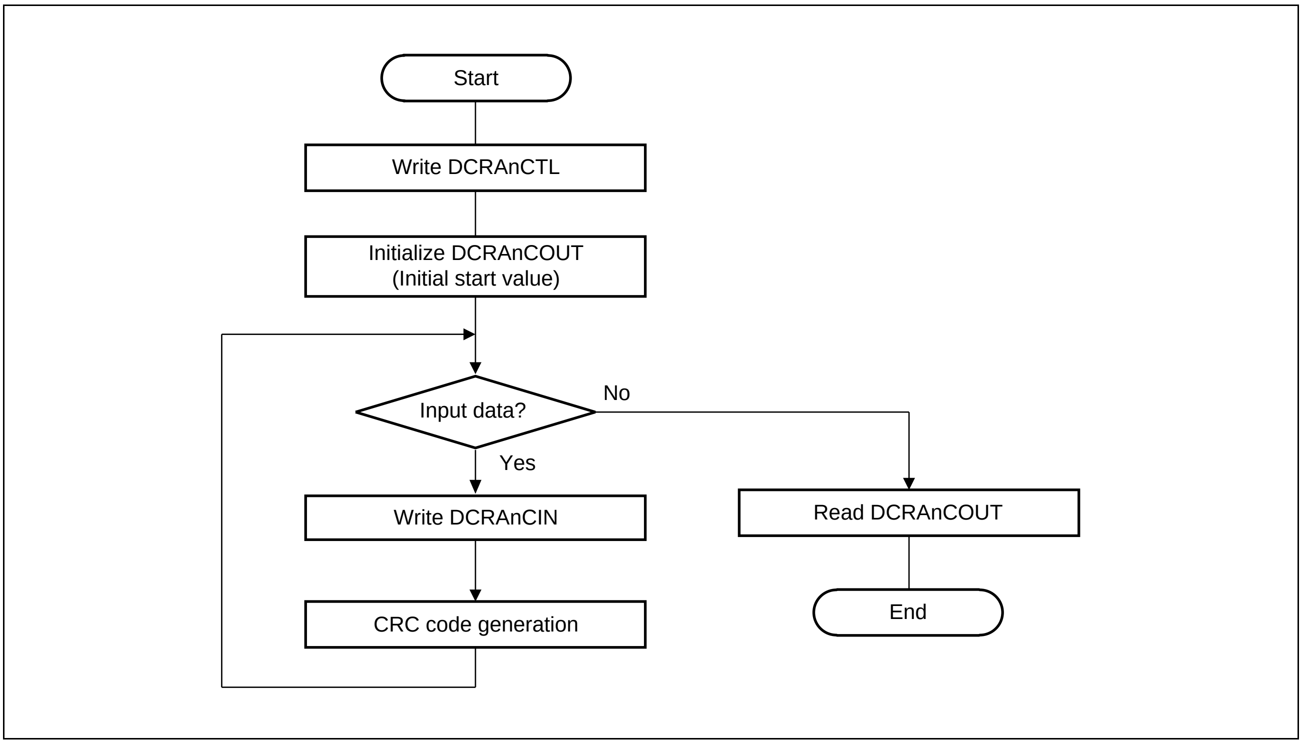


Figure 41.2 Flowchart of Data CRC Function A

NOTES

1. Before writing the first data to DCRAnCIN, the CRC output register DCRAnCOUT must be initialized with the initial start value.
2. DCRAnCOUT must be re-initialized by setting the initial start value when the polynomial is changed by changing DCRAnCTL.DCRAnPOL.
3. Setting example of the initial start values of the respective polynomials
The following is the example of setting values.

Table 41.12 Setting Example of Initial Start Values (When Read at a Reset)

	Initial Start Value	EXOR Value	DCRAnCOUT Read Value
16-bit CCITT	XXXX $FFFF_H$	XXXX 0000_H	XXXX $FFFF_H$
32-bit Ethernet	FFFF $FFFF_H$	FFFF $FFFF_H$	0000 0000_H

Note: X: Undefined

Section 42 Security Function

Please refer to separate documents about security function which describes the two functions as shown below.

- ICUMD for RH850/F1KH-D8, RH850/F1KM-S4
- ICUSE for RH850/F1KM-S1

Section 43 On-Chip Debug Unit (OCD)

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001™-2003 Class 3*[1], a Nexus debug interface standard.

Note 1. This function is supported only by products with an $\overline{\text{EVTO}}$ pin.

CAUTION

The debug functions described in this section are supported by the microcontroller but whether they are usable depends on the debugger. For details on debugging, see the user's manual of the debugger.

43.1 Overview of RH850/F1KH, RH850/F1KM OCD

43.1.1 Functional Overview

The on-chip debug functions described below are supported by the microcontroller.

(1) Debug Interface

This microcontroller supports the following as debug interfaces: Nexus Interface, Low Pin Debug Interface (1 pin) - hereinafter called "LPD (1 pin)", and Low Pin Debug Interface (4 pins) - hereinafter called "LPD (4 pins)".

On-chip debug can be performed using these debug interfaces.

NOTE

When LPD (1 pin) is used, LPD is operating by the clock of MainOSC.

(2) Debug Monitoring Function

Debug-dedicated monitor program space is mounted and is used during debugging.

The basic debug functions below can be used by running a monitoring program.

- Downloading the user-created program
- Reading and writing the memory and registers
- Running the user-created program starting at any address

(3) On-chip Break

A maximum of 12 breakpoints can be specified at any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified for any access (access address, access data).

(4) Software Break

Software break points can be specified at any execution address.

(5) Peripheral Break

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

(6) Forced Break

Execution of the user-created program can be interrupted forcibly.

(7) Forced Reset

This device (microcontroller) can be forcibly reset.

(8) Real Time RAM Monitoring (RRM)

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

(9) Dynamic Memory Modification (DMM)

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

(10) Timer Function

Using a 32-bit counter, the time for running the user-created program can be measured based on the clock for debug.

For the measurement accuracy, see the user’s manual of the debugger.

(11) Mask Function

Masking the following factors is possible.

- All reset sources except for a POC reset and a wakeup reset
- $\overline{\text{MEMC0WAIT}}$ *[1]

Note 1. This function is supported only by products with MEMC.

(12) Hot Plug-in Function

Debugging can be started in normal operating mode without the external reset input.

NOTE

When the hot plug-in function is used in power save mode, the INTDCUTDI interrupt is required to return from power save mode as the wake-up process. About the INTDCUTDI interrupt, see **Table 7A.4, EI Level Maskable Interrupt Sources**, **Table 7BC.5, EI Level Maskable Interrupt Sources (RH850/F1KM-S4)** and **Table 7BC.6, EI Level Maskable Interrupt Sources (RH850/F1KM-S1)** (Channel No.21).

(13) Security Function

To prevent the contents of the flash memory from being read by an unauthorized person, a 128-bit ID code can be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code written to the microcontroller, the flash memory cannot be accessed.

For details on how to set the ID code, see the user’s manual of the debugger.

(14) Trace Function

Execution history, data changes, etc. of the user-created program can be obtained.

NOTE

The trace function is only available in devices with 8-MB memory in RH850/F1KH-D8.

The trace function is only available in devices with 4-MB memory in RH850/F1KM-S4.

The trace function is only available in devices with 1-MB memory in RH850/F1KM-S1.

43.1.2 External Input/Output Pins

Table 43.1 External Input/Output Pins (RH850/F1KH-D8)

Pin	Special Function	PKG No			
		176 Pins	233 Pins	272 Pins	324 Pins
JP0_0	DCUTDI/LPDI/LPDIO	✓	✓	—	✓
JP0_1	DCUTDO/LPDO	✓	✓	—	✓
JP0_2	DCUTCK/LPDCLK	✓	✓	—	✓
JP0_3	DCUTMS	✓	✓	—	✓
JP0_4	$\overline{\text{DCUTRST}}$	✓	✓	—	✓
JP0_5	$\overline{\text{DCURDY}}$ /LPDCLKOUT	✓	✓	—	✓
JP0_6	$\overline{\text{EVTO}}$	✓	✓	—	✓

Table 43.2 External Input/Output Pins (RH850/F1KM-S4)

Pin	Special Function	PKG No				
		100 Pins	144 Pins	176 Pins	233 Pins	272 Pins
JP0_0	DCUTDI/LPDI/LPDIO	✓	✓	✓	✓	✓
JP0_1	DCUTDO/LPDO	✓	✓	✓	✓	✓
JP0_2	DCUTCK/LPDCLK	✓	✓	✓	✓	✓
JP0_3	DCUTMS	✓	✓	✓	✓	✓
JP0_4	$\overline{\text{DCUTRST}}$	✓	✓	✓	✓	✓
JP0_5	$\overline{\text{DCURDY}}$ /LPDCLKOUT	✓	✓	✓	✓	✓
JP0_6	$\overline{\text{EVTO}}$	—	✓	✓	✓	✓

Table 43.3 External Input/Output Pins (RH850/F1KM-S1)

Pin	Special Function	PKG No			
		48 Pins	64 Pins	80 Pins	100 Pins
JP0_0	DCUTDI/LPDI/LPDIO	✓	✓	✓	✓
JP0_1	DCUTDO/LPDO	✓	✓	✓	✓
JP0_2	DCUTCK/LPDCLK	✓	✓	✓	✓
JP0_3	DCUTMS	✓	✓	✓	✓
JP0_4	$\overline{\text{DCUTRST}}$	✓	✓	✓	✓
JP0_5	$\overline{\text{DCURDY}}$ /LPDCLKOUT	✓	✓	✓	✓

43.2 Peripheral Break Control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

During peripheral break, the peripheral modules operate as follows.

a. Modules that stop unconditionally regardless of the EPC.SVSTOP setting

Table 43.4 Modules that Stop Unconditionally Regardless of the EPC.SVSTOP Setting (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Module
Window watchdog timer (WDTA)

CAUTION

RH850/F1KH-D8:
Do not write to the WDTAn when using the individual break function.
In order for the WDTAn of a CPUn for which break points have not been configured to receive clear commands, writing to the WDTAn by using SVACCESS is not disabled.
If WDTA0MD or a trigger register is written on the debugger GUI or by some other means, it might cause an unintentional reset.
To prevent this, be sure to configure synchronous breaks for all CPUs. (SVACCESS is enabled by a tool.)

b. Modules that continue to operate by the setting of emulation registers even when EPC.SVSTOP = 1

Table 43.5 Modules that Continue to Operate by the Setting of Emulation Registers even when EPC.SVSTOP = 1 (RH850/F1KH-D8)

Module	Emulation Register	n
OS timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: Stops during break 1: Continues during break	0 to 9
Timer array unit D (TAUD)	TAUDnEMU.TAUDnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Timer array unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Real-time clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0 to 4
Clocked serial interface H (CSIH)	CSIHnEMU.CSIHnSVSDIS 0: Stops during break 1: Continues during break	0 to 4
Timer motor control function (TAPA)	TAPAnEMU.TAPAnSVSDIS 0: Stops during break 1: Continues during break	0
Encoder timer (ENCA)	ENCAnEMU.ENCAnSVSDIS 0: Stops during break 1: Continues during break	0
PWM output/diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: Stops during break 1: Continues during break	0
	PWSAnEMU.PWSAnSVSDIS 0: Stops during break 1: Continues during break	0
A/D converter (ADCA)	ADCAnEMU.ADCAnSVSDIS 0: Stops during break 1: Continues during break	0, 1

Table 43.6 Modules that Continue to Operate by the Setting of Emulation Registers even when EPC.SVSTOP = 1 (RH850/F1KM-S4)

Module	Emulation Register	n
OS timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: Stops during break 1: Continues during break	0 to 4
Timer array unit D (TAUD)	TAUDnEMU.TAUDnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Timer array unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Real-time clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Clocked serial interface H (CSIH)	CSIHnEMU.CSIHnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Timer motor control function (TAPA)	TAPAnEMU.TAPAnSVSDIS 0: Stops during break 1: Continues during break	0
Encoder timer (ENCA)	ENCAnEMU.ENCAnSVSDIS 0: Stops during break 1: Continues during break	0
PWM output/diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: Stops during break 1: Continues during break	0
	PWSAnEMU.PWSAnSVSDIS 0: Stops during break 1: Continues during break	0
A/D converter (ADCA)	ADCAnEMU.ADCAnSVSDIS 0: Stops during break 1: Continues during break	0, 1

Table 43.7 Modules that Continue to Operate by the Setting of Emulation Registers even when EPC.SVSTOP = 1 (RH850/F1KM-S1)

Module	Emulation Register	n
OS timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit D (TAUD)	TAUDnEMU.TAUDnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Real-time clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface H (CSIH)	CSIHnEMU.CSIHnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Timer motor control function (TAPA)	TAPAnEMU.TAPAnSVSDIS 0: Stops during break 1: Continues during break	0
Encoder timer (ENCA)	ENCAnEMU.ENCAnSVSDIS 0: Stops during break 1: Continues during break	0
PWM output/diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: Stops during break 1: Continues during break	0
	PWSAnEMU.PWSAnSVSDIS 0: Stops during break 1: Continues during break	0
A/D converter (ADCA)	ADCAnEMU.ADCAnSVSDIS 0: Stops during break 1: Continues during break	0

CAUTION

For details on the registers, see the register description of the corresponding section.

c. Modules that stop when EPC.SVSTOP = 1

Table 43.8 Modules that Stop when EPC.SVSTOP = 1 (RH850/F1KH-D8, RH850/F1KM-S4, RH850/F1KM-S1)

Module
LIN/UART interface (RLIN3)
Low-power sampler (LPS)

43.3 Hot Plug-in in Each Mode

43.3.1 RUN Mode

When Hot Plug-in has occurred in RUN mode, it is necessary to maintain the current state.

43.3.2 STOP/DeepSTOP Mode

When Hot Plug-in has occurred in STOP/DeepSTOP mode, it is necessary to transfer in RUN mode.

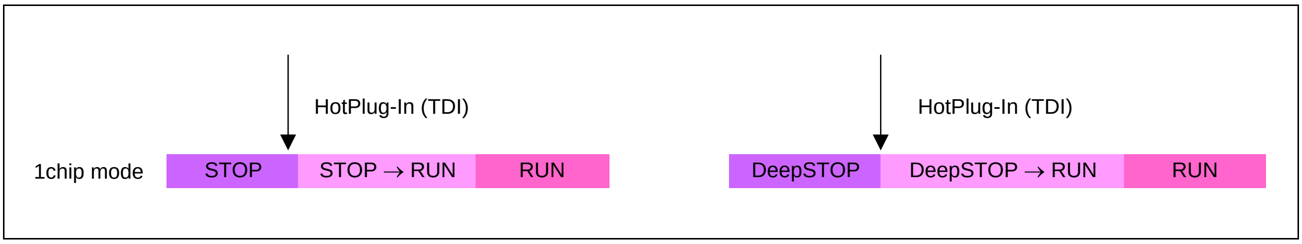


Figure 43.1 Hot Plug-in in STOP/DeepSTOP Mode

43.3.3 Cyclic RUN Mode

When Hot Plug-in has occurred in Cyclic RUN mode, it is necessary to transfer in DeepSTOP mode and wakeup to RUN mode.

All regulators in ISO are made operating state in OCD mode.

Then, on the occurrence of interrupt of “DCUTDI Low level Detection interrupt (INTDCUTDI)”, it is necessary for a transfer order to be executed to DeepSTOP mode by user (software) if on-chip debugging is performed.

The sequence of the time when to enter to OCD mode in Cyclic RUN mode is mentioned below.

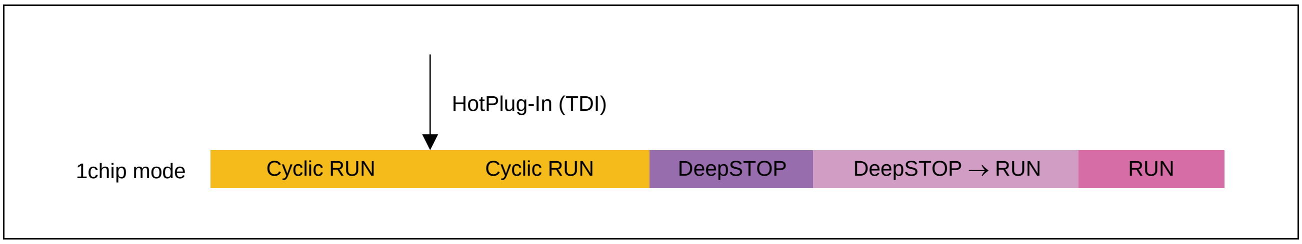


Figure 43.2 Hot Plug-in in Cyclic RUN Mode

43.3.4 Cyclic STOP Mode

When Hot Plug-in has occurred in Cyclic STOP mode, it is necessary to transfer in Cyclic RUN mode.

Then, on the occurrence of wakeup factor of DCUTDI, it is necessary for a transfer to be executed as the same case of Cyclic RUN mode.
The sequence of the time when to enter to OCD mode in Cyclic STOP mode is mentioned below.

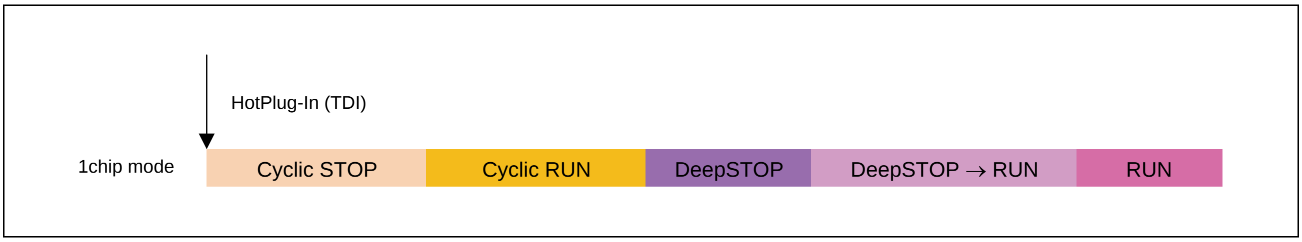


Figure 43.3 Hot Plug-in in Cyclic STOP Mode

43.4 Registers

43.4.1 List of Registers

OCD register is listed in the following table.

Table 43.9 List of Register

Module Name	Register Name	Symbol	Address
—	Emulation peripheral control register	EPC	—

43.4.2 EPC — Emulation Peripheral Control Register

This register stops operation of peripheral functions (timer, serial interface, and A/D converter) in debug mode (SVSTOP).

Access: Accessing from the user program is prohibited.

Address: —

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SVSTOP	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—

Table 43.10 EPC Register Contents

Bit Position	Bit Name	Function
7	Reserved	—
6	SVSTOP	Stops operation of peripheral functions (timer, serial interface, and A/D converter) during debugging. 0: Does not stop operation 1: Stops operation
5 to 0	Reserved	—

NOTE

EPC is set by the debugger. Setting by the user program is prohibited. As for the setting of the debugger, see the user's manual of the debugger.

43.5 Cautions on Using On-Chip Debugging

43.5.1 Treatment of Devices Used for Debugging

Do not install a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during system debugging and thus the write/erase endurance of the flash memory cannot be guaranteed.

43.5.2 Reset Assertion when a Debugger is Connected

If a program in which a reset is asserted at the start of program execution is executed when a debugger is being used, the microcontroller is reset before preparation for communications between the OCD emulator and microcontroller is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcomputer. To ensure that the debugger operates properly when debugging a program in which a reset is asserted at the start of program execution, insert a wait between the start of program and reset assertion.

43.5.3 Restrictions when HS IntOSC is Used as the Main Clock Source Instead of MainOSC

Need to change the debug interface configuration (OPJTAG) to LPD (4 pins) in advance if E1 Emulator is used for debug.

LPD (1 pin) mode cannot be used.

43.5.4 Restrictions when the Writing of OCD_MD and Reset are Occur at the Same Time, or Restrictions when the Writing of MTR (DBG_CTRLP) and Reset are Occur at the Same Time

When the writing of OCD_MD and reset are occur at the same time, or the writing of MTR (DBG_CTRLP) and reset are occur at the same time, it is a possibility of terminal hazard.

If the debugger was disconnected, please reconnect the debugger.

43.5.5 Transition to DeepSTOP Mode when a Debugger is Connected

When a debugger is in use, when a program to transition to DeepSTOP mode is executed immediately following the start of the program, the microcomputer stops power supply to the Isolated area (ISO area) before preparation for communications between the OCD emulator and microcomputer is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcomputer. A wait must be inserted during the period from release from the reset state and transition to DeepSTOP mode by the emulator for correct operation of the debugger when debugging a program to transition to DeepSTOP mode immediately after the program starts.

Section 44 Flash Memory

This section describes the flash memory mounted on RH850/F1KH, RH850/F1KM.

The first part in this section describes the characteristics of the mounted flash memory and the characteristics specific to RH850/F1KH, RH850/F1KM, such as the memory map, flash memory programming, and ECC.

44.1 Features of RH850/F1KH, RH850/F1KM Flash Memory

- Includes code flash memory and data flash memory

  The code flash memory can store program codes and data and has the user area and the extended user area.

  The data flash memory is used for storing data.

- Method of flash memory programming

  Flash memory programming via a serial interface and programming of flash memory by a user program (self-programming) are supported.

- Support for BGO (Back Ground Operation)

  RH850/F1KH-D8, RH850/F1KM-S4:

  - The BGO function allows programs to be executed in the code flash memory while the data flash memory is being programmed/erased.
  - The BGO function allows programs to be executed in the code flash memory while the code flash memory is being programmed/erased. (There are restrictions of programming/erasing area and reading area combination. See **Table 44.6, Conditions under which Background Operation is Usable (RH850/F1KH-D8)** and **Table 44.7, Conditions under which Background Operation is Usable (RH850/F1KM-S4)**).

  RH850/F1KM-S1:

  - The BGO function allows programs to be executed in the code flash memory while the data flash memory is being programmed/erased.

- Flash memory data security
  - Support for security functions to protect against illicit tampering with or reading out of data in the flash memory
  - Support for protection functions to protect against erroneous overwriting of the flash memory

- Option byte function

  RH850/F1KH-D8, RH850/F1KM-S4:

  - Sets the operation after releasing reset for ports, WDTA, CVM, clock divider mode for CPUCLK, FlexRay, Ethernet AVB and $\overline{\text{RESETOUT}}$.

  RH850/F1KM-S1:

  - Sets the operation after releasing reset for ports, WDTA, CVM and $\overline{\text{RESETOUT}}$.

- Support for the error detection/correction function (ECC) in the code flash memory and data flash memory

  Built-in ECC function can detect 2-bit errors and detect/correct 1-bit errors.

- Interrupts can be acknowledged in self-programming mode.

For code flash sizes and data flash sizes of each product, see the following sections.

- **Section 4A.1, Address Space**
- **Section 4B.1, Address Space**
- **Section 4C.1, Address Space**

44.2 Structure of Memory

44.2.1 Mapping of Code Flash Memory

Figure 44.1, Mapping of the Code Flash Memory (RH850/F1KH-D8), **Figure 44.2, Mapping of the Code Flash Memory (RH850/F1KM-S4)**, **Figure 44.3, Mapping of the Code Flash Memory (RH850/F1KM-S1)**, illustrates the mapping of the code flash memory.

The user area of the code flash memory is divided into 8 and 32 KB blocks, which serve as the units of erasure. A single 32 KB block is also incorporated as the extended user area. The user area and extended user area are available as areas for storing the user program.

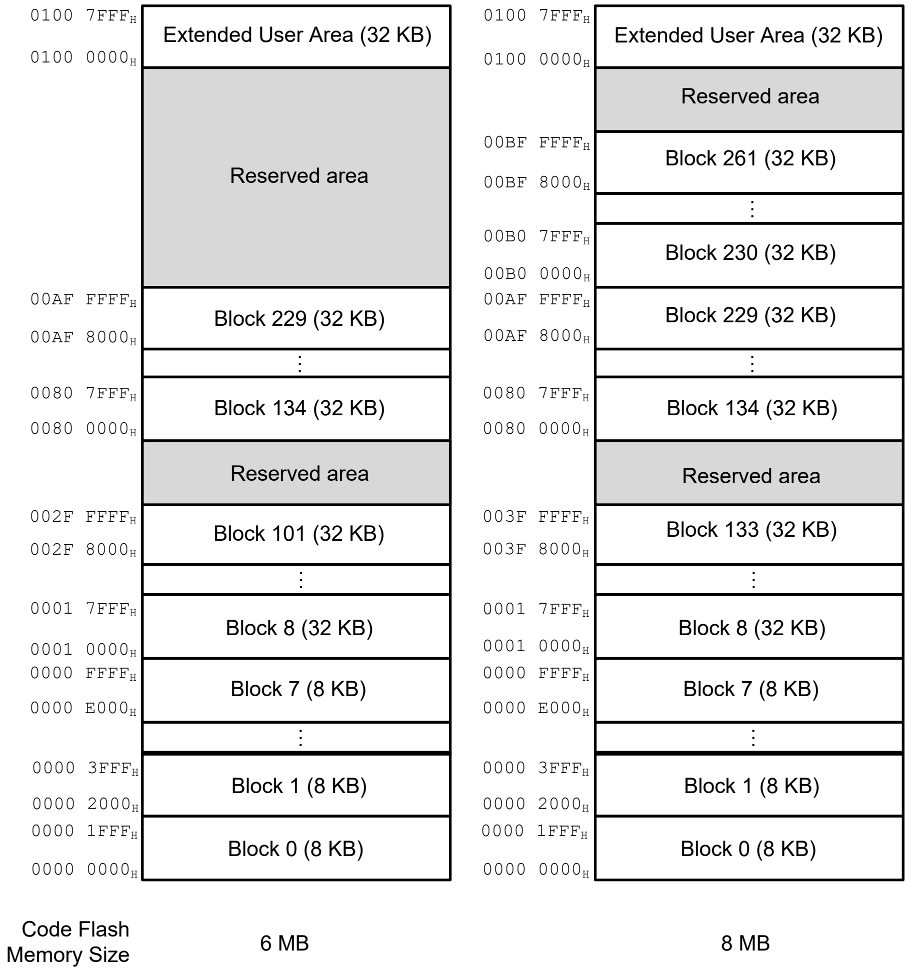


Figure 44.1 Mapping of the Code Flash Memory (RH850/F1KH-D8)

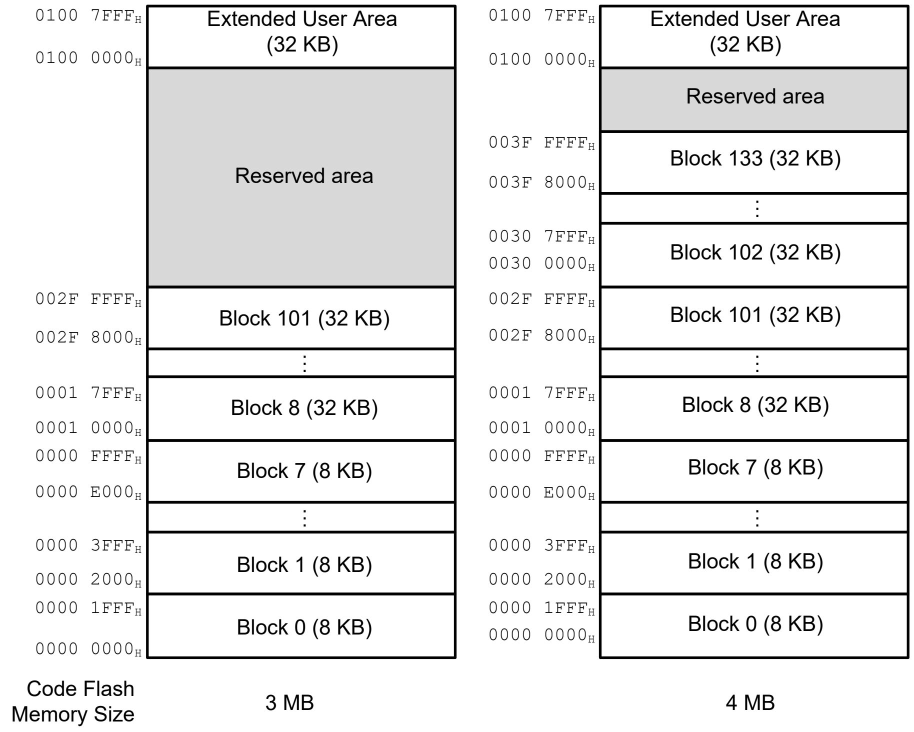


Figure 44.2 Mapping of the Code Flash Memory (RH850/F1KM-S4)

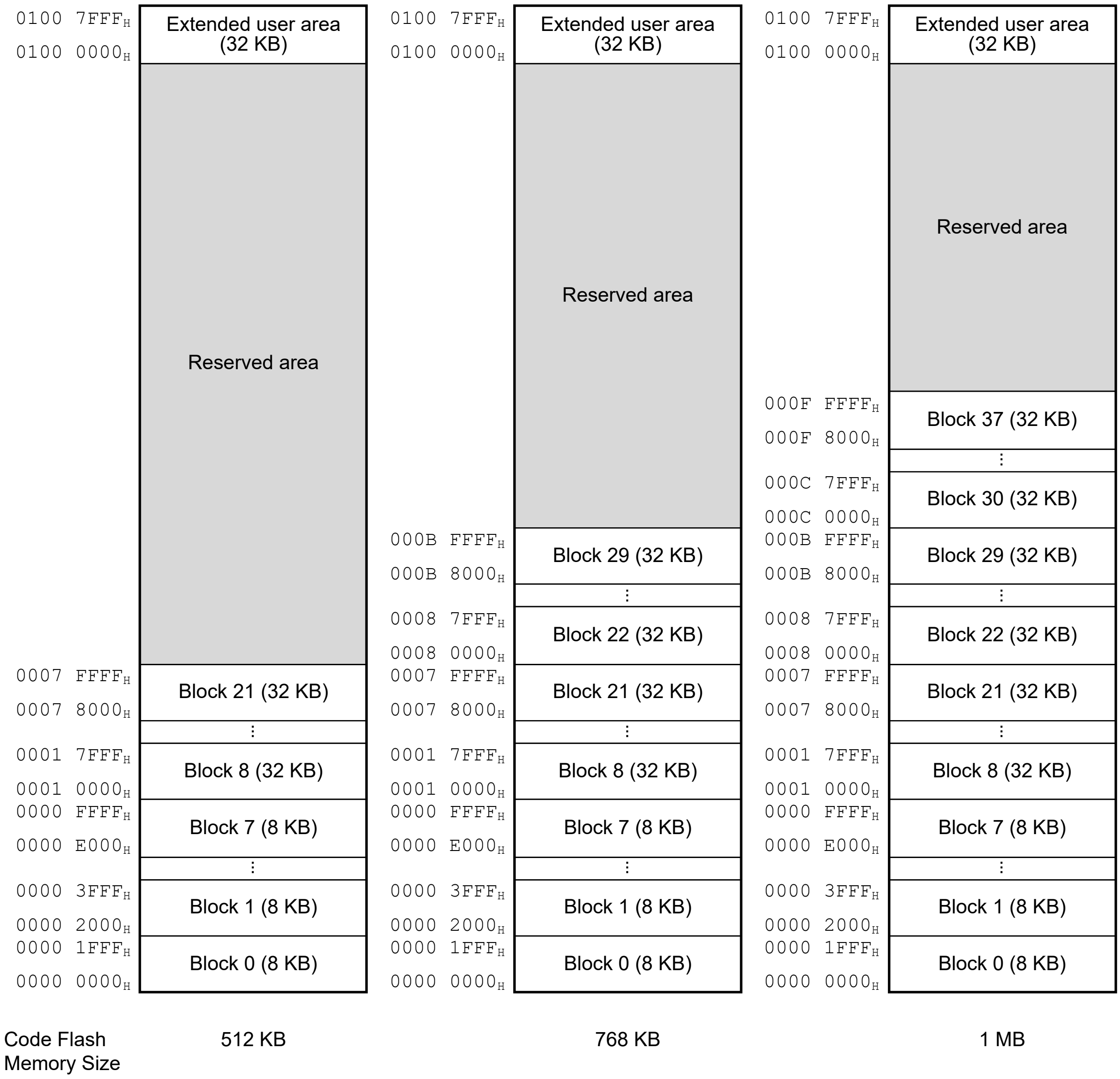


Figure 44.3 Mapping of the Code Flash Memory (RH850/F1KM-S1)

44.2.2 Mapping of Data Flash Memory

The data area of the data flash memory is divided into 64-byte blocks, with each being a unit for erasure. **Figure 44.4, Mapping of the Data Flash Memory (RH850/F1KH-D8)**, **Figure 44.5, Mapping of the Data Flash Memory (RH850/F1KM-S4)**, **Figure 44.6, Mapping of the Data Flash Memory (RH850/F1KM-S1)** shows the mapping of the data flash memory.

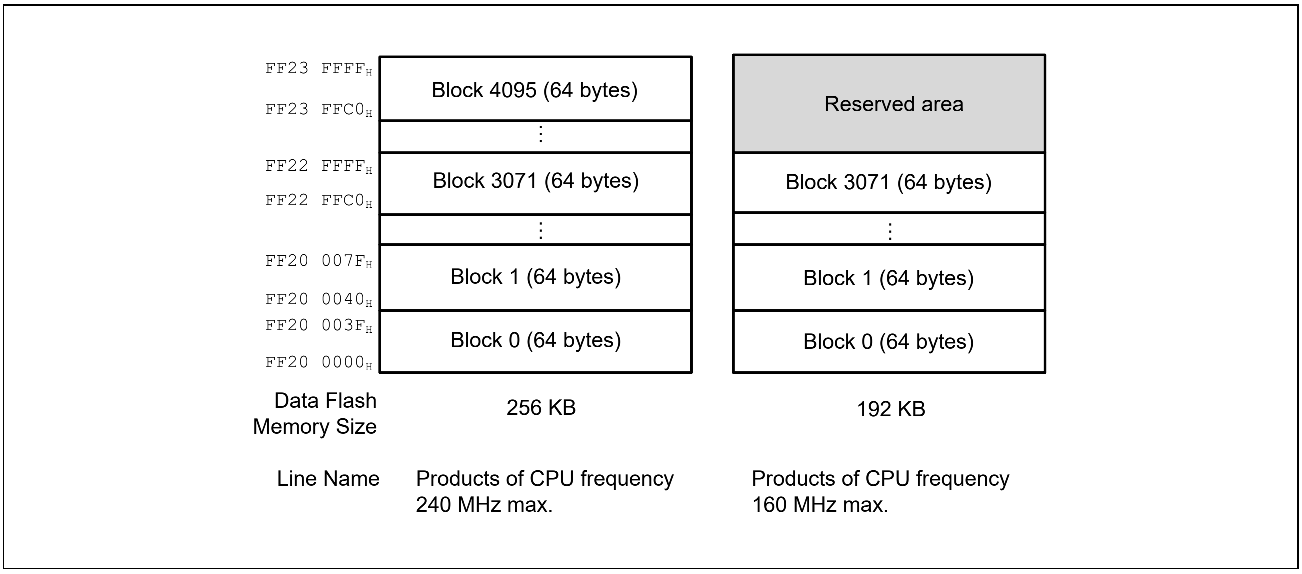


Figure 44.4 Mapping of the Data Flash Memory (RH850/F1KH-D8)

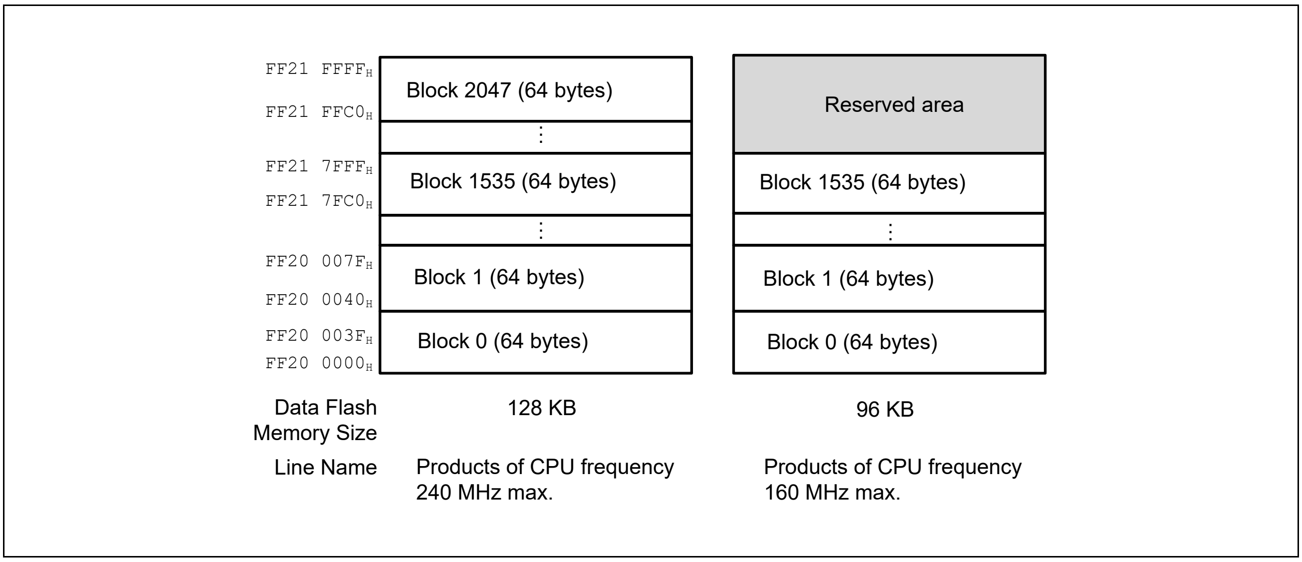


Figure 44.5 Mapping of the Data Flash Memory (RH850/F1KM-S4)

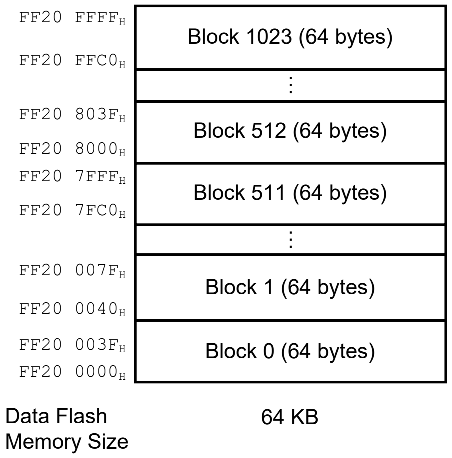


Figure 44.6 Mapping of the Data Flash Memory (RH850/F1KM-S1)

44.3 Operating Modes Associated with Flash Memory

Figure 44.7, Mode Transition Associated with Flash Memory is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 6, Operating Mode**.

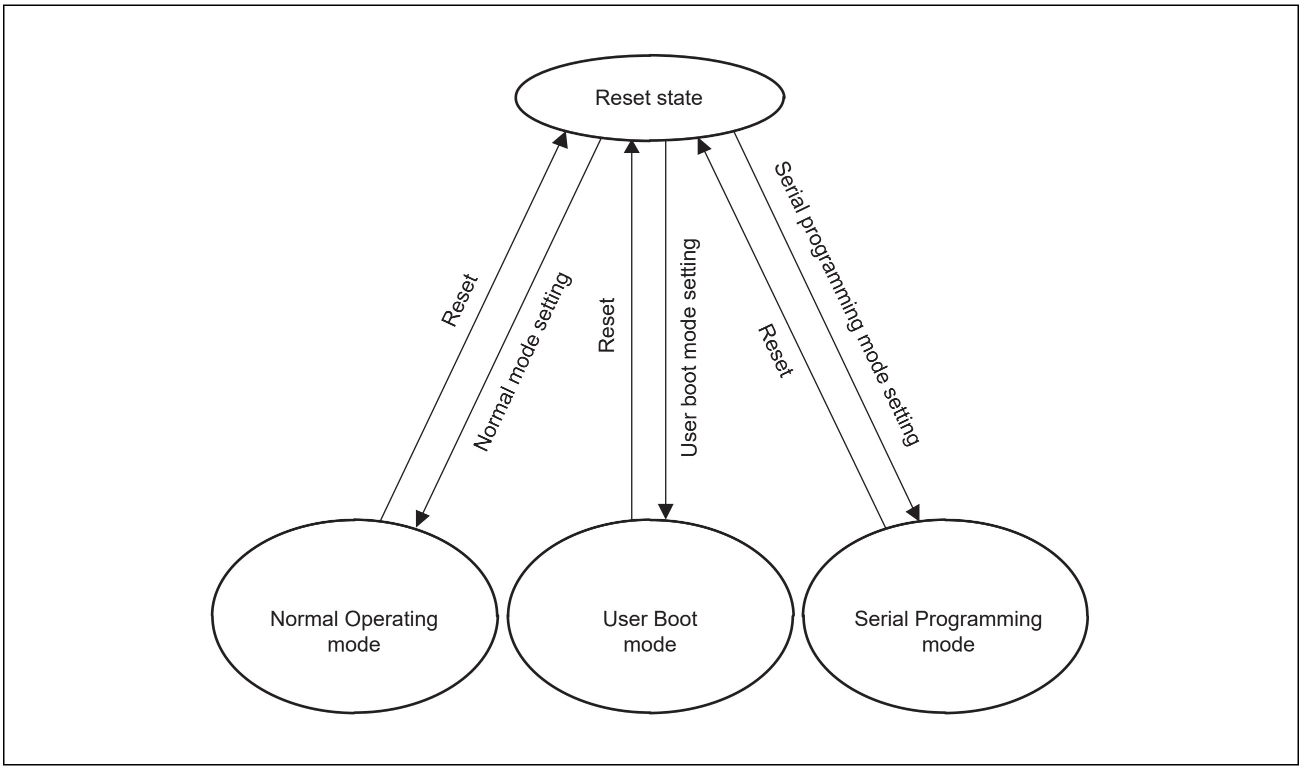


Figure 44.7 Mode Transition Associated with Flash Memory

Table 44.1, Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 44.1 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release

Item	Normal Operating Mode	User Boot Mode	Serial Programming Mode
Programmable and erasable area	• User area • Extended user area • Data area	• User area • Data area	• User area • Extended user area • Data area
Boot program after reset release	Program in user area or extended user area (Changeable by using the variable reset vector)	Program in extended user area. (Reset vector is 0100 0000_H)	Firmware program for serial programming

44.4 Functions

44.4.1 Functional Overview

The flash memory can be updated via a serial interface by a dedicated flash memory programmer (serial programming), before being mounted on the target system or on a flash adapter system.

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as external communications, etc., and this allows programming under various conditions. **Table 44.2, Methods of Programming** gives an overview of the methods of programming and the corresponding operating modes.

Table 44.2 Methods of Programming

Method of Programming	Description	Operating Mode
Serial programming	A dedicated flash memory programmer allows on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	The user program that is written to code flash memory in advance by serial programming also allows updating the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data in code flash memory while the data flash memory is self-programming. For this reason, it is possible to update the data flash memory by executing a program written to the code flash memory. RH850/F1KH-D8, RH850/F1KM-S4: In addition, when the specific conditions of the code flash memory range for updating and the code flash memory range for reading are satisfied, the background operation can be used. (see **Table 44.6, Conditions under which Background Operation is Usable (RH850/F1KH-D8)**, **Table 44.7, Conditions under which Background Operation is Usable (RH850/F1KM-S4)**). In this case, the code flash memory can be update using the instructions for programming in the code flash memory by self-programming. When background operation cannot be used, instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being updated by self-programming. In such cases, a program for updating must be transferred to the internal RAM in advance and executed. RH850/F1KM-S1: Instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being updated by self-programming. In such cases, a program for updating must be transferred to the local RAM in advance and executed.	Normal operating mode User boot mode

Renesas provides a library for self-programming. For details on this library, see the user's manuals for the code flash library and data flash library of this device.

Table 44.3, Basic Functions at a Glance lists the functions of the flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the flash memory by a library function or the user program enables self-programming.

Table 44.3 Basic Functions at a Glance

Function	Description	Level of Support (✓: Supported, Δ: Conditionally Supported, —: Not Supported)	
		Serial Programming	Self-Programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	✓	Δ (Only data flash is supported)
Block erasure	This is for erasing the contents of a specified block of memory.	✓	✓
Programming	This is for writing to a specified address.	✓	✓
Verification and checksum	Data that are read out from flash memory are compared with data transferred from the flash memory programmer.	✓	— (Reading of data by the user program is possible)
Reading	Data that have been written to the flash memory are read out.	✓	✓
Setting for OTP (one-time programming)	A specified block of code flash memory is set for OTP (OTP can only be set, that is, it is not possible to release a block's OTP setting).	✓	✓
Setting an ID	An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the on-chip debugger, and programming of the code flash memory by self-programming. In the initial state of the product at shipment, the ID codes are FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF$_H$.	✓	✓
Security settings	Security settings are for use in serial programming.	✓	Δ (Only when setting is prohibited after being permitted)
Protection settings	Settings for block protection of code flash memory and variable reset vector are provided.	Δ (Setting of the reset vector values for variable reset vector function is not supported.)	✓
Setting of option bytes	Option bytes are set to change them from the initial values.	✓	✓
Clearing the configuration	ID setting, security settings, protection settings, and option byte settings are initialized. Execution of the configuration clear command is prohibited in the following cases: ● The lock bit is set to one of blocks. ● One block of a code flash and a data flash is not blank. ● Prohibited by security function (**Table 44.4, Summary of Security Functions**)	✓	—

For details on serial programming, see the user's manual of the flash programmer.

For details on self-programming, see the user's manuals for the code flash library and data flash library of this device.

The flash memory supports various security functions.

The OTP setting and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, prohibiting connection of a dedicated flash memory programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.

The security functions supported by the flash memory are listed in **Table 44.4, Summary of Security Functions**.

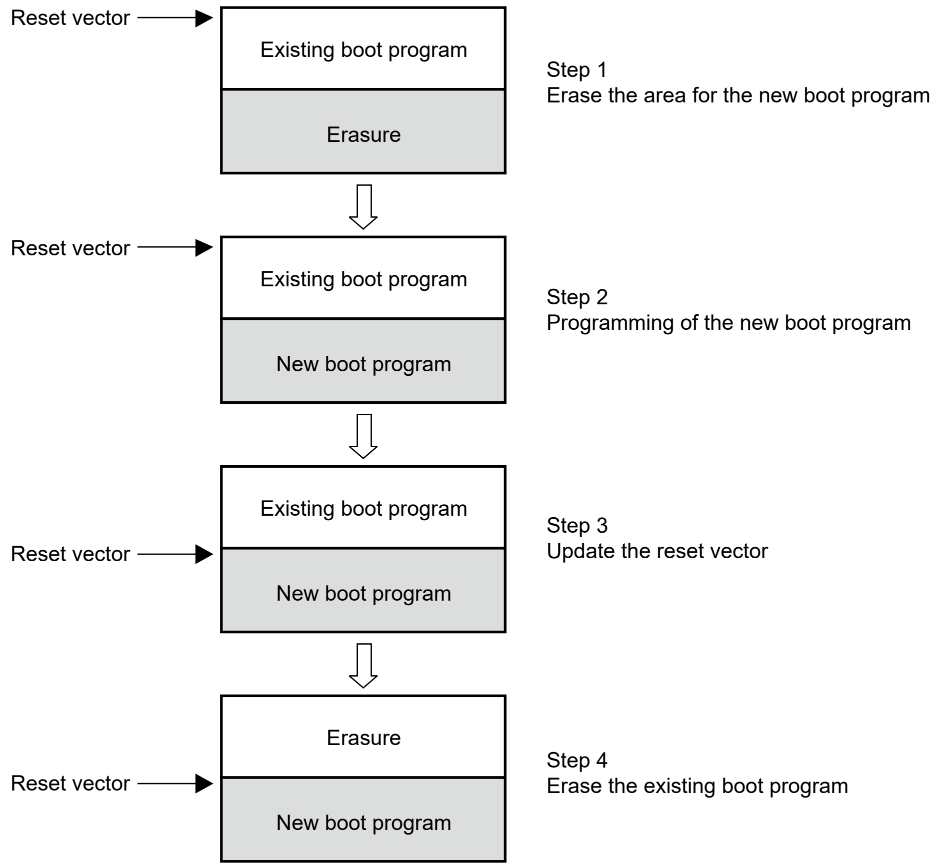
Table 44.4 Summary of Security Functions

Function	Description
OTP	OTP can be individually set for each block of the user area and the extended user area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self-programming is prohibited. Once set, the OTP setting cannot be released. Furthermore, since execution of the configuration clearing command is prohibited for any area for which OTP has been set, changing a security setting from “prohibited” to “permitted” is not possible.
ID authentication	The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of self-programming.
Prohibition of connection of a dedicated flash memory programmer	The connection of a dedicated flash memory programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a dedicated flash memory programmer is prohibited, changing a security setting from “prohibited” to “permitted” is not possible.
Prohibition of block erasure commands	Block erasure commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erasure commands are prohibited, changing a security setting from “prohibited” to “permitted” is not possible.
Prohibition of programming commands	Block erasure commands and programming commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.

The flash memory supports various protection functions. The protection functions supported by the flash memory are listed in **Table 44.5, Summary of Protection Functions**.

Table 44.5 Summary of Protection Functions

Function	Description
Block protection	Lock bit settings can be individually made to enable or disable programming and erasure of each block of the user area and the extended user area of code flash memory. Programming and erasure by self-programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when the lock bit function is disabled after having been enabled. When a block of code flash memory is erased, the lock bit for that block is also erased.
Hardware protection	The level on the FLMD0 pin can be set to prohibit programming and erasure of the code flash memory. – FLMD0 = 0: Programming prohibited – FLMD0 = 1: Programming permitted
Variable reset vector	The protection settings include control of the reset vector. As shown in **Figure 44.8, Utilizing the Variable Reset Vector Function to Update the Boot Program**, after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the user area and extended user area.



Note: After step 4, a reset leads to updating of the reset vectors for which RBASE is changed.

Figure 44.8 Utilizing the Variable Reset Vector Function to Update the Boot Program

44.5 Serial Programming

A dedicated flash memory programmer can be used to handle flash memory in serial programming mode.

Serial Programming

The microcontroller is mounted on the system board at the time of serial programming. Providing a connector to the board enables handling of the microcontroller by the flash memory programmer to proceed.

44.5.1 Environments for Programming

The recommended environments for handling the flash memory of the microcontroller with data are described below.

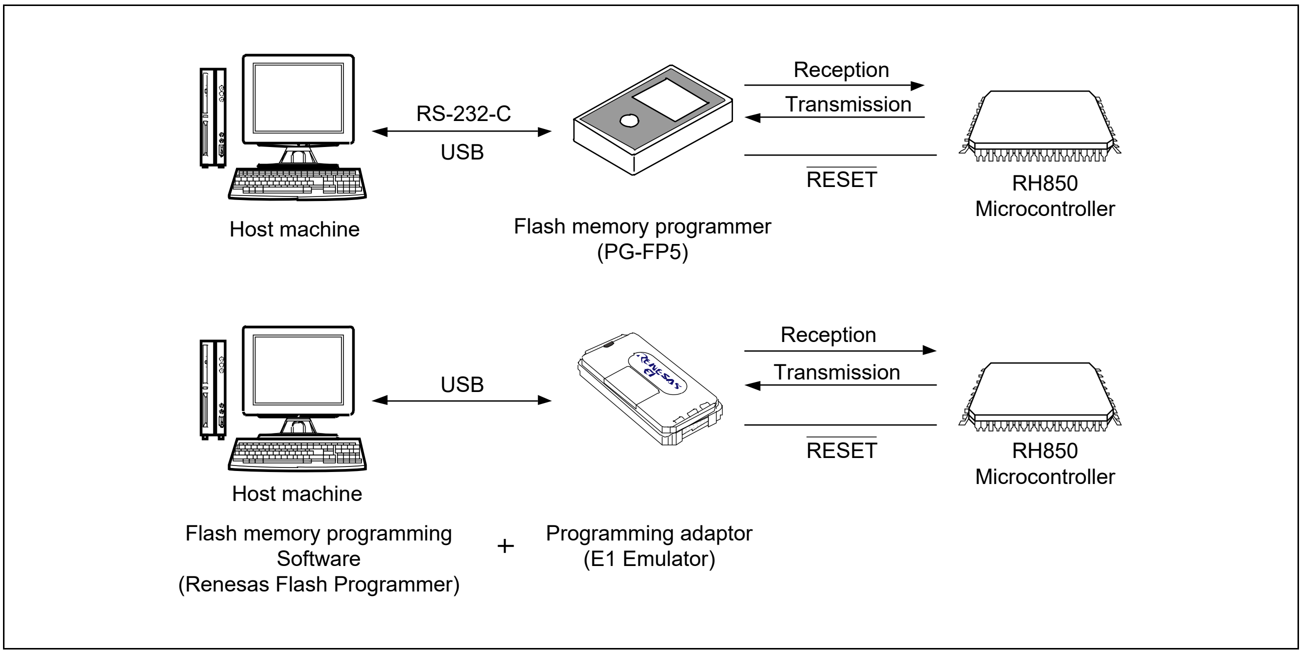


Figure 44.9 Environments for Handling Programs of the Flash Memory

By using the PG-FP5 flash memory programmer or the combination of the Renesas Flash Programmer (software for writing to flash memory) running on the host machine and the E1 emulator as an programming adaptor, the user is easily able to erase, program, and verify the contents of the on-chip memory of flash-memory-equipped microcontrollers from Renesas Electronics.

The PG-FP5 flash memory programmer handles programming from a host machine or programming in stand-alone mode while the Renesas Flash Programmer only handles programming from a host machine.

NOTE

For details on the PG-FP5, see *the PG-FP5 Flash Memory Programmer User's Manual*. For details on the Renesas Flash Programmer of flash programming software, see *the Renesas Flash Programmer Flash Programming Software User's Manual*.

44.6 Communication Modes

44.6.1 Asynchronous Flash Programming Interface — 1-Wire UART

The single-wire asynchronous serial programming interface, 1-wire UART is connected to the flash memory programmer with the following port.

- FPDR (JP0_0): Receive data input/transmit data output

44.6.2 Asynchronous Flash Programming Interface — 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR (JP0_0): Receive data input
- FPDT (JP0_1): Transmit data output

44.6.3 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FPDR (JP0_0): Receive data input
- FPDT (JP0_1): Transmit data output
- FPCK (JP0_2): Serial clock input

The flash memory programmer outputs the serial data clock SCK, and the microcontroller operates as a slave.

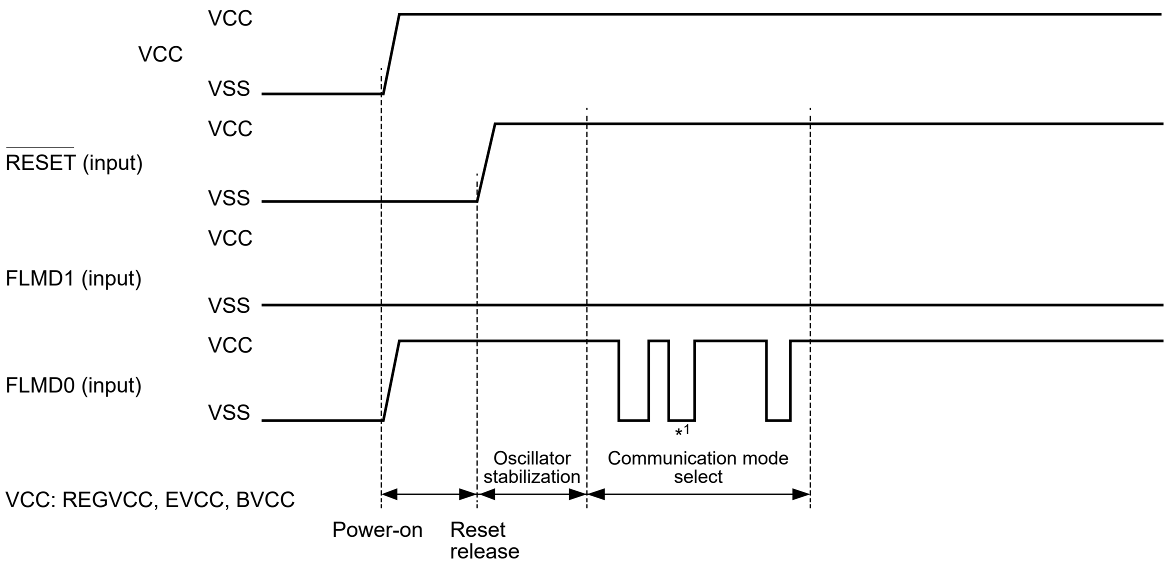
NOTE

For details on Renesas Flash Programmer, see *the Renesas Flash Programmer Flash Programming Software User's Manual*.

44.6.4 Selection of Communication Method

The communication method can be selected by pulse input to the FLMD0 pin (up to 7 pulses) after transition to the flash memory programming mode. The FLMD0 pulse is generated by a dedicated flash memory programmer.

Figure 44.10, Selection of Communication Method shows the relation between the number of pulses and communication method.



Note 1. The number of clock cycles to be inserted depends on the communication method as in the following table.

FLMD0 Pulse	Communication Method
0	2-wire UART
7	1-wire UART
3	CSI
Other than above	Setting prohibited

Figure 44.10 Selection of Communication Method

44.7 Self-Programming

44.7.1 Outline

The RH850/F1KH and RH850/F1KM supports programming of the flash memory by the user program itself. Renesas Electronics provides a code flash library and a data flash library for use with user programs. These libraries can be used for writing to the code flash memory and to the data flash memory.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to internal RAM in advance of the programming operation, and executed from the given destination to perform the programming.

In addition, when the specific conditions of the code flash memory range for updating and the code flash memory range for reading are satisfied, the background operation can be used. (see **Table 44.6, Conditions under which Background Operation is Usable (RH850/F1KH-D8)**, **Table 44.7, Conditions under which Background Operation is Usable (RH850/F1KM-S4)**, **Table 44.8, Conditions under which Background Operation is Usable (RH850/F1KM-S1)**). In this case, the code flash memory can be update using the instructions for programming in the code flash memory by self-programming. The programming program can also be copied to the internal RAM in advance and executed to program the code flash memory.

When background operation cannot be used, the programming program can be copied to the internal RAM in advance and executed to program the code flash memory.

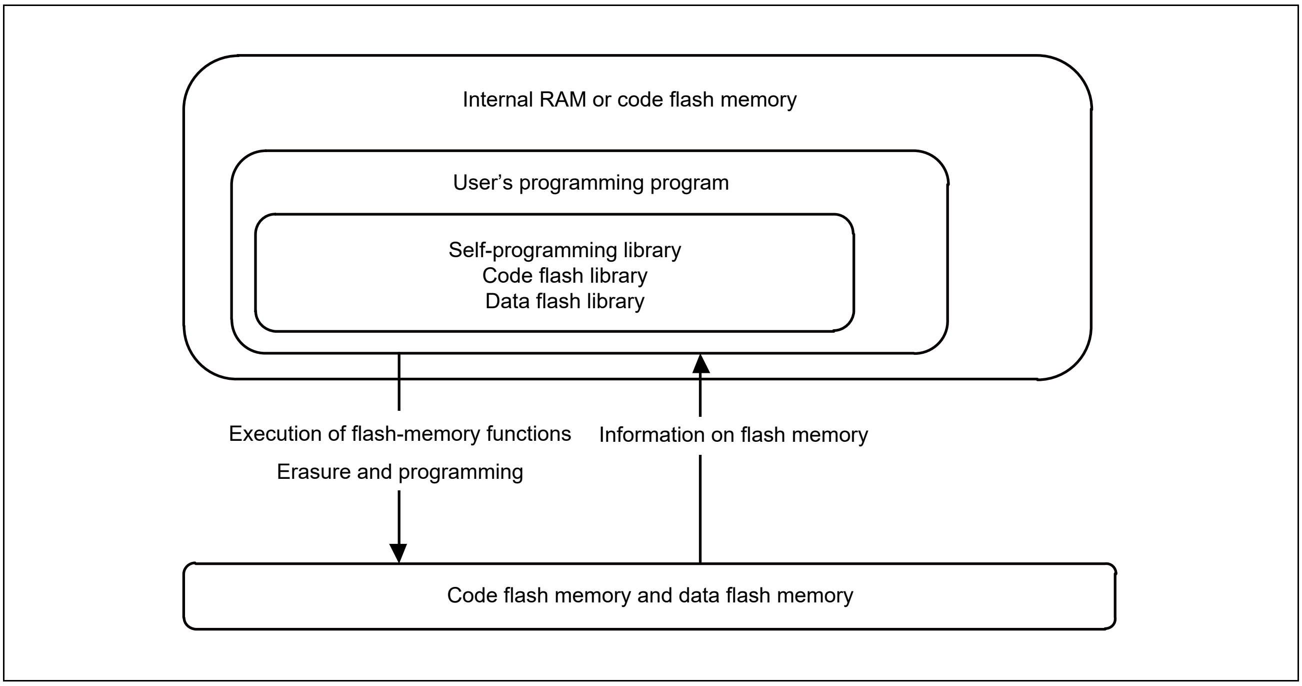


Figure 44.11 Schematic View of Self-Programming

For details on the self-programming of flash memory, see the user's manuals for the code flash and data flash libraries for this device.

44.7.2 Background Operation

Background operations can be used when the combination of the flash memory for writing and the flash memory for reading is any of those listed below.

Table 44.6 Conditions under which Background Operation is Usable (RH850/F1KH-D8)

		Writing					
		[FLI0] Address: 0000 0000$_H$ to 001F FFFF$_H$	[FLI1] Address: 0020 0000$_H$ to 003F FFFF$_H$ (Code Flash 8 MB), 0020 0000$_H$ to 002F FFFF$_H$ (Code Flash 6 MB)	[FLI2] Address: 0080 0000$_H$ to 009F FFFF$_H$	[FLI3] Address: 00A0 0000$_H$ to 00BF FFFF$_H$ (Code Flash 8 MB), 00A0 0000$_H$ to 00AF FFFF$_H$ (Code Flash 6 MB)	Extended User Area*[1]	Data Flash Memory
Reading	[FLI0] Address: 0000 0000$_H$ to 001F FFFF$_H$	—	✓	✓	✓	—	✓
	[FLI1] Address: 0020 0000$_H$ to 003F FFFF$_H$ (Code Flash 8 MB), 0020 0000$_H$ to 002F FFFF$_H$ (Code Flash 6 MB)	✓	—	✓	✓	✓	✓
	[FLI2] Address: 0080 0000$_H$ to 009F FFFF$_H$	✓	✓	—	✓	✓	✓
	[FLI3] Address: 00A0 0000$_H$ to 00BF FFFF$_H$ (Code Flash 8MB), 00A0 0000$_H$ to 00A0 FFFF$_H$ (Code Flash 6MB)	✓	✓	✓	—	✓	✓
	Extended User Area	—	✓	✓	✓	—	✓
	Data Flash Memory	—	—	—	—	—	—

Note: ✓: possible, —: not possible

Note 1. Extended user area cannot be written in User Boot mode.

Table 44.7 Conditions under which Background Operation is Usable (RH850/F1KM-S4)

		Writing			
		[FLI0] Address: 0000 0000$_H$ to 001F FFFF$_H$	[FLI1] Address: 0020 0000$_H$ to 003F FFFF$_H$ (Code Flash 4 MB), 0020 0000$_H$ to 002F FFFF$_H$ (Code Flash 3 MB)	Extended User Area*1	Data Flash Memory
Reading	[FLI0] Address: 0000 0000$_H$ to 001F FFFF$_H$	—	✓	—	✓
	[FLI1] Address: 0020 0000$_H$ to 003F FFFF$_H$ (Code Flash 4 MB), 0020 0000$_H$ to 002F FFFF$_H$ (Code Flash 3 MB)	✓	—	✓	✓
	Extended User Area	—	✓	—	✓
	Data Flash Memory	—	—	—	—

Note: ✓: possible, —: not possible

Note 1. Extended user area cannot be written in User Boot mode.

Table 44.8 Conditions under which Background Operation is Usable (RH850/F1KM-S1)

Range for Writing	Range for Reading
Data flash memory	Code flash memory

44.7.3 Enabling Self-Programming

The self-programming function can be activated in normal operating mode and user boot mode.

Erasure and programming of the code flash memory by the self-programming function is enabled by making the FLMD0 pin high level.

This prevents unnecessary overwriting of the program if the device operates incorrectly.

The FLMD0 pin is made high level by using one of the following methods.

- The FLMD0 pin is externally pulled up.
- The FLMD0 pin is pulled up by the FLMDCNT register.

The outline of the FLMDCNT register is described in **Section 44.7.3.1, FLMDCNT Register**.

Table 44.9 List of Register

Module Name	Register Name	Symbol	Address
FLMD	FLMDCNT register	FLMDCNT	FFA0 0000_H

44.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

The correct write sequence using the FLMDPCMD register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFA0 0000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPUP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.10 FLMDCNT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FLMDPUP	FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected

44.8 Reading Flash Memory

44.8.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception.

44.8.2 Reading Data Flash Memory

Configure the number of read cycles in the EEPRDCYCL register prior to reading data from data flash memory in normal mode. Once this register is properly configured, data can be read by simply accessing addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programmed again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

Table 44.11 List of Registers

Module Name	Register Name	Symbol	Address
DCIB	Data flash wait cycle control register	EEPRDCYCL	FFC5 9810_H
SCDS	Product name storage register	PRDNAME1	FFCD $00D0_H$
	Product name storage register	PRDNAME2	FFCD $00D4_H$
	Product name storage register	PRDNAME3	FFCD $00D8_H$
	Chip ID register 1 LL	CHIPID1LL	FFCD $00E0_H$
	Chip ID register 1 LH	CHIPID1LH	FFCD $00E4_H$
	Chip ID register 1 HL	CHIPID1HL	FFCD $00E8_H$
	Chip ID register 1 HH	CHIPID1HH	FFCD $00EC_H$
	Chip ID register 2 LL	CHIPID2LL	FFCD $00F0_H$
	Chip ID register 2 LH	CHIPID2LH	FFCD $00F4_H$
	Chip ID register 2 HL	CHIPID2HL	FFCD $00F8_H$
	Chip ID register 2 HH	CHIPID2HH	FFCD $00FC_H$

44.8.2.1 EEPRDCYCL — Data Flash Wait Cycle Control Register

This register is used to specify the number of wait cycles to be inserted when reading the data in the data flash.

Set the number of wait cycles to be inserted in the clock cycle when reading the data flash according to the operating clock frequency of the CPU (f_{CPUCLK}).

Access:	This register can be read or written in 8-bit units.
Address:	FFC5 9810_H
Value after reset:	$0F_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	WAIT[3:0]			
Value after reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.12 EEPRDCYCL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	WAIT[3:0]	Number of Wait Cycles RH850/F1KH-D8, RH850/F1KM-S4: [CKDIVMD = 1, including products of CPU frequency 160 MHz max.]

WAIT[3:0]	Number of Wait Cycles	CPU Operation Frequency		
		$f_{\text{CPUCLK\_H}}$ ≤ 80 MHz	80 MH z < $f_{\text{CPUCLK\_H}}$ ≤ 160 MHz	160 MHz < $f_{\text{CPUCLK\_H}}$ ≤ 240 MHz
0000	1	✓	Setting prohibited	Setting prohibited
0001	2	✓	✓	Setting prohibited
0010	3	✓	✓	✓
0011	4	✓	✓	✓
0100	5	✓	✓	✓
0101	6	✓	✓	✓
0110	7	✓	✓	✓
0111	8	✓	✓	✓
1000	9	✓	✓	✓
Other than above	10	✓	✓	✓

[CKDIVMD = 0]

WAIT[3:0]	Number of Wait Cycles	CPU Operation Frequency		
		$f_{\text{CPUCLK\_H}}$ ≤ 40 MHz	40 MH z < $f_{\text{CPUCLK\_H}}$ ≤ 80 MHz	80 MHz < $f_{\text{CPUCLK\_H}}$ ≤ 120 MHz
0000	1	✓	Setting prohibited	Setting prohibited
0001	2	✓	✓	Setting prohibited
0010	3	✓	✓	✓
0011	4	✓	✓	✓
0100	5	✓	✓	✓
0101	6	✓	✓	✓
0110	7	✓	✓	✓
0111	8	✓	✓	✓
1000	9	✓	✓	✓
Other than above	10	✓	✓	✓

RH850/F1KM-S1:

WAIT[3:0]	Number of Wait Cycles	CPU Operation Frequency		
		$f_{\text{CPUCLK\_M}}$ ≤ 40 MHz	40 MHz < $f_{\text{CPUCLK\_M}}$ ≤ 80 MHz	80 MHz < $f_{\text{CPUCLK\_M}}$ ≤ 120 MHz
0000	1	✓	Setting prohibited	Setting prohibited
0001	2	✓	✓	Setting prohibited
0010	3	✓	✓	✓
0011	4	✓	✓	✓
0100	5	✓	✓	✓
0101	6	✓	✓	✓
0110	7	✓	✓	✓
0111	8	✓	✓	✓
1000	9	✓	✓	✓
Other than above	10	✓	✓	✓

NOTES

1. The read access time to the data flash is calculated by the number of wait cycles.
   RH850/F1KH-D8, RH850/F1KM-S4:
   Read access time to the data flash = {20 + (Number of wait cycles × 8)} / CPU operating frequency
   (CKDIVMD = 1, including products of CPU frequency 160 MHz max.)
   Read access time to the data flash = {20 + (Number of wait cycles × 4)} / CPU operating frequency
   (CKDIVMD = 0)
   RH850/F1KM-S1:
   Read access time to the data flash = {17 + (Number of wait cycles × 4)} / CPU operating frequency
   However, the time may be changed depending on the combination of instructions before and after the execution.
2. ✓ indicates the number of wait cycles that can be set.

44.8.2.2 PRDNAMEn — Product Name Storage Register (n = 1 to 3)

This register stores the product name. The product part name is stored in 16-byte ASCII code, and PRDNAME1, PRDNAME2, and PRDNAME3 correspond to the fourth to first bytes, eighth to fifth bytes, and twelfth to ninth bytes of the product part name respectively.

Access: These registers are read-only registers that can be read in 32-bit units.

Address: PRDNAME1: FFCD 00D0$_H$

PRDNAME2: FFCD 00D4$_H$

PRDNAME3: FFCD 00D8$_H$

Value after reset: See **Table 44.14, List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KH-D8)**, **Table 44.16, List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KM-S4)**, and **Table 44.18, List of Registers Related to Product Information (RH850/F1KM-S1)**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRDNAMEn[31:24]*1								PRDNAMEn[23:16]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRDNAMEn[15:8]*1								PRDNAMEn[7:0]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. n = 1 to 3.

Table 44.13 PRDNAMEn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Product name fourth byte (PRDNAME1), eighth byte (PRDNAME2) twelfth byte (PRDNAME3)
23 to 16	—	Product name third byte (PRDNAME1), seventh byte (PRDNAME2) eleventh byte (PRDNAME3)
15 to 8	—	Product name second byte (PRDNAME1), sixth byte (PRDNAME2) tenth byte (PRDNAME3)
7 to 0	—	Product name first byte (PRDNAME1), fifth byte (PRDNAME2) ninth byte (PRDNAME3)

Table 44.14, List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KH-D8), **Table 44.16, List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KM-S4)**, and **Table 44.18, List of Registers Related to Product Information (RH850/F1KM-S1)** list registers related to product information.

Table 44.14 List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KH-D8)

Product Part Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701715	3746 3752	3137 3130	2020 2035
R7F701714	3746 3752	3137 3130	2020 2034
R7F701711	3746 3752	3137 3130	2020 2031
R7F701710	3746 3752	3137 3130	2020 2030
R7F701709	3746 3752	3037 3130	2020 2039
R7F701708	3746 3752	3037 3130	2020 2038

Table 44.15 Reserved

Table 44.16 List of Registers Related to Product Information (Products of CPU frequency 240 MHz max.) (RH850/F1KM-S4)

Product Part Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701653	3746 3752	3536 3130	2020 2033
R7F701652	3746 3752	3536 3130	2020 2032
R7F701651	3746 3752	3536 3130	2020 2031
R7F701650	3746 3752	3536 3130	2020 2030
R7F701649	3746 3752	3436 3130	2020 2039
R7F701648	3746 3752	3436 3130	2020 2038
R7F701647	3746 3752	3436 3130	2020 2037
R7F701646	3746 3752	3436 3130	2020 2036
R7F701645	3746 3752	3436 3130	2020 2035
R7F701644	3746 3752	3436 3130	2020 2034

Table 44.17 Reserved

Table 44.18 List of Registers Related to Product Information (RH850/F1KM-S1)

Product Part Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701684	3746 3752	3836 3130	2020 2034
R7F701685	3746 3752	3836 3130	2020 2035
R7F701686	3746 3752	3836 3130	2020 2036
R7F701687	3746 3752	3836 3130	2020 2037
R7F701688	3746 3752	3836 3130	2020 2038
R7F701689	3746 3752	3836 3130	2020 2039
R7F701690	3746 3752	3936 3130	2020 2030
R7F701691	3746 3752	3936 3130	2020 2031
R7F701692	3746 3752	3936 3130	2020 2032
R7F701693	3746 3752	3936 3130	2020 2033
R7F701694	3746 3752	3936 3130	2020 2034
R7F701695	3746 3752	3936 3130	2020 2035

44.8.2.3 CHIPIDnXX — Chip ID Register (n = 1, 2, XX = LL, LH, HL, HH)

The RH850/F1KH and RH850/F1KM provides the option of a unique chip ID of the device.

Access: These registers are read-only registers that can be read in 32-bit units.

Address: CHIPID1LL: FFCD $00E0_H$
CHIPID1LH: FFCD $00E4_H$
CHIPID1HL: FFCD $00E8_H$
CHIPID1HH: FFCD $00EC_H$
CHIPID2LL: FFCD $00F0_H$
CHIPID2LH: FFCD $00F4_H$
CHIPID2HL: FFCD $00F8_H$
CHIPID2HH: FFCD $00FC_H$

Value after reset: Unique value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHIPIDnXX[31:24]*1								CHIPIDnXX[23:16]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CHIPIDnXX[15:8]*1								CHIPIDnXX[7:0]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. n = 1, 2
XX = LL, LH, HL, HH

44.9 Option Bytes

The option bytes of the flash memory are an expansion area and hold data specified by the user for a variety of purposes. Initial settings for peripheral modules and so on as specified by the option bytes become effective on release from the reset state. **Table 44.19, Value of Shipping (RH850/F1KH-D8)**, **Table 44.20, Value of Shipping (RH850/F1KM-S4, RH850/F1KM-S1)** shows the value of shipping.

Table 44.19 Value of Shipping (RH850/F1KH-D8)

Option Byte	Value
Option byte 0	FA27 FFCF$_H$
Option byte 1	FFFF FDF4$_H$

Table 44.20 Value of Shipping (RH850/F1KM-S4, RH850/F1KM-S1)

Option Byte	Value
Option byte 0	FA27 FFCF$_H$
Option byte 1	FFFF FDFF$_H$

44.9.1 Option Byte Setting

Be sure to set the option byte area that corresponds to the optional functions listed below, before writing a program to the flash memory.

The optional functions specified by the option bytes are as follows.

- Function of port group JP0
- Activation code method of WDTA2*[2]
- Start mode of WDTA2*[2]
- Enabling or disabling WDTA2*[2]
- Activation code method of WDTA1
- Start mode of WDTA1
- Enabling or disabling WDTA1
- Activation code method of WDTA0
- Start mode of WDTA0
- Enabling or disabling WDTA0
- Initial value of the overflow interval time for WDTA2*[2], WDTA1 and WDTA0
- Enabling or disabling of $\overline{\text{RESETOUT}}$
- Enabling the high voltage monitor
- Enabling the low voltage monitor
- Clock divider mode for CPUCLK*[1]
- Enabling or disabling FlexRay*[1]
- Enabling or disabling Ethernet AVB 0*[1]
- Enabling or disabling Ethernet AVB 1*[2]
- Enabling or disabling PE2*[2]
- Enabling or disabling PE2 Boot*[2]

Note 1. This function is not supported in RH850/F1KM-S1.

Note 2. This function is supported in RH850/F1KH-D8.

44.9.2 OPBT0 — Option Byte 0

The settings and bit positions of the option bytes are listed below.

For details on how to set an option byte, see the user’s manual of the flash programmer for serial programming as well as the code flash library for self-programming.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	OPJTAG[1:0]		—	—	WDT 1_3	—	WDT 1_1	WDT 1_0	WDT 0_3	—	WDT 0_1	WDT 0_0	WDT_2	WDT_1	WDT_0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETNB1 EN	ETNB0 EN	FLXA0 EN	CKDIV MD	—	—	RESET OUTEN	—	—	—	CVM_H D_EN	CVM_L D_EN	PE2BO OTEN	PE2EN	—	—

Note: Value after reset: user defined (values of shipping are shown in **Table 44.19, Value of Shipping (RH850/F1KH-D8)** and **Table 44.20, Value of Shipping (RH850/F1KM-S4, RH850/F1KM-S1)**).

Table 44.21 Option Byte 0 Settings

Bit Position	Bit Name	Function
31	Reserved	When writing, write “1”.
30, 29	OPJTAG[1:0]	These bits control the function of port group JP0. 00: JP0 is used for general purpose/alternative function port. 01: JP0 is used for LPD 4-pin mode. 10: JP0 is used for LPD 1-pin mode. 11: JP0 is used for Nexus I/F.
28, 27	Reserved	When writing, write “1”.
26	WDT1_3	Specifies the activation code method of WDTA1. 0: Fixed activation code 1: Variable activation code
25	Reserved	When writing, write “1”.
24	WDT1_1	Specifies the start mode of WDTA1. 0: Software trigger start mode 1: Default start mode
23	WDT1_0	Enables or disables WDTA1. 0: WDTA1 is disabled 1: WDTA1 is enabled
22	WDT0_3	Specifies the activation code method of WDTA0. 0: Fixed activation code 1: Variable activation code
21	Reserved	When writing, write “1”.
20	WDT0_1	Specifies the start mode of WDTA0. 0: Software trigger start mode 1: Default start mode
19	WDT0_0	Enables or disables WDTA0. 0: WDTA0 is disabled 1: WDTA0 is enabled
18 to 16	WDT_[2:0]	Control of the overflow interval time for WDTA0, WDTA1 and WDTA2. These bits specify the reset value of WDTAnMD.WDTAnOVF[2:0].
15	ETNB1EN	RH850/F1KH-D8: Enables or disables Ethernet AVB 1 0: Ethernet AVB 1 is disabled. 1: Ethernet AVB 1 is enabled. RH850/F1KM-S4, RH850/F1KM-S1: When writing, write “1”.

Table 44.21 Option Byte 0 Settings

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>14</td><td>ETNB0EN</td><td>RH850/F1KH-D8, RH850/F1KM-S4:
Enables or disables Ethernet AVB 0
0: Ethernet AVB 0 is disabled.
1: Ethernet AVB 0 is enabled.
RH850/F1KM-S1:
When writing, write “1”.</td></tr>
<tr><td>13</td><td>FLXA0EN</td><td>RH850/F1KH-D8, RH850/F1KM-S4:
Enables or disables FlexRay
0: FlexRay is disabled.
1: FlexRay is enabled.
RH850/F1KM-S1:
When writing, write “1”.</td></tr>
<tr><td>12</td><td>CKDIVMD</td><td>RH850/F1KH-D8, RH850/F1KM-S4:
Clock divider mode for CPUCLK
<table>
<tr><th></th><th colspan="2">Products of CPU frequency 240 MHz max.</th><th>Products of CPU frequency 160 MHz max.</th></tr>
<tr><td>CKDIVMD</td><td>1</td><td>0[*1]</td><td>1[*2]</td></tr>
<tr><td>CPUCLK_H</td><td>CPUCLK</td><td>CPUCLK</td><td>CPUCLK</td></tr>
<tr><td>CPUCLK_M</td><td>CPUCLK × 1/2</td><td>CPUCLK</td><td>CPUCLK × 1/2</td></tr>
<tr><td>CPUCLK_L</td><td>CPUCLK × 1/4</td><td>CPUCLK × 1/2</td><td>CPUCLK × 1/4</td></tr>
<tr><td>CPUCLK_UL</td><td>CPUCLK × 1/8</td><td>CPUCLK × 1/4</td><td>CPUCLK × 1/8</td></tr>
</table>
RH850/F1KM-S1:
When writing, write “1”.</td></tr>
<tr><td>11, 10</td><td>Reserved</td><td>When writing, write “1”.</td></tr>
<tr><td>9</td><td>RESETOUTEN</td><td>$\overline{\text{RESETOUT}}$ control
0: $\overline{\text{RESETOUT}}$ disable
1: $\overline{\text{RESETOUT}}$ enable</td></tr>
<tr><td>8 to 6</td><td>Reserved</td><td>When writing, write “1”.</td></tr>
<tr><td>5</td><td>CVM_HD_EN</td><td>High Voltage Monitor Enable
0: Disable high voltage detection
1: Enable high voltage detection</td></tr>
<tr><td>4</td><td>CVM_LD_EN</td><td>Low Voltage Monitor Enable
0: Disable low voltage detection
1: Enable low voltage detection</td></tr>
<tr><td>3</td><td>PE2BOOTEN</td><td>RH850/F1KH-D8:
PE2 Boot enable
0: PE2 Boot disable
1: PE2 Boot enable (only when PE2EN is 1)
RH850/F1KM-S4, RH850/F1KM-S1:
When writing, write “1”.</td></tr>
<tr><td>2</td><td>PE2EN</td><td>RH850/F1KH-D8:
PE2 enable
0: PE2 Disable
1: PE2 Enable
RH850/F1KM-S4, RH850/F1KM-S1:
When writing, write “1”.</td></tr>
<tr><td>1, 0</td><td>Reserved</td><td>When writing, write “1”.</td></tr>
</table>

Note 1. When setting “0”, set CPUCLK to 120 MHz or less.

Note 2. When writing, write “1” for Products of CPU frequency 160 MHz max.

44.9.3 OPBT1 — Option Byte 1

The settings and bit positions of the option bytes are listed below.

For details on how to set an option byte, see the user's manual of the flash programmer for serial programming as well as the code flash library for self-programming.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SIDAM	WDT2_3	—	WDT2_1	WDT2_0

Table 44.22 Option Byte 1 Settings

Bit Position	Bit Name	Function
31 to 10	Reserved	When writing, write "1".
9	Reserved	When writing, write "0".
8 to 5	Reserved	When writing, write "1".
4	SIDAM	Self-Programming ID Authentication Mode 0: Self-Programming ID Authentication is NOT necessary for code flash memory programming and configuration setting for reset vector. 1: Self-Programming ID Authentication is necessary for code flash memory programming and configuration setting for reset vector.
3	WDT2_3	RH850/F1KH-D8: Specifies the activation code method of WDTA2. 0: Fixed activation code 1: Variable activation code RH850/F1KM-S4, RH850/F1KM-S1: When writing, write "1".
2	Reserved	When writing, write "1".
1	WDT2_1	RH850/F1KH-D8: Specifies the start mode of WDTA2. 0: Software trigger start mode 1: Default start mode RH850/F1KM-S4, RH850/F1KM-S1: When writing, write "1".
0	WDT2_0	RH850/F1KH-D8: Enables or disables WDTA2. 0: WDTA2 is disabled 1: WDTA2 is enabled RH850/F1KM-S4, RH850/F1KM-S1: When writing, write "1".

44.10 Usage Notes

(1) Reading areas where Programming or Erasure was Interrupted

When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid the reading out of undefined data, which might cause a malfunction, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.

(2) Reading the Code Flash Memory that has been Erased but Not yet been Programmed Again

Note that reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.

(3) Prohibition of Additional Writing

Writing to a given area two or more times is not possible. When overwriting data in an area of flash memory after writing to the area has been completed, erase the area first.

(4) Resets During Programming and Erasure

In the case of the external reset during programming and erasure, wait for at least the minimum value of $\overline{\text{RESET}}$ input low level width once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming and Erasure

Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.

(6) Abnormal Termination of Programming and Erasure

Even if programming/erasure ends abnormally due to the assertion of a reset by the $\overline{\text{RESET}}$ pin, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to ensure that the corresponding area is completely erased before using.

If programming and erasure of code flash memory are not completed normally, the lock bit for the target area may be enabled (locked). In such cases, erase the block to erase the lock bit while the lock bit is in the disabled state (the area is not locked).

(7) Items Prohibited During Programming, Erasure and Blank check.

Do not perform the following operations during programming, erasure, and blank check.

- Set the operating voltage from the power supply outside the allowed range.

RH850/F1KH-D8, RH850/F1KM-S4:

- Change the frequency of the peripheral clock.

RH850/F1KM-S1:

- Change of all clock register described in **Section 12C, Clock Controller**.

The following steps have to be applied during writing and/or erasing:

(a) Please do not select the HS IntOSC and EMCLK as clock source when writing and/or erasing the code/data flash. This is also applicable when the HS IntOSC is set as PLL1 source clock.

(b) Please disable CLMA0, and CLMA3. (Set CLMA0CTL0.CLMA0CLME = 0, CLMA3CTL0.CLMA3CLME = 0)

(c) Please disable the LPS (DPEN = 0, ADEN = 0) when writing and/or erasing the code/data flash.

If (a) to (c) above are difficult to apply, please contact your local sales representative.

(8) Erasure of all flash memory before clearing the configuration

Erase the user area of code flash and all of the data area of data flash before clearing the configuration.

(9) Restriction on Serial Programming when HS IntOSC is Used

RH850/F1KH-D8, RH850/F1KM-S4:

- 1-wire/2-wire UART mode cannot be used.
- The E1 emulator cannot be used.

RH850/F1KM-S1:

Please do not select the HS IntOSC and EMCLK as clock source during serial programming mode.

When the HS IntOSC and EMCLK are set, there is a possibility that a failure occurs during flash write or flash erase. In case the flash write and flash erase process are finished successfully, the write and erase can be guaranteed.

Section 45 RAM

This section describes the RAM mounted on RH850/F1KH, RH850/F1KM.

45.1 Features of RH850/F1KH, RH850/F1KM RAM

- RH850/F1KH and RH850/F1KM includes the following RAMs:
  - Local RAM (LRAM)
    The local RAM is accessible at high speed. Values in the local RAM are not retained in DeepSTOP mode.
  - Global RAM (GRAM)
    RH850/F1KH-D8:
    The global RAM is used for serving as a source or destination for DMA transfer. The global RAM is divided into banks A and B and parallel access to both banks is possible. Values in global RAM are not retained in DeepSTOP mode.
    RH850/F1KM-S4:
    The global RAM is used for serving as a source or destination for DMA transfer. The global RAM is divided into banks A and B and parallel access to both banks is possible. Values in global RAM are not retained in DeepSTOP mode.
    RH850/F1KM-S1:
    There is no global RAM in RH850/F1KM-S1
  - Retention RAM (RRAM)
    RH850/F1KH-D8:
    The retention RAM is a part of the global RAM (Bank B). Values in the retention RAM are retained in DeepSTOP mode.
    In addition, even if the power-supply voltage (REG0VCC / REG1VCC) falls below the POC voltage, data in the retention RAM are retained as long as the voltage does not fall below the RAM retention voltage (VVLVI).
    RH850/F1KM-S4:
    The retention RAM is a part of the global RAM (Bank B). Values in the retention RAM are retained in DeepSTOP mode.
    In addition, even if the power-supply voltage (REGVCC) falls below the POC voltage, data in the retention RAM are retained as long as the voltage does not fall below the RAM retention voltage (VVLVI).
    RH850/F1KM-S1:
    The retention RAM is a part of the local RAM and is also accessible at high speed. Values in the retention RAM are retained in DeepSTOP mode.
    In addition, even if the power-supply voltage (REGVCC) falls below the POC voltage, data in the retention RAM are retained as long as the voltage does not fall below the RAM retention voltage (VVLVI).

Access time for each RAM is shown in the table below.

Table 45.1 RAM Access Time (RH850/F1KH-D8)

Type of Access	RAM	1st Access (CPUCLK_H)	Continuous Access (CPUCLK_H)
Instruction fetch	Local RAM (own PE)	6	6
	Global RAM	6	6
	Retention RAM	6	6
Read access	Local RAM (own PE)	1	1
	Local RAM (other PE)	6	4
	Global RAM	6	2
	Retention RAM	6	2
Write access	Local RAM (own PE)	1	1
	Local RAM (other PE)	1 (Posted)	3
	Global RAM	1 (Posted)	2 (Posted)
	Retention RAM	1 (Posted)	2 (Posted)

Table 45.2 RAM Access Time (RH850/F1KM-S4)

Type of Access	RAM	1st Access (CPUCLK_H)	Continuous Access (CPUCLK_H)
Instruction fetch	Local RAM	6	6
	Global RAM	6	6
	Retention RAM	6	6
Read access	Local RAM	1	1
	Global RAM	6	2
	Retention RAM	6	2
Write access	Local RAM	1	1
	Global RAM	1 (Posted)	2 (Posted)
	Retention RAM	1 (Posted)	2 (Posted)

Table 45.3 RAM Access Time (RH850/F1KM-S1)

Type of Access	RAM	1st Access (CPUCLK_M)	Continuous Access (CPUCLK_M)
Instruction fetch	Local RAM	5	5
	Retention RAM	5	5
Read access	Local RAM	1	1
	Retention RAM	1	1
Write access	Local RAM	1	1
	Retention RAM	1	1

NOTE

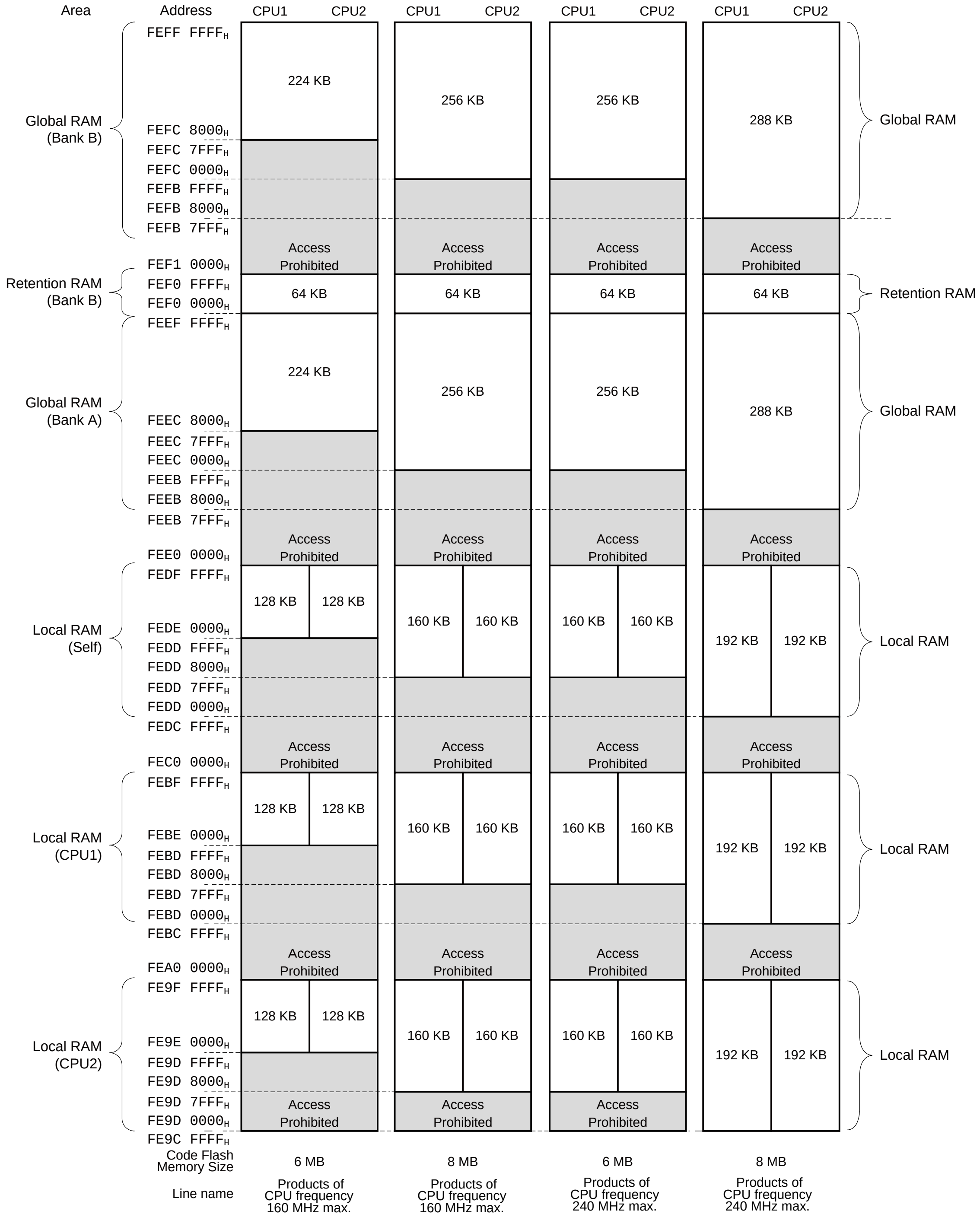
There is possibility that number of access clock of above table is changed depending on the combination before and after instructions.
When RAM access is misaligned, these number is increased.

- Error detection/correction function (ECC) in the local RAM, global RAM and retention RAM

  The ECC function is included, which can detect 2-bit errors and detect/correct 1-bit errors. For details, see **Section 40A, Functional Safety of RH850/F1KH-D8**, **Section 40B, Functional Safety of RH850/F1KM-S4** and **Section 40C, Functional Safety of RH850/F1KM-S1**.

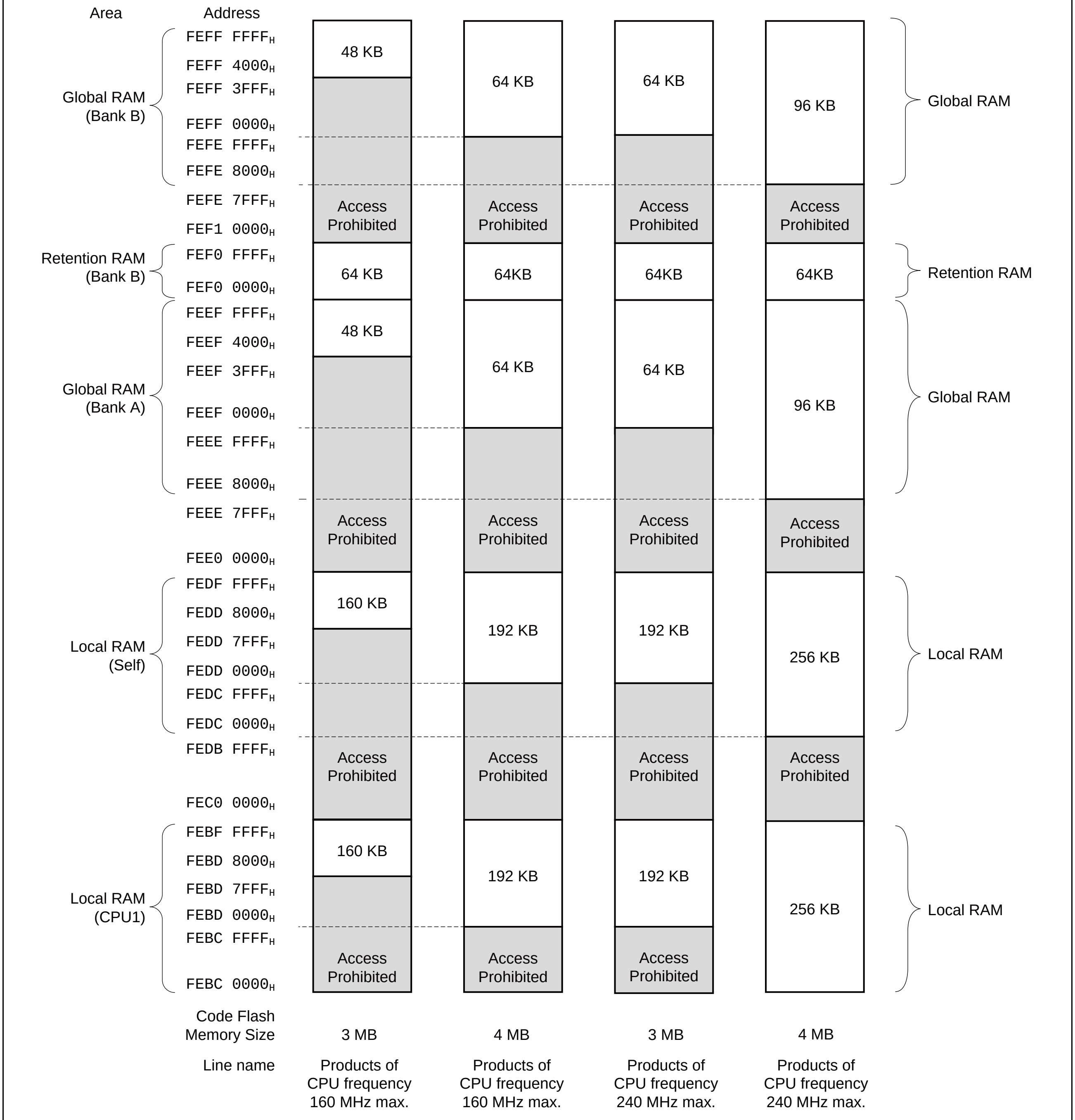
45.2 Memory Configuration

Figure 45.1, Memory Map of the Local RAM, the Global RAM and the Retention RAM (RH850/F1KH-D8), **Figure 45.2, Memory Map of the Local RAM, the Global RAM and the Retention RAM (RH850/F1KM-S4)** and **Figure 45.3, Memory Map of the Local RAM and the Retention RAM (RH850/F1KM-S1)** show the memory map of the local RAM, global RAM and the retention RAM.



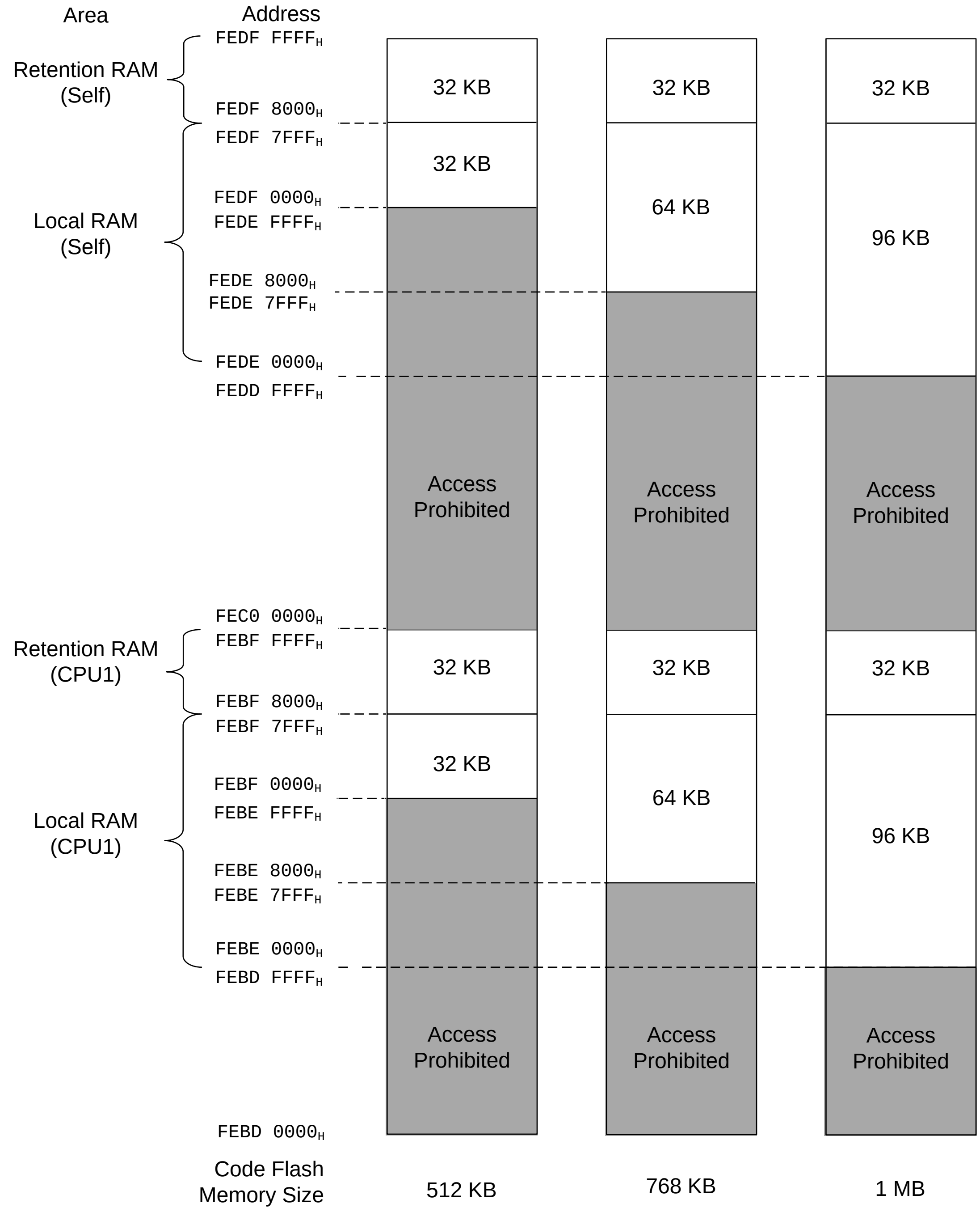
Note: Access prohibited area is prohibited, operation is not guaranteed.

Figure 45.1 Memory Map of the Local RAM, the Global RAM and the Retention RAM (RH850/F1KH-D8)



Note: Access prohibited area is prohibited, operation is not guaranteed.

Figure 45.2 Memory Map of the Local RAM, the Global RAM and the Retention RAM (RH850/F1KM-S4)



Note: Access prohibited area is prohibited, operation is not guaranteed.

Figure 45.3 Memory Map of the Local RAM and the Retention RAM (RH850/F1KM-S1)

45.3 Global RAM Zero Fill (GRZF)

45.3.1 Number of Units

This microcontroller has the following number of GRZF units.

Table 45.4 Number of Units (RH850/F1KH-D8)

Product Name	RH850/F1KH-D8 176 Pins	RH850/F1KH-D8 233 Pins	RH850/F1KH-D8 324 Pins
Number of Units	1	1	1
Name	GRZF	GRZF	GRZF

Table 45.5 Number of Units (RH850/F1KM-S4)

Product Name	RH850/F1KM-S4 100 Pins	RH850/F1KM-S4 144 Pins	RH850/F1KM-S4 176 Pins	RH850/F1KM-S4 233 Pins	RH850/F1KM-S4 272 Pins
Number of Units	1	1	1	1	1
Name	GRZF	GRZF	GRZF	GRZF	GRZF

Table 45.6 Number of Units (RH850/F1KM-S1)

Product Name	RH850/F1KM-S1 48 Pins	RH850/F1KM-S1 64 Pins	RH850/F1KM-S1 80 Pins	RH850/F1KM-S1 100 Pins
Number of Units	—	—	—	—
Name	—	—	—	—

45.3.2 Register Base Address

GRZF base address is listed in the following table.

GRZF register addresses are given as an offset from the base address in general.

Table 45.7 Register Base Address (RH850/F1KH-D8)

Base Address Name	Base Address
<GRZF_base>	FFC0 3000_H

Table 45.8 Register Base Address (RH850/F1KM-S4)

Base Address Name	Base Address
<GRZF_base>	FFC0 3000_H

45.3.3 Clock Supply

The GRZF clock supply is shown in following table.

Table 45.9 Clock Supply (RH850/F1KH-D8)

Unit Name	Unit Clock Name	Supply Clock Name
GRZF	Unit clock	CPUCLK_H
	Register access	CPUCLK_UL

Table 45.10 Clock Supply (RH850/F1KM-S4)

Unit Name	Unit Clock Name	Supply Clock Name
GRZF	Unit clock	CPUCLK_H
	Register access	CPUCLK_UL

45.3.4 Interrupt Requests

GRZF interrupt requests are listed in the following table:

Table 45.11 Interrupt Request (RH850/F1KH-D8)

Unit Interrupt Signal	Description	Interrupt Number
INTGRZF	Interrupt for GRZF	376

Table 45.12 Interrupt Request (RH850/F1KM-S4)

Unit Interrupt Signal	Description	Interrupt Number
INTGRZF	Interrupt for GRZF	376

45.3.5 Reset Sources

GRZF reset sources are listed in the following table. GRZF is initialized by these reset sources the All reset sources.

Table 45.13 Reset Sources (RH850/F1KH-D8)

Unit Name	Reset Sources
GRZF	All reset sources (ISORES)

Table 45.14 Reset Sources (RH850/F1KM-S4)

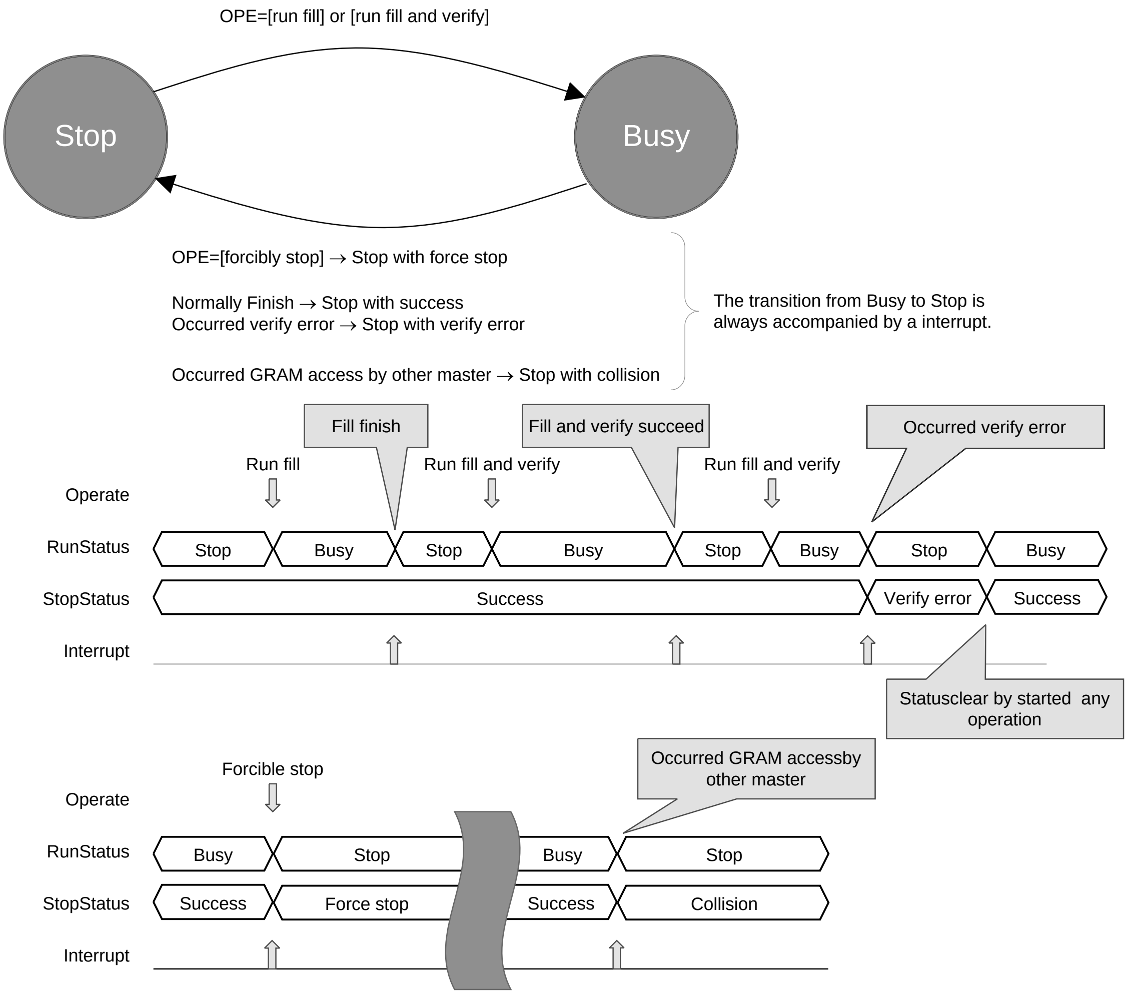
Unit Name	Reset Sources
GRZF	All reset sources (ISORES)

45.3.6 Functional Overview

This function is memory ZERO fill and verify check for GRAM area (including RRAM)

GRZF cannot arbitrate access from other bus-master. So, when GRZF is in busy status, do not access the same bank on GRAM that GRZF is accessing. If other bus-master accesses to the same bank that GRZF is accessing, the operation of GRZF is terminated and INTGRZF occurs with error status. Also, the access of other bus master will not be fulfilled (Writing is ignored, and read data cannot be guaranteed).

Other bus-master can access to the bank that GRZF is not accessing.



Note: ● Necessary cycle@CPUCLK_H base

RH850/F1KH-D8:
- Only fill: size (word) × (2 + WAIT^{*1})
- Fill and verify: size (word) × (4 + 2 x WAIT^{*1})

Note 1. WAIT value depends on GRZFCTL.OPE[5:4] setting with operation start trigger.

RH850/F1KM-S4:
- Only fill: size (word) × 2
- Fill and verify: size (word) × 4

Figure 45.4 The Memory ZERO Fill and Verify Check for GRAM Area (including RRAM) Function

45.3.7 Registers

45.3.7.1 List of Registers

GRZF registers are listed in the following table.

For details about <GRZF_base>, see **Section 45.3.2, Register Base Address**.

Table 45.15 List of Registers

Module Name	Register Name	Symbol	Address
GRZF	GRZF control register	GRZFCTL	<GRZF_base> + 00_H
	GRZF status register	GRZFSTS	<GRZF_base> + 04_H
	GRZF start address setting register	GRZFSA	<GRZF_base> + 08_H
	GRZF fill size setting register	GRZFSZ	<GRZF_base> + $0C_H$

45.3.7.2 GRZFCTL — GRZF Control Register

This register is used to configure various operation settings of the global RAM Zero Fill.

Access: This register is a write-only register that can be written in 32-bit units in RH850/F1KM-S4.

This register can be read or written in 32-bit units in RH850/F1KH-D8.

Address: <GRZF_base> + 00$_\text{H}$

Value after reset: 0000 0000$_\text{H}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KEY2[7:0]								KEY1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY0[7:0]								OPE[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	R^{*1}/W	R^{*1}/W	W	W	W	W

Note 1. These bits can read in RH850/F1KH-D8.

Table 45.16 GRZFCTL Register Contents

Bit Position	Bit Name	Function
31 to 24	KEY2[7:0]	Must be write fix value(C0$_\text{H}$) if any operation.
23 to 16	KEY1[7:0]	Must be write fix value(FF$_\text{H}$) if any operation.
15 to 8	KEY0[7:0]	Must be write fix value(EE$_\text{H}$) if any operation.
7 to 0	OPE[7:0]	Operation type. 0000 0000$_\text{B}$: forcibly stop 0000 0001$_\text{B}$: run zero fill 0000 0010$_\text{B}$: run zero fill and verify 0001 0001$_\text{B}$: run zero fill wait 1 cycle^{*1} 0001 0010$_\text{B}$: run zero fill and verify wait 1 cycle^{*1} 0010 0001$_\text{B}$: run zero fill wait 2 cycles^{*1} 0010 0010$_\text{B}$: run zero fill and verify wait 2 cycles^{*1} 0011 0001$_\text{B}$: run zero fill wait 3 cycles^{*1} 0011 0010$_\text{B}$: run zero fill and verify wait 3 cycles^{*1} other than above: setting prohibited Execution of these operations, you must write the KEY0-2 at the same time.

Note 1. Bits [5:4] are used to set up the WAIT function in RH850/F1KH-D8.
01$_\text{B}$: Wait 1 cycle CPUCLK_H
10$_\text{B}$: Wait 2 cycles CPUCLK_H
11$_\text{B}$: Wait 3 cycles CPUCLK_H

45.3.7.3 GRZFSTS — GRZF Status Register

This register is the status of the memory ZERO fill performed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GRZF_base> + 04_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSTS	—	—	—	—	—	SSTS[1:0]		—	—	—	PSTS[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.17 GRZFSTS Register Contents

Bit Position	Bit Name	Function
31	RSTS	Run status 0: Stop 1: Busy
30 to 26	Reserved	When read, the value after reset is returned.
25, 24	SSTS[1:0]	Stop status 00_B: No error 01_B: Forcibly stop 10_B: Stop with verify error 11_B: Stop with collision other bus-master SSTS will be cleared at the next startup
23 to 21	Reserved	When read, the value after reset is returned.
20 to 0	PSTS[20:0]	Progress status (executed address)

NOTE

The value of progress address indicated by the PSTS is not guaranteed, if stopped by the forcibly stop or contention access of other bus master. Please restart without changing value of the start address and size.

45.3.7.4 GRZFSA — GRZF Start Address Setting Register

This register is used to set start address of the memory ZERO fill performed.

Access: This register can be read or written in 32-bit units.

Address: <GRZF_base> + 08_H

Value after reset: FEE0 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OFST[10:0]											BANK	START[19:16]			
Value after reset	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	START[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Table 45.18 GRZFSA Register Contents

Bit Position	Bit Name	Function
31 to 21	OFST[10:0]	These bits are fixed to $7F7_H$ to indicate the GRAM start address configured when this register is read in 32-bit units.
20	BANK	Target Bank of global RAM 0: BankA 1: BankB (Including retention RAM)
19 to 0	START[19:0]	Target Start Address (bits 9 to 0 is fixed to 0) 1 KB unit

NOTE

This register can be rewritten only when the Stop state. Writing in operation will be ignored.

45.3.7.5 GRZFSZ — GRZF Fill Size Setting Register

This register is used to set fill size of the memory ZERO fill performed

Access: This register can be read or written in 32-bit units.

Address: <GRZF_base> + 0C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SIZE[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIZE[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Table 45.19 GRZFSZ Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	SIZE[19:0]	Memory fill size setting (bits 9 to 0 is fixed to 0). Size the actual initialization will be the SIZE + 1 KB.

CAUTIONS

1. This register can be rewritten only when the Stop state. Writing in operation will be ignored.
2. The size setting across BankA and BankB is prohibited.

45.3.8 Operation

This module effects only global RAM bank A and global RAM bank B. The RRAM area belong to global RAM bank B. So this module also effects the RRAM area.

Global RAM Zero Fill (GRZF) consists of the following function:

- FILL function

  Write zero to GRAM

- VERI function
  Write zero to GRAM and verify the data GRAM have been zero

The transition state from Busy state to Stop state always generate the Interrupt request.

The cause for transition from Busy state to Stop state are shown below:

- Success in FILL function to all the target banks.
- Conflict occur when the other master accesses the target bank while running.
- Verify error occur when the RAM area is not zero at VERI function.
- Forcibly stop by GRZFCTL.OPE[7:0] = 0000 $0000_{B.}$

45.3.8.1 Procedure for FILL and VERI Function

The procedure for FILL and VERI function is shown below.

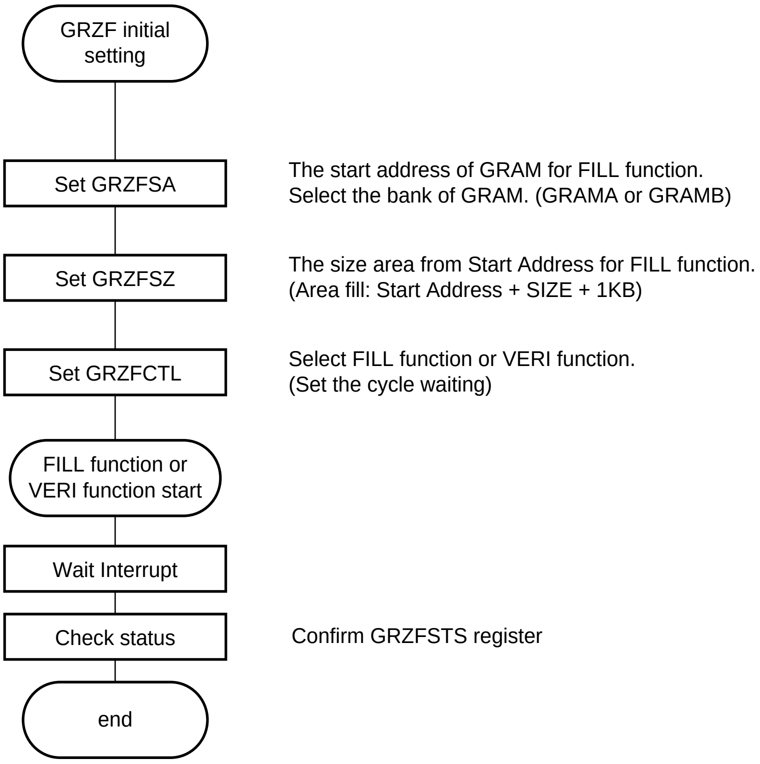


Figure 45.5 FILL and VERI Flow

45.3.8.2 Interrupt Sources

GRZF can generate the following interrupt requests:

- The interrupt occurs when transition from Busy to Stop in both FILL function and VERI function.
- When an interrupt occurred, the source of the interrupt is identified by checking GRZFSTS.SSTS. This status flag holds the value until the next startup (accordingly, does not change when the GRZF module transition from Busy to Stop).

45.4 Usage Notes

- Before accessing the global, local or retention RAM with ECC error detection and correction enabled, initialize the RAM by setting the access size to the largest bit length.
- Accessing the RAM before initializing it may lead to the detection of ECC errors. ECC errors may be detected if initialization is not handled in the maximum unit of access, for example, when 32-bit word RAM is accessed in 8- or 16-bit units.

The following notes apply to the GRZF. See **Figure 40A.2, Block Diagram of Global RAM ECC** and **Figure 40B.2, Block Diagram of Global RAM ECC**.

- When GRZF accesses to GRAM, GRZF does not use ECC encoder or decoder. When there is any differences between the read value and the write value in 39 bits including GRZF 32 bits (all 0) and ECC code 7 bits, a verify error is occurred in GRZF. The verify error can be checked by the GRZFSTS.SSTS[1:0].
- Global RAM Guard (GRG) cannot guard the access from GRZF to GRAM.

Section 46 Boundary Scan

This section contains a generic description of boundary scan.

The RH850/F1KH, RH850/F1KM has a JTAG interface and provides a boundary scan function.

46.1 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1 that is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of the RH850/F1KH, RH850/F1KM conforms to IEEE Std 1149.1-2001.

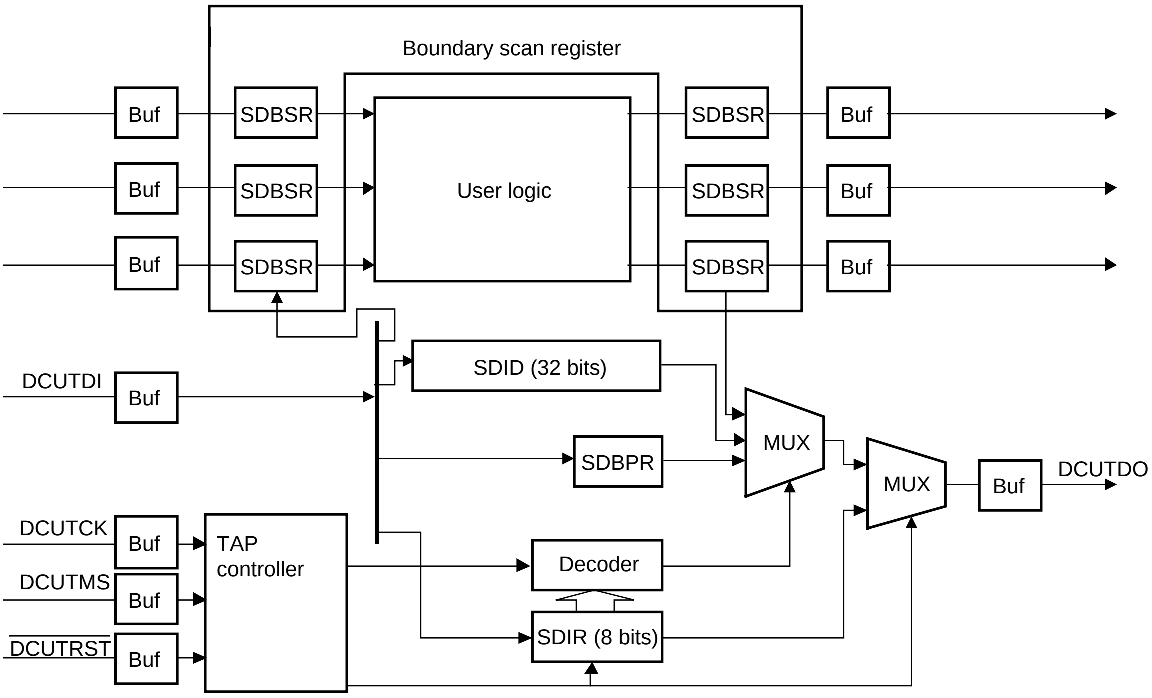
46.2 Features of RH850/F1KH, RH850/F1KM Boundary Scan

- Five control signals (DCUTCK, DCUTDI, DCUTDO, DCUTMS, and $\overline{\text{DCUTRST}}$)
- TAP controller
- Instruction register
- Bypass register
- Boundary scan register

The JTAG interface has four instructions.

- BYPASS
  Test mode conforming to the IEEE 1149.1
- EXTEST
  Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD
  Test mode conforming to the IEEE 1149.1
- IDCODE
  Test mode conforming to the IEEE 1149.1

Figure 46.1, Block Diagram of JTAG Interface shows a block diagram of the JTAG interface.



Note: SDBSR: Boundary scan register
SDBPR: Bypass register
Buf: I/O buffer
SDIR: Instruction register
SDID: ID register
MUX: Multiplexer

Figure 46.1 Block Diagram of JTAG Interface

46.3 External Input/Output Pins

There are five JTAG control signals: DCUTCK, DCUTDI, DCUTMS, DCUTDO, and $\overline{\text{DCUTRST}}$.

Table 46.1, Pin Configuration shows the pin configuration.

Table 46.1 Pin Configuration

Pin Name	Description
DCUTCK	Serial data input/output clock pin Data is input to DCUTDI and is output from DCUTDO in synchronization with this clock signal.
DCUTMS	Mode select input pin Changing the level of this signal in synchronization with DCUTCK changes the state of the TAP controller. For the protocol, see **Figure 46.2, TAP Controller State Transition Diagram**.
$\overline{\text{DCUTRST}}$	Reset input pin A low-level input of this signal resets the JTAG interface. This signal is accepted asynchronously with DCUTCK.
DCUTDI	Serial data input pin Data is input in synchronization with DCUTCK and sent to the JTAG interface.
DCUTDO	Serial data output pin Data to be read from the JTAG interface is output in synchronization with DCUTCK.

46.4 Registers

The JTAG interface has the following registers. None of the registers can be accessed by the CPU.

- SDIR: Instruction register
- SDID: ID register
- SDBPR: Bypass register
- SDBSR: Boundary scan register

Table 46.2 Register Configuration

Register Name	Symbol	Access Size	Value After Reset*[1]
Instruction register	SDIR	8	55_H
ID register	SDID	32	See **Table 46.5, ID Register Codes (RH850/F1KH-D8)**, **Table 46.6, ID Register Codes (RH850/F1KM-S4)**, **Table 46.7, ID Register Codes (RH850/F1KM-S1)**.
Bypass register	SDBPR	1	Undefined
Boundary scan register	SDBSR	—	Undefined

Note 1. Registers are initialized when $\overline{\text{DCUTRST}}$ pin is 0 or when TAP is in the Test-Logic-Reset state.

Instructions can be serially transferred from the serial data input pin (DCUTDI) and input to the instruction register (SDIR). The bypass register (SDBPR) is a 1-bit register, to which DCUTDI and DCUTDO are connected in BYPASS mode. The boundary scan register (SDBSR) is connected to DCUTDI and DCUTDO in SAMPLE/PRELOAD mode and EXTEST mode. The ID register (SDID) is a 32-bit register, from which the fixed code is output via DCUTDO in IDCODE mode.

Table 46.3, Serial Transfer Types shows the serial transfer types possible with the JTAG interface registers.

Table 46.3 Serial Transfer Types

Register	Serial Input	Serial Output
SDIR	Possible	Impossible*[1]
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDID	Impossible	Possible

Note 1. A fixed value is read out.

46.4.1 Instruction Register (SDIR)

The instruction register (SDIR) is an 8-bit register that holds a boundary scan instruction. SDIR is initialized by a low-level input of $\overline{\text{DCUTRST}}$ or in the TAP Test-Logic-Reset state. Operation is not guaranteed if a reserved instruction is set in this register.

Table 46.4 Boundary Scan Instructions

Instruction Code								Description
0	0	0	0	0	0	0	0	JTAG EXTEST
0	1	0	0	0	0	0	0	JTAG SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	JTAG IDCODE (initial value)
1	1	1	1	1	1	1	1	JTAG BYPASS
Other than above								Reserved

46.4.2 ID Register (SDID)

The ID register (SDID) is a 32-bit register with a device specific ID.

SDID can be read from the JTAG interface when the IDCODE instruction is set, but cannot be accessed from the CPU.

For the read values, see **Table 46.5, ID Register Codes (RH850/F1KH-D8)**, **Table 46.6, ID Register Codes (RH850/F1KM-S4)**, **Table 46.7, ID Register Codes (RH850/F1KM-S1)** and **Table 46.2, Register Configuration**.

Table 46.5 ID Register Codes (RH850/F1KH-D8)

PKG	ID Register Codes			
	31 to 28	27 to 12	11 to 1	0
324 pins	0001	1000 0011 1100 1010	0100 0100 011	1
233 pins	0001	1000 0011 1100 1100	0100 0100 011	1
176 pins	0001	1000 0011 1100 1101	0100 0100 011	1

Table 46.6 ID Register Codes (RH850/F1KM-S4)

PKG	ID Register Codes			
	31 to 28	27 to 12	11 to 1	0
272 pins	0001	1000 0011 1000 1111	0100 0100 011	1
233 pins	0001	1000 0011 1001 0000	0100 0100 011	1
176 pins	0001	1000 0011 1001 0001	0100 0100 011	1
144 pins	0001	1000 0011 1001 0010	0100 0100 011	1
100 pins	0001	1000 0011 1001 0011	0100 0100 011	1

Table 46.7 ID Register Codes (RH850/F1KM-S1)

PKG	ID Register Codes			
	31 to 28	27 to 12	11 to 1	0
100 pins	0001	1000 0011 1001 0100	0100 0100 011	1
80 pins	0001	1000 0011 1001 0101	0100 0100 011	1
64 pins	0001	1000 0011 1001 0110	0100 0100 011	1
48 pins	0001	1000 0011 1001 0111	0100 0100 011	1

46.4.3 Bypass Register (SDBPR)

The bypass register (SDBPR) is a 1-bit register. When SDIR is set to BYPASS mode, SDBPR is connected to the position between DCUTDI and DCUTDO. The value after reset is undefined. SDBPR is not initialized by a power-on reset or by a low-level input of $\overline{\text{DCUTRST}}$.

46.4.4 Boundary Scan Register (SDBSR)

The boundary scan register (SDBSR) is a shift register for controlling the external I/O pins. When SDIR is set to SAMPLE/PRELOAD or EXTEST mode, SDBSR is connected to the position between DCUTDI and DCUTDO. The value after reset is undefined. SDBSR is not initialized by a power-on reset or a low-level input of $\overline{\text{DCUTRST}}$.

46.5 Operation

46.5.1 TAP Controller

Figure 46.2, TAP Controller State Transition Diagram shows the state transition of the TAP controller. The transition condition is the DCUTMS value at the rising edge of DCUTCK.

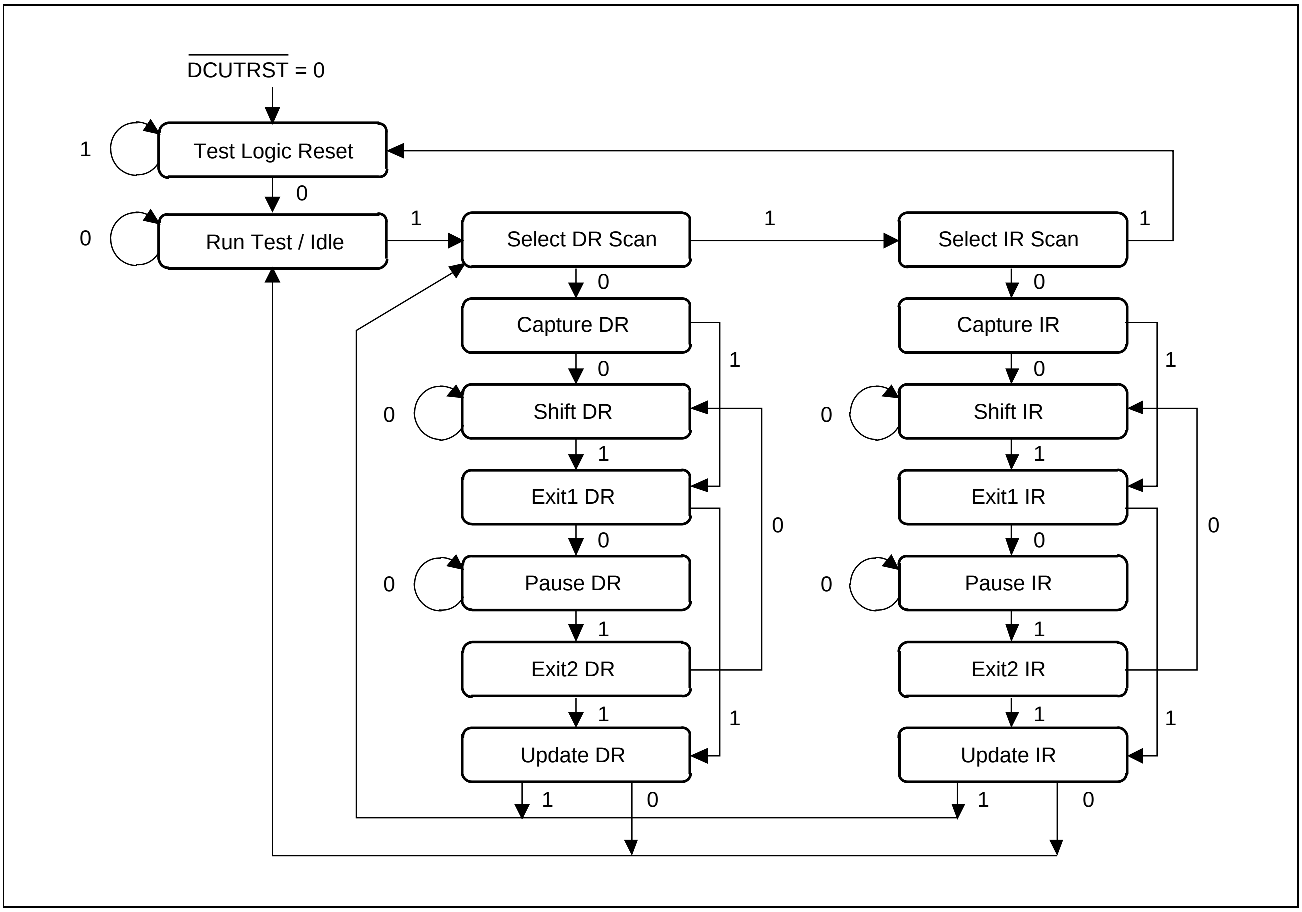


Figure 46.2 TAP Controller State Transition Diagram

NOTE

The DCUTDI value is sampled at the rising edge of DCUTCK and is shifted at the falling edge. DCUTDO is in the high-impedance state in the states other than Shift-DR and Shift-IR. A low-level input of $\overline{\text{DCUTRST}}$ causes transition to Test-Logic-Reset state asynchronously with DCUTCK.

46.5.2 Supported Instructions

46.5.2.1 BYPASS

The BYPASS instruction is a required standard instruction to operate the bypass register. This instruction shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this instruction, the test circuit has no effect on the system circuit.

46.5.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input the value to the boundary scan register from the internal circuits of this device; to output the value from the scan path; and to load data onto the scan path. During execution of this instruction, the level of the input pin of this device is sent to the internal circuits as is, and the value of the internal circuits is output to the outside via the output pin as is. Executing this instruction has no effect on the system circuit of this device.

The SAMPLE operation allows taking in the snapshots of the value to be transferred to the internal circuits from the input pin or the value to be transferred to the output pin from the internal circuits to the boundary scan register and allows reading the snapshots from the scan path. Snapshots can be taken in without preventing the normal operation of this device.

The PRELOAD operation allows setting the initial value to the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. If the EXTEST instruction is executed without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transfer to the output latch) because the parallel output latch value is always output to the output pin by the EXTEST instruction.

46.5.2.3 EXTEST

The EXTEST instruction is used to test the external circuits when this device is mounted on the printed-circuit board. When this instruction is executed, the output pin is used to output the test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed-circuit board; whereas the input pin is used to take in the test result from the printed-circuit board to the boundary scan register. When the EXTEST instruction is executed N times for testing, the test data for the Nth execution is scanned in at the (N - 1)th scan-out.

If the data is loaded onto the boundary scan register of the output pin in the Capture-DR state of this instruction, it is not used for testing the external circuits (replaced through shift operation).

46.5.2.4 IDCODE

The IDCODE instruction sets the JTAG interface pins to IDCODE mode, which is defined by the JTAG standard. When the JTAG interface is initialized (by a low-level input of $\overline{\text{DCUTRST}}$ or placing TAP in the Test-Logic-Reset state), IDCODE mode is set.

46.5.3 Pins Subject to Boundary Scan

All pins, excluding pins such as external clock input pins or power supply pins, are subject to boundary scan.

The pins which are not subjected to boundary scan are listed in **Table 46.8, Pins not Subject to Boundary Scan (RH850/F1KH-D8)** and **Table 46.9, Pins not Subject to Boundary Scan (RH850/F1KM-S4, RH850/F1KM-S1)**.

Table 46.8 Pins not Subject to Boundary Scan (RH850/F1KH-D8)

Type	Pin Name
JTAG interface	DCUTCK, DCUTDI, DCUTDO, DCUTMS, $\overline{\text{DCUTRST}}$
Power supply pins	REG0VCC, REG1VCC, AWOVCL, AWOVSS, ISOVCL, ISOVSS EVCC, EVSS, BVCC, BVSS
Power supply pins (A/D converter)	A0VREF, A1VREF, A0VSS, A1VSS
Clock	X1, X2, XT1, IP0_0/XT2
Mode setting	P10_8/FLMD1, P10_1/MODE0, P10_2/MODE1

Table 46.9 Pins not Subject to Boundary Scan (RH850/F1KM-S4, RH850/F1KM-S1)

Type	Pin Name
JTAG interface	DCUTCK, DCUTDI, DCUTDO, DCUTMS, $\overline{\text{DCUTRST}}$
Power supply pins	REGVCC, AWOVCL, AWOVSS, ISOVCL, ISOVSS EVCC, EVSS, BVCC[*1], BVSS[*1]
Power supply pins (A/D converter)	A0VREF, A1VREF[*1], A0VSS, A1VSS[*1]
Clock	X1, X2, XT1[*1], IP0_0/XT2[*1]
Mode setting	P10_8/FLMD1, P10_1/MODE0, P10_2/MODE1

Note 1. Only available for 272/233/176/144-pin versions.

The following signals are only sampled in boundary scan mode.

Table 46.10 Pins Subject to Boundary Scan (Sampling Only)

Function	Pin Name
Reset	$\overline{\text{RESET}}$
Mode setting	FLMD0

The following pins are shared by the analog buffer. Accordingly, boundary scan only applies to general I/O pins.

Table 46.11 Pins Subject to Boundary Scan (Only General I/O Pins)

Function	Pin Name
ADCA0 input	P8_0-12, P9_0-4, AP0_0-15
ADCA1 input	P18_0-15, P19_0-3, AP1_0-15

NOTE

In boundary scan mode, the level of the following pins must be fixed:
P10_1 = Low, P10_2 = High, and P10_8 = High

46.6 Usage Notes

1. Once an instruction is set, it is not modified until another instruction is issued. To issue the same instruction twice in a row, insert an instruction that has no effect on chip operation (such as BYPASS) between the instructions.
2. To start the system in boundary scan mode, de-assert $\overline{\text{DCUTRST}}$ while $\overline{\text{RESET}}$ is high. Also be sure to set DCUTMS to high before de-asserting $\overline{\text{DCUTRST}}$ and ensure that DCUTMS remains high for 600 ns + five DCUTCK clock cycles after de-asserting $\overline{\text{DCUTRST}}$.
3. For the maximum clock frequency that can be input to DCUTCK, see **Section 47A, Electrical Characteristics of RH850/F1KH-D8**, **Section 47B, Electrical Characteristics of RH850/F1KM-S4** and **Section 47C, Electrical Characteristics of RH850/F1KM-S1**.
4. If serial transfer is performed exceeding the number of bits of the register connected between DCUTDI and DCUTDO, the data that is input from DCUTDI is output from DCUTDO as is.
5. If the serial transfer sequence is corrupted, be sure to assert $\overline{\text{DCUTRST}}$. In this case, transfer starts again from the beginning regardless of the point of transfer corruption.
6. Data is output via DCUTDO at the falling edge of DCUTCK.
7. To facilitate debugging, route $\overline{\text{DCUTRST}}$ on the board in such a way that patterns can be easily cut.

Section 47A Electrical Characteristics of RH850/F1KH-D8

47A.1 Overview

The electrical spec of this device is guaranteed by the following operational condition. But, this condition is different depends on each characteristics, so refer to each chapter for more detail.

47A.1.1 Pin Groups

47A.1.1.1 324-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR0	REG0VCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P3, P8, P9, P20, P23 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P13, P18, P19, P21, P22, P24
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47A.1.1.2 Reserved

47A.1.1.3 233-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR0	REG0VCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P3, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P13, P18, P19
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47A.1.1.4 176-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR0	REG0VCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P18
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47A.1.2 General Measurement Conditions

47A.1.2.1 Common Conditions

- Power supply
  - REG0VCC = EVCC = VPOC*[1] to 5.5 V
  - REG1VCC = VPOC*[1] to 3.6 V, REG1VCC ≤ REG0VCC
  - BVCC = VPOC*[1] to REG0VCC
  - A0VREF = 3.0 V to 5.5 V
  - A1VREF = 3.0 V to 5.5 V
  - AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V
- Capacitance of the internal regulator
  - CAWOVCL: 0.1 µF ±30%
  - CISOVCL: 0.1 µF ±30% per pin
- Operating temperature
  - Tj = –40 to +130°C @R7F7017xx3ABG*[2]
  - Tj = –40 to +150°C @R7F7017xx4ABG*[2]
    @R7F7017yy3AFP*[2]

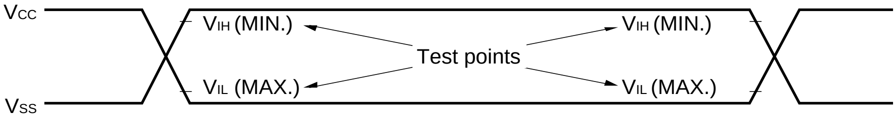
  xx = 10, 11, 14, 15
  yy = 08, 09
- Load conditions
  - CL = 30 pF

Note 1. “VPOC” means POC (power-on clear) detection voltage. For more detail, see **Section 47A.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

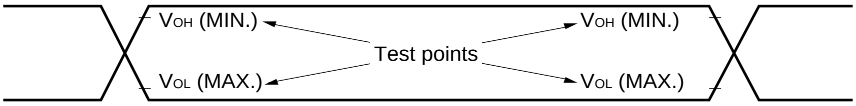
Note 2. Regarding operation temperature of each product, see **Section 1A.3, RH850/F1KH Product Lineup**.

47A.1.2.2 AC Characteristic Measurement Condition

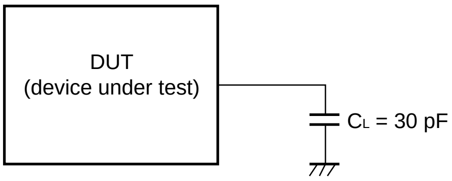
(1) AC Test Input Measurement Points



(2) AC Test Output Measurement Points



(3) Load Conditions



CAUTION

If the load capacitance exceeds 30pF due to the circuit configuration, it is recommended to insert a buffer in order to reduce capacitance to less than 30pF.

47A.2 Absolute Maximum Ratings

CAUTIONS

1. Do not directly connect outputs (or input/outputs) to each other, power supply and ground.
2. Even momentarily exceeding the absolute maximum rating for just one item creates a threat of failure in the reliability of the products. That is, the absolute maximum ratings are the levels that raise a threat of physical damage to the products. Be sure to use the products only under conditions that do not exceed the ratings. The quality and normal operation of the product are guaranteed under the standards and conditions given as DC and AC characteristics.
3. When designing an external circuit ensure that the connections don’t conflict with the port state of this device.

47A.2.1 Supply Voltages

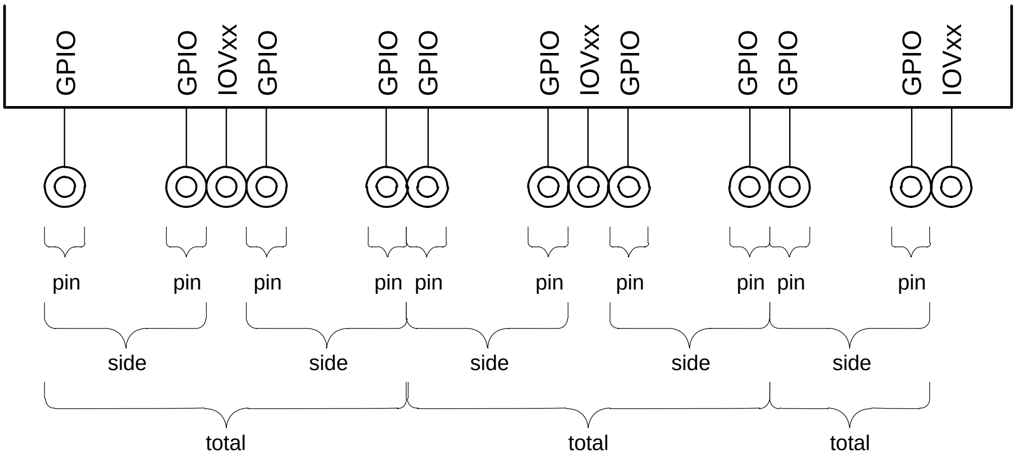
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
System supply voltage	REG0VCC		–0.5		6.5	V
	REG1VCC		–0.5		6.5	V
	AWOVSS		–0.5		0.5	V
	ISOVSS		–0.5		0.5	V
Port supply voltage	EVCC		–0.5		6.5	V
	BVCC		–0.5		6.5	V
	EVSS		–0.5		0.5	V
	BVSS		–0.5		0.5	V
A/D-converter supply voltage	A0VREF		–0.5		6.5	V
	A1VREF		–0.5		6.5	V
	A0VSS		–0.5		0.5	V
	A1VSS		–0.5		0.5	V

47A.2.2 Port Voltages

Item	Pin Group*[1]	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	PgR0	VI		–0.5		REG0VCC + 0.5 (Do not exceed 6.5 V)	V
	PgE			–0.5		EVCC + 0.5 (Do not exceed 6.5 V)	V
	PgB			–0.5		BVCC + 0.5 (Do not exceed 6.5 V)	V
	PgA0			–0.5		A0VREF + 0.5 (Do not exceed 6.5 V)	V
	PgA1			–0.5		A1VREF + 0.5 (Do not exceed 6.5 V)	V

Note 1. The characteristics of the alternative-function pins are the same as those of the port pins unless otherwise specified.

47A.2.3 Port Current



Definition of the condition:

- Per pin: Output current of one GPIO
- Per side: Total output current of all GPIO pins on one side of one IOVxx
- Total: Total output current of both sides of one IOVxx

Note:

- GPIO: General-purpose I/O pin (JP0, P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, P23, P24, AP0, AP1)
- IOVxx: Power supply pin for I/O pins (EVCC/EVSS, BVCC/BVSS, A0VREF/A0VSS, A1VREF/A1VSS)

47A.2.3.1 324-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_10 to P20_15)			–48	mA
			Per side (total of P20_0 to P20_9)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Per side (total of P3_1 to P3_12)			–48	mA
			Per side (total of P23_0 to P23_10)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P18_0 to P18_7)			–48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			–48	mA
			Per side (total of P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			–48	mA
			Per side (total of P21_0, P21_2 to P21_14, P22_0 to P22_2)			–48	mA
			Per side (total of P22_3 to P22_8)			–48	mA
			Per side (total of P21_1, P22_9 to P22_15)			–48	mA
			Per side (total of P24_0 to P24_7)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA

(324-pin version)

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_10 to P20_15)			48	mA
			Per side (total of P20_0 to P20_9)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Per side (total of P3_3 to P3_12)			48	mA
			Per side (total of P23_0 to P23_10)			48	mA
			Total (EVCC)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			48	mA
			Per side (total of P21_0, P21_2 to P21_14 , P22_0 to P22_6)			48	mA
			Per side (total of P21_1, P22_7 to P22_15)			48	mA
			Per side (total of P24_0 to P24_7)			48	mA
			Total (BVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VREF)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VREF)			48	mA

47A.2.3.2 Reserved

47A.2.3.3 233-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P18_0 to P18_7)			–48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			–48	mA
			Per side (total of P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA

(233-pin version)

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Total (EVCC)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			48	mA
			Total (BVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VREF)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VREF)			48	mA

47A.2.3.4 176-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P10_6 to P10_9, P18_0 to P18_7)			–48	mA
			Per side (total of P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Total (EVSS)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5)			48	mA
			Total (BVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VSS)			48	mA

47A.2.4 Temperature Condition

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Storage temperature	Tstg		–55		150	°C
Junction temperature	Tj	R7F7017xx3ABG	–40		130	°C
		R7F7017xx4ABG R7F7017yy3AFP	–40		150	°C

Note: xx = 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 15
yy = 00, 01, 08, 09

Regarding operation temperature of each product, see **Section 1A.3, RH850/F1KH Product Lineup**.

47A.3 Operational Condition

47A.3.1 Recommended Operating Conditions

Products of CPU frequency 240 MHz max.

(1/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CPU clock frequency	f_{CPUCLK_H}	CKDIVMD = 1			240	MHz
		CKDIVMD = 0			120	MHz
	f_{CPUCLK_M}				120	MHz
	f_{CPUCLK_L}	for OSTMn			60	MHz
		for MEMC[*5]				
	f_{CPUCLK_UL}				30	MHz
Peripheral clock (clock domain) frequency[*1]	f_{CKSCLK_AWDTA}	for WDTA0			240[*2]	kHz
	f_{CKSCLK_ATAUJ}	for TAUJ0			40	MHz
		for TAUJ2				
	f_{CKSCLK_ARTCA}	for RTCA0			4	MHz
	f_{CKSCLK_AADCA}	for ADCA0			40	MHz
	f_{CKSCLK_AFOUT}	for FOUT			24	MHz
	$f_{CKSCLK\_ICPUCLK}$	for CPU subsystem			240 / 120	MHz
	f_{CKSCLK_IPERI1}	for TAUD0			80	MHz
		for TAUJ1				
		for TAUJ3				
		for ENCA0				
		for TAPA0				
		for PIC0				
		for SFMA0				
	f_{CKSCLK_IPERI2}	for TAUBn			40	MHz
		for RCFDCn (clkc)				
		for RSENTn				
		for PWBAn				
		for PWGAn				
		for PWSAn				
		for MMCA0				
	f_{CKSCLK_ILIN}	for RLIN24n			40	MHz
		for RLIN3n				
	f_{CKSCLK_IADCA}	for ADCA1			40	MHz
	f_{CKSCLK_ICAN}	for RCFDCn (PCLK)			80	MHz
	$f_{CKSCLK\_ICANOSC}$	for RCFDCn (clk_xincan)			24	MHz
	f_{CKSCLK_ICSI}	for CSIGn			80	MHz
		for CSIHn				
	f_{CKSCLK_IIIC}	for RIICn			40	MHz
	$f_{LS\ IntOSC}$	for WDTA1			240[*2]	kHz
		for WDTA2				
	f_{EMCLK}	for LPSn			8	MHz

(2/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Power supply	REG0VCC	REG0VCC = EVCC	VPOC*[3]		5.5	V
	EVCC					
	REG1VCC	REG1VCC ≤ REG0VCC	VPOC*[3]		3.6	V
	BVCC		VPOC*[3]		REG0VCC	V
	A0VREF		3.0		5.5	V
	A1VREF					
Normal operation voltage	AWOVCL		1.1	1.25	1.35	V
	ISOVCL					
Limited operation voltage*[4]	AWOVCL		1.35		1.43	V
	ISOVCL					

Note 1. For clock specification of peripherals, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**.

Note 2. This frequency depends on the internal oscillator (LS IntOSC).

Note 3. “VPOC” means POC (power-on clear) detection voltage (TYP. 2.85 V). For detail, see **Section 47A.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.
In addition, the guaranteed operation in DC characteristic.
And AC characteristic is guaranteed when more than 3.0 V.
When the power supply voltage is VPOC to 3.0 V, the device does not malfunction.

Note 4. Reliability restrictions from 1.35 V to 1.43 V.

Note 5. Devided by 2 on MEMC internal.

Products of CPU frequency 160 MHz max.

(1/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CPU clock frequency	f_{CPUCLK_H}				160	MHz
	f_{CPUCLK_M}				80	MHz
	f_{CPUCLK_L}	for OSTMn			40	MHz
		for MEMC*5				
	f_{CPUCLK_UL}				20	MHz
Peripheral clock (clock domain) frequency*1	f_{CKSCLK_AWDTA}	for WDTA0			240*2	kHz
	f_{CKSCLK_ATAUJ}	for TAUJ0			40	MHz
		for TAUJ2				
	f_{CKSCLK_ARTCA}	for RTCA0			4	MHz
	f_{CKSCLK_AADCA}	for ADCA0			40	MHz
	f_{CKSCLK_AFOUT}	for FOUT			24	MHz
	$f_{CKSCLK\_ICPUCLK}$	for CPU subsystem			160	MHz
	f_{CKSCLK_IPERI1}	for TAUD0			80	MHz
		for TAUJ1				
		for TAUJ3				
		for ENCA0				
		for TAPA0				
		for PIC0				
		for SFMA0				
	f_{CKSCLK_IPERI2}	for TAUBn			40	MHz
		for RCFDCn (clkc)				
		for RSENTn				
		for PWBAn				
		for PWGAn				
		for PWSAn				
		for MMCA0				
	f_{CKSCLK_ILIN}	for RLIN24n			40	MHz
		for RLIN3n				
	f_{CKSCLK_IADCA}	for ADCA1			40	MHz
	f_{CKSCLK_ICAN}	for RCFDCn (pclk)			80	MHz
	$f_{CKSCLK\_ICANOSC}$	for RCFDCn (clk_xincan)			24	MHz
	f_{CKSCLK_ICSI}	for CSIGn			80	MHz
		for CSIHn				
	f_{CKSCLK_IIIC}	for RIICn			40	MHz
	$f_{LS\ IntOSC}$	for WDTA1			240*2	kHz
		for WDTA2				
	f_{EMCLK}	for LPSn			8	MHz
Power supply	REG0VCC	REG0VCC = EVCC	VPOC*3		5.5	V
	EVCC					
	REG1VCC	REG1VCC ≤ REG0VCC	VPOC*3		3.6	V
	BVCC		VPOC*3		REG0VCC	V
	A0VREF		3.0		5.5	V
	A1VREF					

(2/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Normal operation voltage	AWOVCL ISOVCL		1.1	1.25	1.35	V
Limited operation voltage*4	AWOVCL		1.35		1.43	V

Note 1. For clock specification of peripherals, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**.

Note 2. This frequency depends on the internal oscillator (LS IntOSC).

Note 3. “VPOC” means POC (power-on clear) detection voltage (TYP. 2.85 V). For detail, see **Section 47A.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.
In addition, the guaranteed operation in DC characteristic.
And AC characteristic is guaranteed when more than 3.0 V.
When the power supply voltage is VPOC to 3.0 V, the device does not malfunction.

Note 4. Reliability restrictions from 1.35 V to 1.43 V.

Note 5. Devided by 2 on MEMC internal.

47A.3.2 Oscillator Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C

(1) MainOSC (In Case of Using a Crystal/Ceramic)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MainOSC frequency*[3]	f_{MOSC}		8		24	MHz
MainOSC current consumption	I_{MOSC}	After stabilization		1.9*[2]	2.3*[2]	mA
MainOSC oscillation start point	V_{MOSCSP}		VPOC			V
MainOSC oscillation operating point	V_{MOSCOP}			0.5 × REG0VCC *[2]		V
MainOSC oscillation amplitude	$V_{MOSCAMP}$		0.4 × REG0VCC – 0.2*[2]			V
MainOSC oscillation stabilization time	t_{MSTB}			2*[1],*[2]		ms
MainOSC transconductance	g_{m_MOSC}	MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 00		11.1*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 01		10.6*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 10		9.3*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 11		7.8*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 00		8.6*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 01		7.8*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 10		6.1*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 11		4.0*[1]		mA/V

Note 1. Oscillator stabilization time is time until being set (“1”) in MOSCS.MOSCCLKACT bit after MOSCE.MOSCENTRG bit is written “1”, and depends on the setting value of MOSCST register. Please decide appropriate oscillation stabilization time by matching test with resonator and oscillation circuit.

Note 2. This is reference value.

Note 3. The following four crystal/ceramic resonator frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

(2) MainOSC (In Case of External Clock Input to X1)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
X1 clock Input frequency*[1]	f_{EX}		8		24	MHz
X1 clock Input cycle time	t_{EXCYC}		41.7		125	ns
X1 High level Input voltage	V_{IH}		0.7 × REG0VCC		REG0VCC + 0.5	V
		@Flash Programing Interface*[2]	0.8 × REG0VCC		REG0VCC + 0.5	V
X1 Low level Input voltage	V_{IL}		–0.5		0.3 × REG0VCC	V
		@Flash Programing Interface*[2]	–0.5		0.2 × REG0VCC	V
X1 Input leakage current	I_{LIH}	VI = REG0VCC			0.5	μA
	I_{LIL}	VI = 0 V			–0.5	μA
X1 clock Input low-level pulse width	t_{EXL}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input high-level pulse width	t_{EXH}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input period jitter			–0.3		0.3	ns

Note 1. The following four external clock input frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

Note 2. X2 should be open and its parasitic capacitance should be less than 5 pF.

(3) SubOSC

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SubOSC frequency	f_{SOSC}	Crystal	30	32.768	38	kHz
SubOSC current consumption	I_{SOSC}	After stabilization		1.5*[2]	4*[2]	μA
SubOSC DC operating point	$V_{SOSCDCOP}$			0.65*[2]		V
SubOSC oscillation stabilization time	t_{SSTB}			*[1]		s

Note 1. Oscillator stabilization time is time until being set (“1”) in SOSCS.SOSCCLKACT bit after SOSCE.SOSCENTRG bit is written “1”, and depends on the setting value of SOSCST register. Please decide appropriate oscillation stabilization time by matching test with resonator and oscillation circuit.

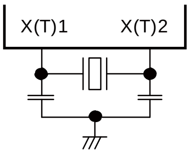
Note 2. This is reference value.

CAUTION

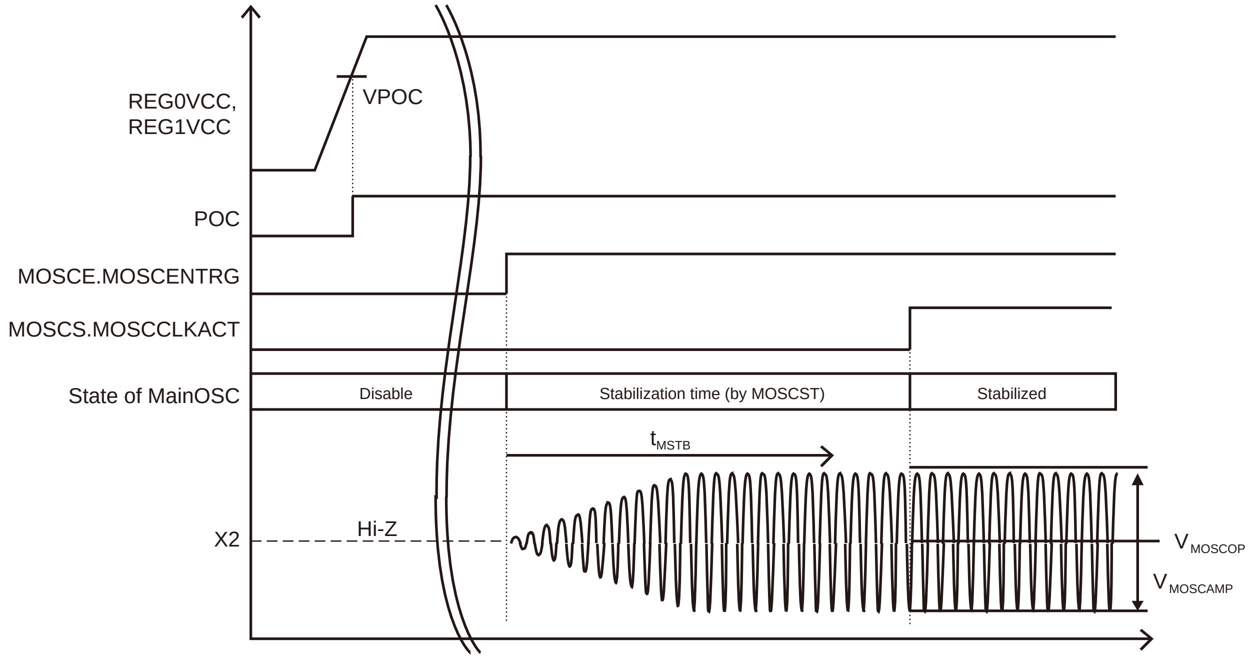
The oscillation stabilization time differs according the matching with the external resonator circuit. It is recommended to determine the oscillation stabilization time by an oscillator matching test.

NOTE

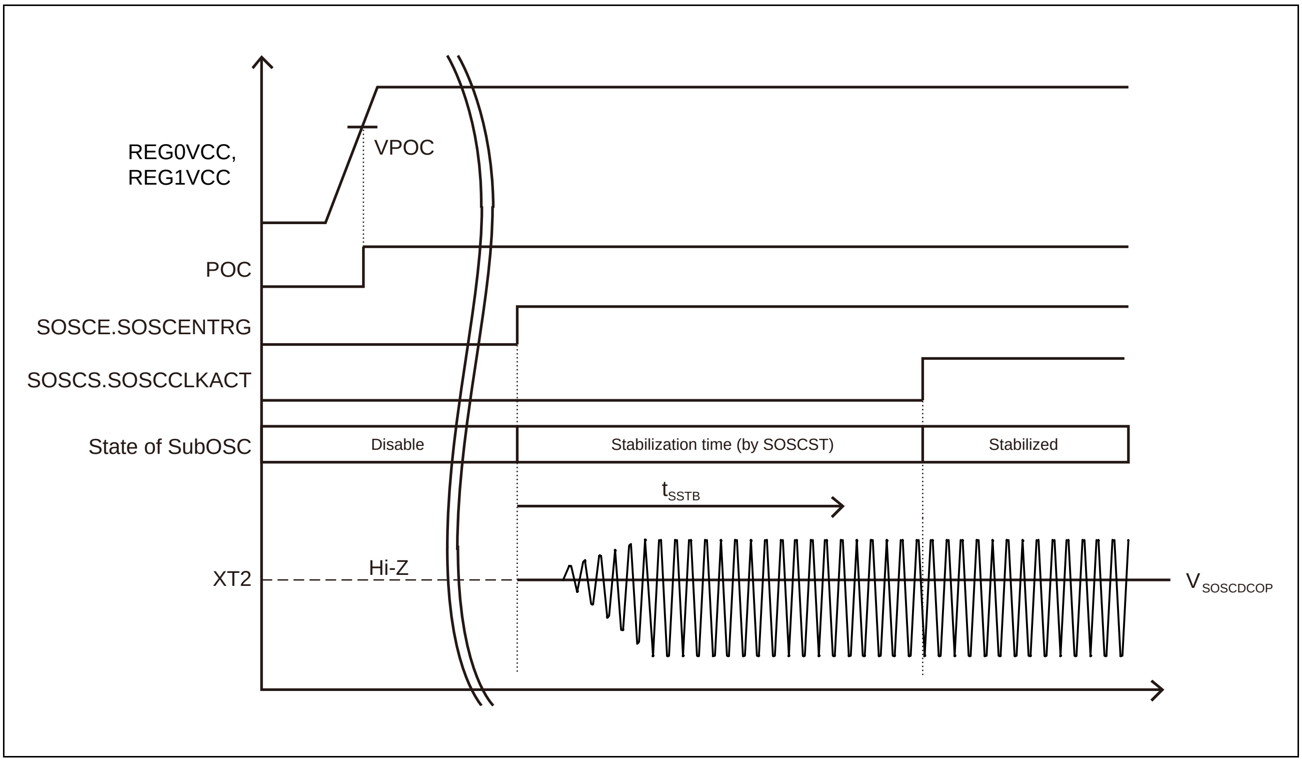
Recommended oscillator circuit is shown below.



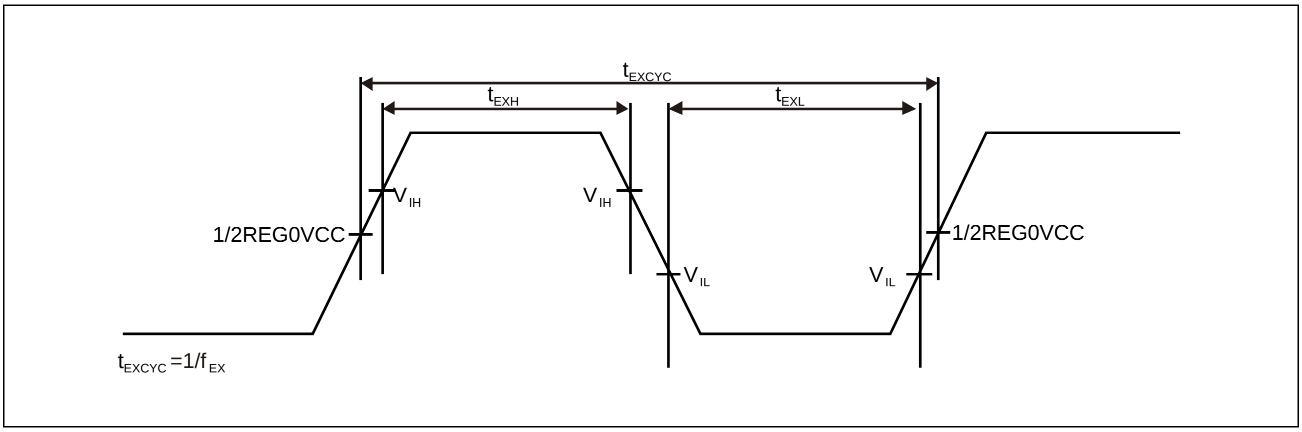
MainOSC



SubOSC



External clock



47A.3.3 Internal Oscillator Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LS IntOSC frequency	f_{RL}		220.8	240	259.2	kHz
HS IntOSC frequency*[3]	f_{RH}		7.6	8	8.4	MHz
		After user trimming @ trimming temp*[2]	7.92	8	8.08	MHz
HS IntOSC current consumption	I_{RH}	After stabilization			170*[1]	µA
HS IntOSC oscillation stabilization time	t_{RHSTB}				54.4	µs

Note 1. This is reference value.

Note 2. The HS IntOSC frequency may not meet the specification range (8.00 MHz ±0.08 MHz after user trimming @ trimming temp) in the while writing/erasing the code/data flash.

Note 3. The HS IntOSC frequency may not meet the specification range in the Cyclic STOP/Cyclic RUN mode.

47A.3.4 PLL Characteristics

47A.3.4.1 PLL0 (for CPU, with SSCG) Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Input frequency	$f_{\mathrm{PLL0CLKIN}}$	MainOSC			8		24	MHz
		HS IntOSC*3			7.6	8.0	8.4	MHz
		HS IntOSC After user trimming @ trimming temp*3			7.92	8.0	8.08	MHz
Output frequency	f_{CPLL0OUT}	SSCG mode	MainOSC	Products of CPU frequency 240 MHz max.	105.8		240	MHz
				Products of CPU frequency 160 MHz max.	105.8		160	MHz
			HS IntOSC*3		67		84	MHz
			HS IntOSC After user trimming @ trimming temp*3		69.8		80.8	MHz
Modulation frequency	f_{MOD}				20		100	kHz
Frequency dithering range*2	f_{DIT}				0.82	1.0	1.18	%
					1.64	2.0	2.36	%
					2.46	3.0	3.54	%
					3.28	4.0	4.72	%
					4.10	5.0	5.90	%
					4.92	6.0	7.08	%
					6.56	8.0	9.44	%
					8.20	10.0	11.80	%
Lock time*1	t_{LCK0}	SSCG mode	PLL0ST = 0000 $1B80_{\mathrm{H}}$		814.9	880	956.6	µs

Note 1. Lock time is time until being set (“1”) in PLL0S.PLL0CLKACT bit after PLL0E.PLL0ENTRG bit is written “1”.

Note 2. "Frequency dithering range” is set by PLL0ADJ[2:0] bits of PLL0C registers.

Note 3. The HS IntOSC has a frequency deviation. When the HSIntOSC is used the frequency deviation should be considered for the customer application as it affects peripheral functions (e.g. TAUx, ADCAn, etc.).

47A.3.4.2 PLL1 (for CPU/Peripheral) Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input frequency	$f_{PLL1CLKIN}$	MainOSC	8		24	MHz
		HS IntOSC*[3]	7.6	8.0	8.4	MHz
		HS IntOSC After user trimming @ trimming temp*[3]	7.92	8.0	8.08	MHz
Output frequency	$f_{CPLL1OUT}$	MainOSC	80		120	MHz
		HS IntOSC*[3]	76	80	84	MHz
	$f_{PPLLOUT}$		76	80	84	MHz
Output period jitter*[1]	t_{CPJ1}		–100		100	ps
Long term jitter*[1]	t_{LTJ}	term = 1 μs	–500		500	ps
		term = 10 μs	–1		1	ns
		term = 20 μs	–2		2	ns
Lock time*[2]	t_{LCK1}		104	112.3	122.1	μs

Note 1. This is reference value.

Note 2. Lock time is time until being set (“1”) in PLL1S.PLL1CLKACT bit after PLL1E.PLL1ENTRG bit is written “1”.

Note 3. The HS IntOSC has a frequency deviation. When the HSIntOSC is used the frequency deviation should be considered for the customer application as it affects peripheral functions (e.g. TAUx, ADCAn, etc.).

47A.4 DC Characteristics

47A.4.1 Capacitance

Condition: REG0VCC = REG1VCC = EVCC = BVCC = A0VREF = A1VREF = AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, Ta = 25°C

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input capacitance	CI*[1]	f = 1 MHz 0 V for non measurement pins			10	pF
Input/output capacitance	CIO*[2]				10	pF

Note 1. CI: Capacitance between the input pin and ground

Note 2. CIO: Capacitance between the input/output pin and ground

47A.4.2 Pin Characteristics

Condition: Some of the conditions mentioned in this chapter can be selected by software and described in the hardware user's manual.

(1/6)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
$\overline{\text{RESET}}$	—	—	✓	—	—	—	—	—	—
FLMD0	—	✓	—	—	—	—	—	✓	✓
AP0_0	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_1	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_2	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_3	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_4	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_5	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_6	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_7	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_8	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_9	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_10	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_11	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_12	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_13	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_14	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_15	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_0	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_1	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_2	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_3	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_4	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_5	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_6	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_7	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_8	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_9	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_10	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_11	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_12	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_13	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_14	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_15	✓	—	—	—	—	✓	Slow	—	✓*1
IP0_0	—	—	—	—	—	—	—	—	—
JP0_0	—	✓	—	✓	✓	—	Slow	✓	✓
JP0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
JP0_2	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_3	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_4	—	—	—	✓	—*5	—	Slow	✓	✓
JP0_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(2/6)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
JP0_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_2	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_3	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_5	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_5	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P1_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_0	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P10_1	—	✓	—	✓	✓	—	Slow/Fast*3	✓	✓
P10_2	—	✓	—	✓	✓	—	Slow/Fast*3	✓	✓
P10_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_4	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P10_5	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P10_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(3/6)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P10_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_2	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_3	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_7	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_10	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P11_11	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P11_12	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P11_15	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P12_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_4	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P12_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_2	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_3	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_4	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P13_5	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P18_0	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_1	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_2	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_3	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_4	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_5	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_6	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_7	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_8	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_9	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4

(4/6)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P18_10	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_11	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_12	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_13	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_14	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_15	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_0	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_1	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_2	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_3	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P2_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_4	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P2_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_0	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P21_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(5/6)

	Port Input Buffer Function						Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P21_2	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P21_3	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P21_4	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P21_5	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P21_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_7	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P21_8	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P21_9	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P21_10	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P21_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_0	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P22_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_6	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P22_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P22_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_1	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P23_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_3	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P23_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P23_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(6/6)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P24_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P24_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P8_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_4	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_5	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_6	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_7	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_8	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_9	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_10	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_11	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_12	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_4	—	✓	—	✓	—	✓	Slow	✓	✓*4

Note 1. Pull-down resistor for ADC diagnostic purpose. Control via ADC self-diagnostic register.

Note 2. Supports Cload: 100 pF

Note 3. Supports Cload: 50 pF

Note 4. Pull-down resistors for ADC diagnostic and internal pull-down purposes. For ADC diagnostic, control via ADC self-diagnostic register. For internal pull-down, control via PD register.

Note 5. TTL is selected for Boundary scan mode or Nexus in normal operating mode.

Note 6. Only available for 324-pin devices.

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

(1/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High level input voltage	VIH	CMOS		0.65 × IOVCC		IOVCC + 0.3	V
		SHMT1[*3]		0.65 × IOVCC		IOVCC + 0.3	V
		SHMT2		0.75 × IOVCC		IOVCC + 0.3	V
		SHMT4		0.8 × IOVCC		IOVCC + 0.3	V
		TTL	IOVCC = VPOC to 3.6 V	2.0		IOVCC + 0.3	V
			IOVCC = 3.6 V to 5.5 V	2.2		IOVCC + 0.3	V
		IP0_0 pin		0.7 × REG0VCC		REG0VCC	V
Low level input voltage	VIL	CMOS		–0.3		0.35 × IOVCC	V
		SHMT1		–0.3		0.35 × IOVCC	V
		SHMT2		–0.3		0.25 × IOVCC	V
		SHMT4		–0.3		0.5 × IOVCC	V
		TTL		–0.3		0.8	V
		IP0_0 pin		0		0.3 × REG0VCC	V
Input hysteresis for Schmitt	VH	SHMT1		0.3			V
		SHMT2		0.2 × IOVCC			V
		SHMT4		0.1			V
Input leakage current	ILIH	IP0_0 pin, VI = REG0VCC				0.5	µA
		$\overline{\text{RESET}}$, FLMD0, JP0, P0, P1, P2, P3, P8, P9, P20, P23 pin, VI = EVCC[*2]				0.5	µA
		P10, P11, P12, P13, P18, P19, P21, P22, P24 pin, VI = BVCC[*2]				0.5	µA
		AP0 pin, VI = A0VREF[*2], Tj ≤ 130°C				0.3	µA
		AP0 pin, VI = A0VREF[*2]				0.5	µA
		AP1 pin, VI = A1VREF[*2], Tj ≤ 130°C				0.3	µA
		AP1 pin, VI = A1VREF[*2]				0.5	µA
	ILIL	IP0_0 pin, VI = 0 V				–0.5	µA
		$\overline{\text{RESET}}$, FLMD0, JP0, P0, P1, P2, P3, P8, P9, P20 P23 pin, VI = 0 V[*2]				–0.5	µA
		P10, P11, P12, P13, P18, P19, P21, P22 P24 pin, VI = 0V[*2]				–0.5	µA
		AP0 pin, VI = 0 V[*2], Tj ≤ 130°C				–0.3	µA
		AP0 pin, VI = 0 V[*2]				–0.5	µA
		AP1 pin, VI = 0 V[*2], Tj ≤ 130°C				–0.3	µA
		AP1 pin, VI = 0 V[*2]				–0.5	µA
Internal pull-up resistance	RU	except FLMD0 pin, VI = 0 V		20 (275 µA)	40	100	kΩ
		FLMD0 pin, VI = 0V[*3]		4 (1375 µA)		36	kΩ
Internal pull-down resistance	RD	except FLMD0 pin, VI = IOVCC		20 (275 µA)	40	100	kΩ
		FLMD0 pin, VI = EVCC		4 (1375 µA)		36	kΩ

(2/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High level output voltage	VOH	Fast mode	IOH = –5 mA (6 pins)*4	IOVCC – 1.0			V
			IOH = –3 mA (10 pins)*4	IOVCC – 1.0			V
			IOH = –1 mA (16 pins)*4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins)*4	IOVCC – 0.5			V
		Slow mode	IOH = –1 mA (16 pins)*4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins)*4	IOVCC – 0.5			V
Low level output voltage	VOL	Fast mode	IOL = 5 mA (6 pins)*4			0.4	V
			IOL = 3 mA (10 pins)*4			0.4	V
			IOL = 1 mA (16 pins)*4			0.4	V
		Slow mode	IOL = 1 mA (16 pins)*4			0.4	V
Rise/Fall time	t_{KRP}/t_{KFP}	Fast mode (except below pins)*5	CL = 30 pF			7	ns
			CL = 50 pF			12	ns
			CL = 100 pF			24	ns
		Fast mode (P0_5, P0_6, P1_5, P2_4, P10_1, P10_2, P11_2, P11_3, P11_6, P11_7, P23_1, P23_3)*6	CL = 50 pF			6	ns
		Fast mode (P0_2, P0_3)*6	CL = 100 pF			6.15	ns
		Slow mode*5	CL = 30 pF			37	ns
			CL = 50 pF			62	ns
			CL = 100 pF			124	ns
Output frequency	f_O	Fast mode	CL = 30 pF			40	MHz
		Slow mode	CL = 30 pF			10	MHz
			CL = 50 pF			6	MHz
			CL = 100 pF			3	MHz

Note 1. “IOVCC” means the pins are assigned to the power supply (EVCC, BVCC, A0VREF and A1VREF).

Note 2. Not select the analog input function of ADCn.

Note 3. When the internal pull-up resistor of FLMD0 pin is applied by FLMDCNT register, please connect 86 kΩ or more as external pull-down resistor.

Note 4. The number of pin indicates simultaneous ON.

Note 5. Measurement point: 0.1 × IOVCC to 0.9 × IOVCC

Note 6. Measurement point: 0.2 × IOVCC to 0.8 × IOVCC

47A.4.2.1 Output Current

(1) 324-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_10 to P20_15			–30	mA
				P20_0 to P20_9			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
				P3_1 to P3_12			–30	mA
				P23_0 to P23_10			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P18_0 to P18_7			–30	mA
				P18_8 to P18_15, P19_0 to P19_3			–30	mA
				P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			–30	mA
				P21_0, P21_2 to P21_14, P22_0 to P22_2			–30	mA
				P22_3 to P22_8			–30	mA
				P21_1, P22_9 to P22_15			–30	mA
				P24_0 to P24_7			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA

(324-pin version)

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_10 to P20_15			30	mA
				P20_0 to P20_9			30	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
				P3_1 to P3_12			30	mA
				P23_0 to P23_10			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P18_8 to P18_15, P19_0 to P19_3			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			30	mA
				P21_0, P21_2 to P21_14, P22_0 to P22_6			30	mA
				P21_1, P22_7 to P22_15			30	mA
				P24_0 to P24_7			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47A.2.3, Port Current**.

(2) Reserved

(3) 233-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P18_0 to P18_7			–30	mA
				P18_8 to P18_15, P19_0 to P19_3			–30	mA
				P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			30	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15 P3_0, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P18_8 to P18_15, P19_0 to P19_3			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47A.2.3, Port Current**.

(4) 176-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P10_6 to P10_9, P18_0 to P18_7			–30	mA
				P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			11	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47A.2.3, Port Current**.

47A.4.3 Power Supply Currents

Condition: REG0VCC, REG1VCC, EVCC, BVCC, A0VREF and A1VREF total current. But the I/O buffer is stopped.

Products of CPU frequency 240 MHz max.

Item	Symbol	Condition					MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2	Power supply				
RUN mode current	IDDR	Run (240 MHz)	Run	–40 to 150°C	Run (#1)	Total		113	330	mA
						REG1VCC		92	290	mA
				25°C	Stop (#1)	Total		107		mA
						REG1VCC		88		mA
RUN mode current (During data/code flash programming)	IDDR3	Run (240 MHz)	Run	–40 to 150°C	Run (#2)	Total		133	350	mA
						REG1VCC		92	290	mA
RUN mode current (With code flash background operation)	IDDRBGO	Run (240 MHz)	Run	–40 to 150°C	Run (#6)	Total		133	350	mA
						REG1VCC		92	290	mA
RUN mode current (HALT state)	IDDH	Run (240 MHz)	Run	–40 to 150°C	Run (#3)	Total		108	325	mA
						REG1VCC		88	286	mA

Products of CPU frequency 160 MHz max.

Item	Symbol	Condition					MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2	Power supply				
RUN mode current	IDDR	Run (160 MHz)	Run	–40 to 150°C	Run (#1)	Total		83	293	mA
						REG1VCC		62	253	mA
				25°C	Stop (#1)	Total		77		mA
						REG1VCC		58		mA
RUN mode current (During data/code flash programming)	IDDR3	Run (160 MHz)	Run	–40 to 150°C	Run (#2)	Total		103	313	mA
						REG1VCC		62	253	mA
RUN mode current (With code flash background operation)	IDDRBGO	Run (160 MHz)	Run	–40 to 150°C	Run (#6)	Total		103	313	mA
						REG1VCC		62	253	mA
RUN mode current (HALT state)	IDDH	Run (160 MHz)	Run	–40 to 150°C	Run (#3)	Total		78	288	mA
						REG1VCC		58	249	mA

Products of CPU frequency 240 MHz max, 160 MHz max.

		Condition								
Item	Symbol	CPU	PLL	Tj	Peripheral*2	Power supply	MIN.	TYP.*1	MAX.	Unit
STOP mode current	IDDS	Stop	Stop	–40 to 90°C	Stop (#2)	Total		2.2	48	mA
						REG1VCC		2	45	mA
				110°C	Stop (#2)	Total			88	mA
						REG1VCC			83	mA
				135°C	Stop (#2)	Total			138	mA
						REG1VCC			130	mA
DeepSTOP mode current	IDDDS	Power off	Power off	–40 to 85°C	Stop (#3)	Total		52	800	µA
						REG1VCC		1	10	µA
				105°C	Stop (#3)	Total			1480	µA
						REG1VCC			30	µA
				125°C	Stop (#3)	Total			2140	µA
						REG1VCC			60	µA
Cyclic RUN mode current	IDDCR	Run (HS IntOSC)	Stop	–40 to 90°C	Run (#4)	Total		9.8	58	mA
						REG1VCC		9.1	49	mA
				115°C	Run (#4)	Total			97	mA
						REG1VCC			86	mA
				135°C	Run (#4)	Total			146	mA
						REG1VCC			133	mA
Cyclic STOP mode current	IDDCS	Stop	Stop	–40 to 90°C	Run (#5)	Total		2.4	50	mA
						REG1VCC		2.2	47	mA
				110°C	Run (#5)	Total			88	mA
						REG1VCC			83	mA
				135°C	Run (#5)	Total			138	mA
						REG1VCC			130	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- REG0VCC = EVCC = BVCC = A0VREF = A1VREF = 5.0 V
- REG1VCC = 3.3 V
- AWOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V

Note 2. Operating condition of each peripheral function is shown in the table of next page.

Caution: It must be ensured that the junction temperature in the Ta range remains below Tj ≤ 150°C and does not exceed its limit under application conditions (thermal resistance, power supply current, peripheral current (if not included in power supply current), port output current and injection current).

Function		Run						Stop		
		(#1)	(#2)	(#3)	(#4)	(#5)	(#6)	(#1)	(#2)	(#3)
AWO	MainOSC	Run	Run	Run	Stop	Stop	Run	Run	Stop	Stop
	SubOSC	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	HS IntOSC	Run	Run	Run	Run	Stop	Run	Run	Stop	Stop
	FOUT	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	LPS	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	RRAM	Read/Write	Read/Write	No access	Fetch	No access	Read/Write	Read/Write	No access	No access
	WDTA0	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	TAUJ0, TAUJ2	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Run	Stop	Stop	Stop
	RTCA0	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Run	Stop	Stop	Stop
	CLMA0	Run	Run	Run	Run	Stop	Run	Stop	Stop	Stop
	CLMA1	Run	Run	Run	Stop	Stop	Run	Stop	Stop	Stop
	ADCA0	Run*1	Run*1	Run*1	Stop	Stop	Run*1	Stop	Stop	Stop

		Run						Stop		
Function		(#1)	(#2)	(#3)	(#4)	(#5)	(#6)	(#1)	(#2)	(#3)
ISO	CPU1 (PE1)	Run (PLL0)	Run (PLL0)	HALT (PLL0)	Run (HS IntOSC)	Stop	Run (PLL0)	Run (PLL0)	Stop	Power off
	CPU2 (PE2)	Run (PLL0)	Run (PLL0)	HALT (PLL0)	Stop	Stop	Run (PLL0)	Run (PLL0)	Stop	
	ICUMD	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	DMA	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	PLL0	Run	Run	Run	Stop	Stop	Run	Run	Stop	
	PLL1	Run	Run	Run	Stop	Stop	Run	Run	Stop	
	Code flash (FLI0)	Fetch	Fetch	No access	No access	No access	Fetch	Fetch	No access	
	Code flash (FLI1)	Fetch	Fetch	No access	No access	No access	Write/Erase	Fetch	No access	
	Code flash (FLI2)	Fetch	Fetch	No access	No access	No access	Fetch	Fetch	No access	
	Code flash (FLI3)	Fetch	Fetch	No access	No access	No access	Fetch	Fetch	No access	
	Data flash	Read	Write/Erase	No access	No access	No access	No access	Read	No access	
	LRAM (PE1)	Read/Write	Read/Write	No access	No access	No access	Read/Write	Read/Write	No access	
	LRAM (PE2)	Read/Write	Read/Write	No access	No access	No access	Read/Write	Read/Write	No access	
	GRAM	Read/Write*2	Read/Write*2	No access	No access	No access	Read/Write*2	Read/Write*2	No access	
	OSTMn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	WDTA1	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	WDTA2	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	TAUD0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAUBn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAUJ1, TAUJ3	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAPA, PIC	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	ENCA0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	PWM-diag	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RLIN3n	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RLIN24n	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	RCFDCn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	CSIGn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	CSIHn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RIICn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	FlexRay	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	ETNBn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	SFMA0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	KR	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	RSENTn	Run	Run	Run	Stop	Stop	Wait	Stop	Stop	
	MMCAn	Run	Run	Run	Stop	Stop	Wait	Stop	Stop	
	CLMA2	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	CLMA3	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	ADCA1	Run	Run	Run	Stop	Stop	Run	Stop	Stop	

Note 1. T&H used.

Note 2. GRZF not used.

47A.4.4 Injection Currents

Table 47A.1 Definition of Pin Group (324-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR0	REG0VCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P3, P20, P23
PgB	BVCC, BVSS	P10, P11, P12, P13, P21, P22, P24
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18, P19
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47A.2 Reserved

Table 47A.3 Definition of Pin Group (233-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR0	REG0VCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P3, P20
PgB	BVCC, BVSS	P10, P11, P12, P13
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18, P19
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47A.4 Definition of Pin Group (176-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR0	REG0VCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P20
PgB	BVCC, BVSS	P10, P11, P12
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

47A.4.4.1 Absolute Maximum Ratings

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgB	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgB’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
		PgA1	Per pin			10	mA
			Total			60	mA
		PgR0	Per pin			10	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgB	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgB’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA
		PgA1	Per pin			–10	mA
			Total			–60	mA
		PgR0	Per pin			–10	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0 and PgA1, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

47A.4.4.2 DC Characteristics for Overload Current

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgB	Per pin			2	mA
			Total			50	mA
		PgE’	Per pin			3	mA
			Total			20	mA
		PgB’	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
		PgA1	Per pin			3	mA
			Total			20	mA
		PgR0	Per pin			2	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgB	Per pin			–2	mA
			Total			–50	mA
		PgE’	Per pin			–3	mA
			Total			–20	mA
		PgB’	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA
		PgA1	Per pin			–3	mA
			Total			–20	mA
		PgR0	Per pin			–2	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

47A.4.5 Power Management Characteristics

47A.4.5.1 Regulator Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	REG0VCC		VPOC*[1]		5.5	V
	REG1VCC	REG1VCC ≤ REG0VCC	VPOC*[1]		3.6	V
Output voltage	AWOVCL	AWOVCL pin	1.15	1.25	1.35	V
	ISOVCL	ISOVCL pin	1.15	1.25	1.35	V
Capacitance	CAWOVCL	AWOVCL pin	0.07	0.10	0.13	μF
	CISOVCL	ISOVCL pin	0.07	0.10	0.13	μF
Equivalent series resistance for load capacitance	RVRAWO	for CAWOVCL			40*[2]	mΩ
	RVRISO	for CISOVCL			40*[2]	mΩ
Inrush current during power-on	REG0VCC				200	mA
	REG1VCC				350	mA

Note 1. “VPOC” means POC (power-on clear) detection voltage (typ. 2.85 V). For detail, see **Section 47A.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

Note 2. This is reference value.

47A.4.5.2 Voltage Detector (POC, LVI, VLVI, CVM) Characteristics

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Detection voltage (REG0VCC, REG1VCC)	VPOC	POC			2.7	2.85	3.0	V
Detection voltage (REG0VCC)	VLVI0	LVI	Rise		3.87	4.0	4.13	V
			Fall		3.9	4.0	4.1	V
	VLVI1		Rise		3.57	3.7	3.83	V
			Fall		3.6	3.7	3.8	V
	VLVI2		Rise		3.37	3.5	3.63	V
			Fall		3.4	3.5	3.6	V
	VVLVI	VLVI			1.8	1.9	2.0	V
Detection voltage (ISOVCL)	VCVMH	CVM	High voltage[Caution]		1.35	1.39	1.43	V
	VCVML*8		Low voltage[Caution]		1.10	1.15	1.20	V
Response time	t_{D_POC1}*6	POC	At power-on (Rise)	*1			2	ms
				*2			6.3	ms
			After power-on (Rise)	*3			2	ms
				*4			5	ms
	t_{D_POC2}*7		After power-on (Fall)	*5			5	μs
	t_{D_LVI}	LVI					2	ms
	t_{D_VLVI}	VLVI		*3			2	ms
				*4			5	ms
	t_{D_CVM}	CVM			0.2		10	μs
Setup time	t_{S_LVI}	LVI	LVICNT0,1 bits are set to 1 (except 00_B), then LVI is ready to operate				80	μs
REG0VCC, REG1VCC minimum width	t_{W_POC}	POC			0.2			ms
REG0VCC minimum width	t_{W_LVI}	LVI			0.2			ms
	t_{W_VLVI}	VLVI			0.2			ms

Note 1. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms

Note 2. Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms

Note 3. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 20 V/ms

Note 4. Voltage slope (t_{VS}): 20 V/ms < t_{VS} ≤ 500 V/ms

Note 5. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 500 V/ms

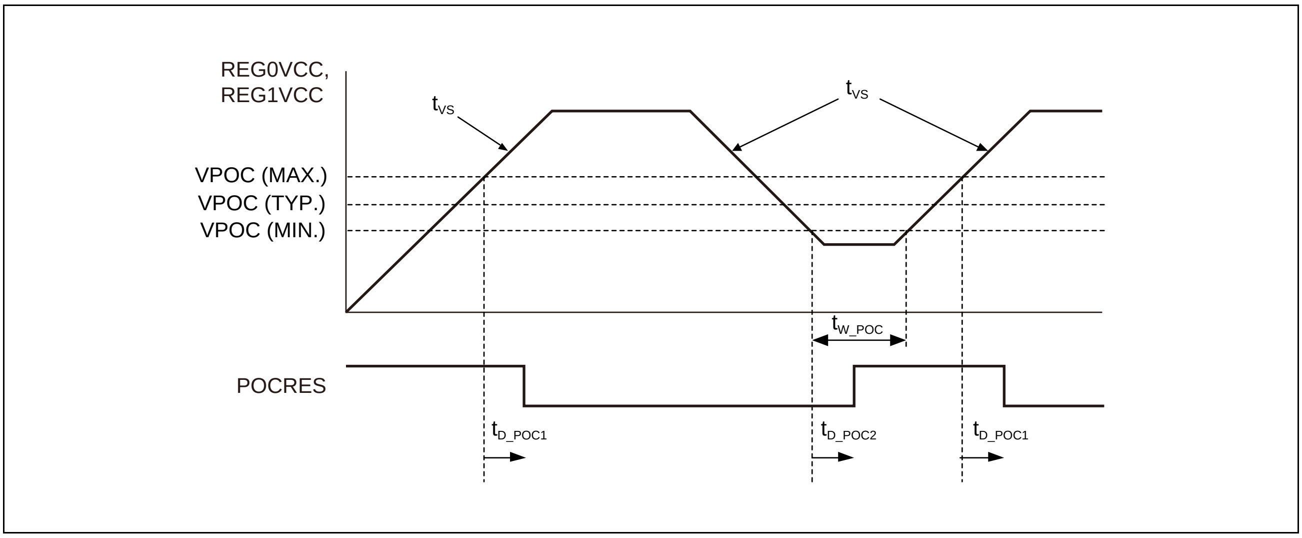
Note 6. t_{D_POC1} is the time from detection voltage to release of reset signal.

Note 7. t_{D_POC2} is the time from detection voltage to occurrence of reset signal.

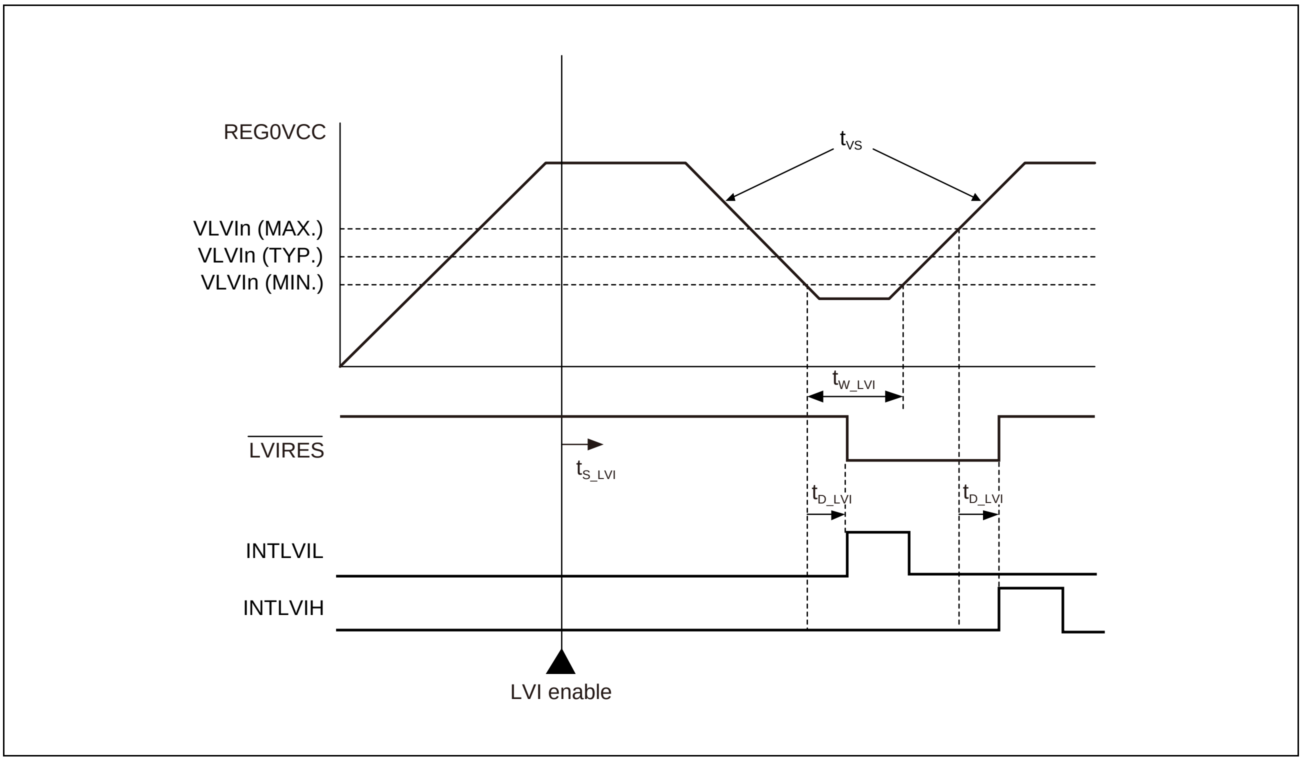
Note 8. The CVM monitors the internal voltage regulator output to ensure that ISOVCL is upper than specified minimum level.

Caution: A detection of the voltage ISOVCL outside the specified level of VCVMH and VCVML is not ensured by CVM.

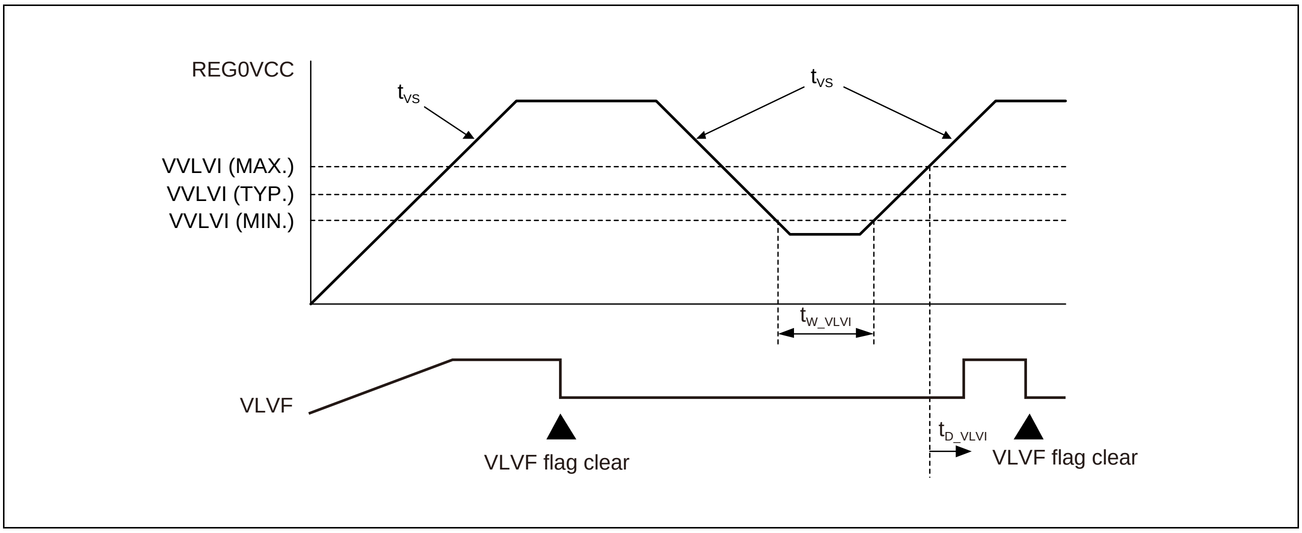
POC


REG0VCC,
REG1VCC
tVS
tVS
VPOC (MAX.)
VPOC (TYP.)
VPOC (MIN.)
tW_POC
POCRES
tD_POC1
tD_POC2
tD_POC1

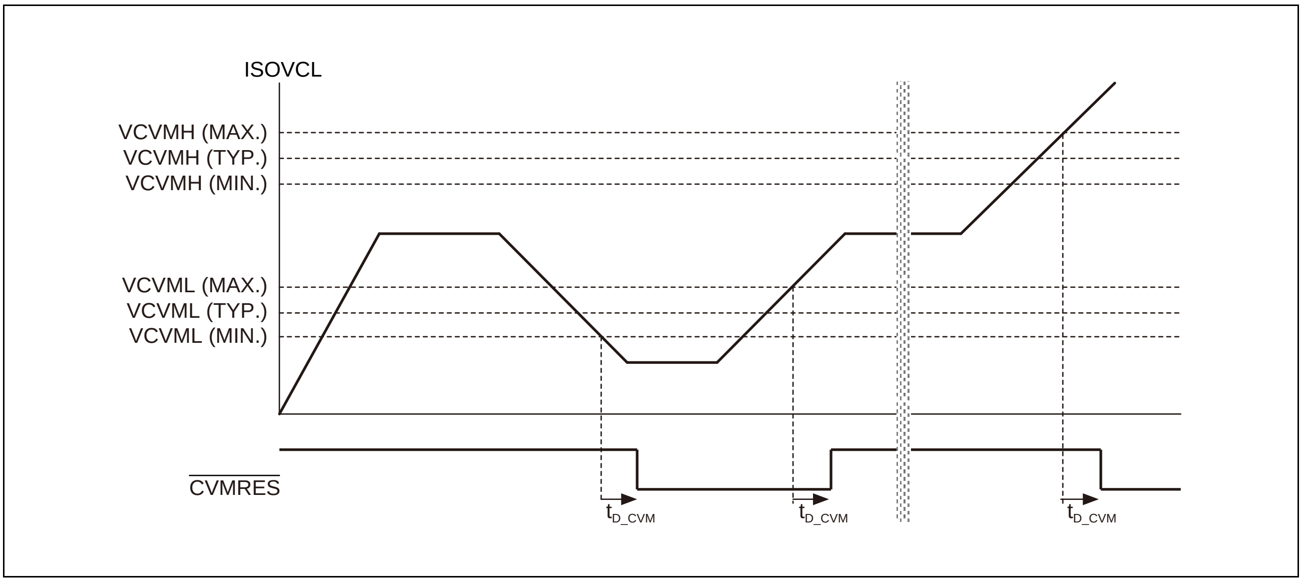
LVI


REG0VCC
tVS
VLVIn (MAX.)
VLVIn (TYP.)
VLVIn (MIN.)
tW_LVI
LVIRES
tS_LVI
tD_LVI
tD_LVI
INTLVIL
INTLVIH
LVI enable

VLVI


REG0VCC
t_{VS}
t_{VS}
VVLVI (MAX.)
VVLVI (TYP.)
VVLVI (MIN.)
t_{W_VLVI}
VLVF
t_{D_VLVI}
VLVF flag clear
VLVF flag clear

CVM


ISOVCL
VCVMH (MAX.)
VCVMH (TYP.)
VCVMH (MIN.)
VCVML (MAX.)
VCVML (TYP.)
VCVML (MIN.)
$\overline{\text{CVMRES}}$
t_{D_CVM}
t_{D_CVM}
t_{D_CVM}

47A.4.5.3 Power Up/Down Timing

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

CAUTION

REG1VCC must not be greater than REG0VCC during power up/down.

Table 47A.5 In Case the $\overline{\text{RESET}}$ Pin is Used (for Normal Operating Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REG0VCC, REG1VCC and IOVCC^{*1})	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REG0VCC ↑, REG1VCC ↑ and IOVCC^{*1} ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↑)	t_{HMDR}		1			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↓)	t_{SMDF}		0			μs
$\overline{\text{RESET}}$ ↓ to REG0VCC ↓, REG1VCC ↓ and IOVCC^{*1} ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

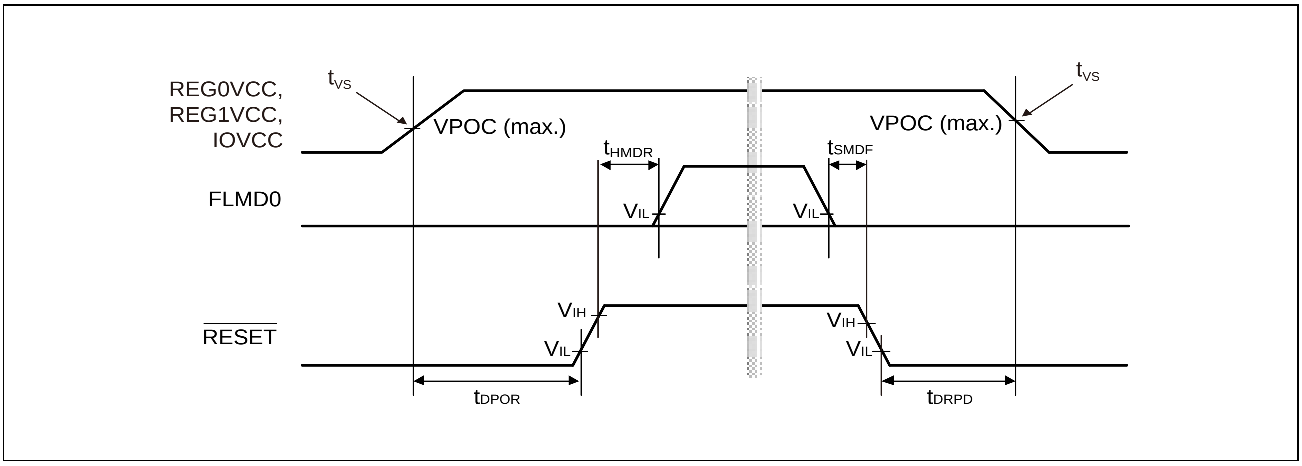


Table 47A.6 In Case the $\overline{\text{RESET}}$ Pin is Used (for Serial Programming Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REG0VCC, REG1VCC and IOVCC^{*1})	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 µs/V)	V/ms
REG0VCC ↑, REG1VCC ↑ and IOVCC^{*1} ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
$\overline{\text{RESET}}$ ↓ to REG0VCC ↓, REG1VCC ↓ and IOVCC^{*1} ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

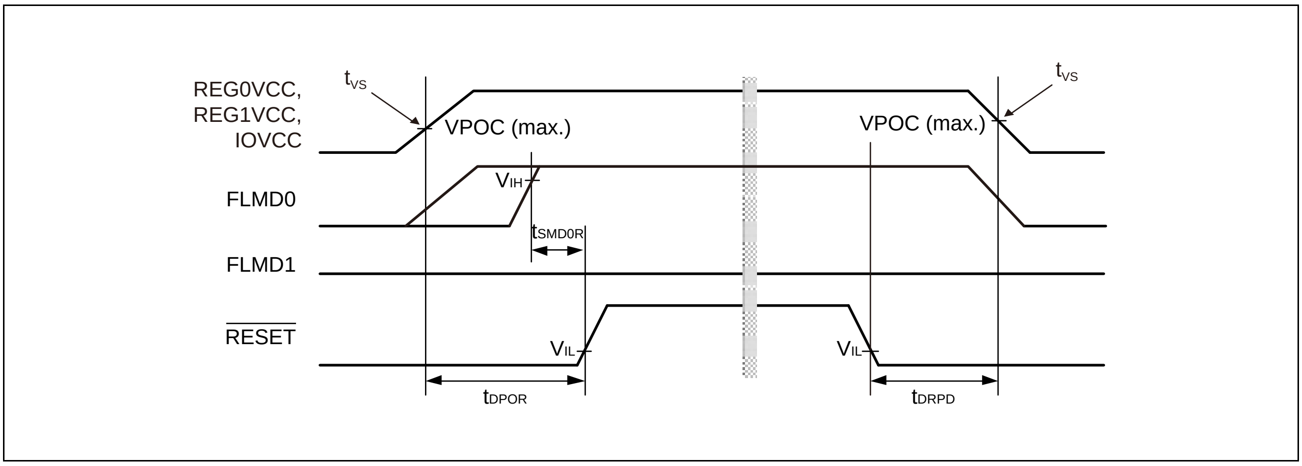


Table 47A.7 In Case the $\overline{\text{RESET}}$ Pin is Used (for Boundary Scan Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REG0VCC, REG1VCC and IOVCC*1)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REG0VCC ↑, REG1VCC ↑ and IOVCC*1 ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1,MODE0,MODE1 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			μs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			μs
FLMD1, MODE0, MODE1, MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			μs
$\overline{\text{RESET}}$ ↓ to REG0VCC ↓, REG1VCC ↓ and IOVCC*1 ↓ delay time	t_{DRPD}		0			ms
$\overline{\text{DCUTRST}}$ input delay time (vs $\overline{\text{RESET}}$ ↑)	t_{DRTRST}		1			ms
$\overline{\text{RESET}}$ hold time (vs $\overline{\text{DCUTRST}}$ ↓)	t_{HRTRST}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

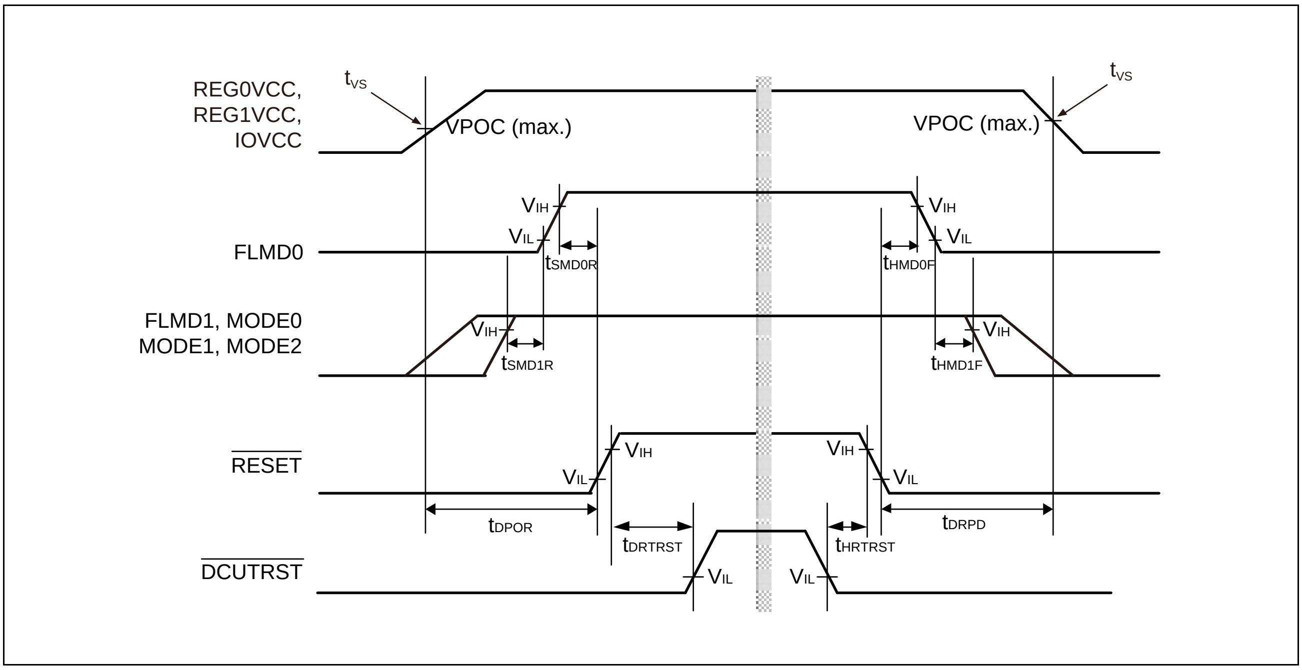


Table 47A.8 In Case the $\overline{\text{RESET}}$ Pin is Used (for User Boot Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REG0VCC, REG1VCC and IOVCC*1)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REG0VCC ↑, REG1VCC ↑ and IOVCC*1 ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1, MODE0, MODE1, MODE2 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			μs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			μs
FLMD1, MODE0, MODE1, MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			μs
$\overline{\text{RESET}}$ ↓ to REG0VCC ↓, REG1VCC ↓ and IOVCC ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

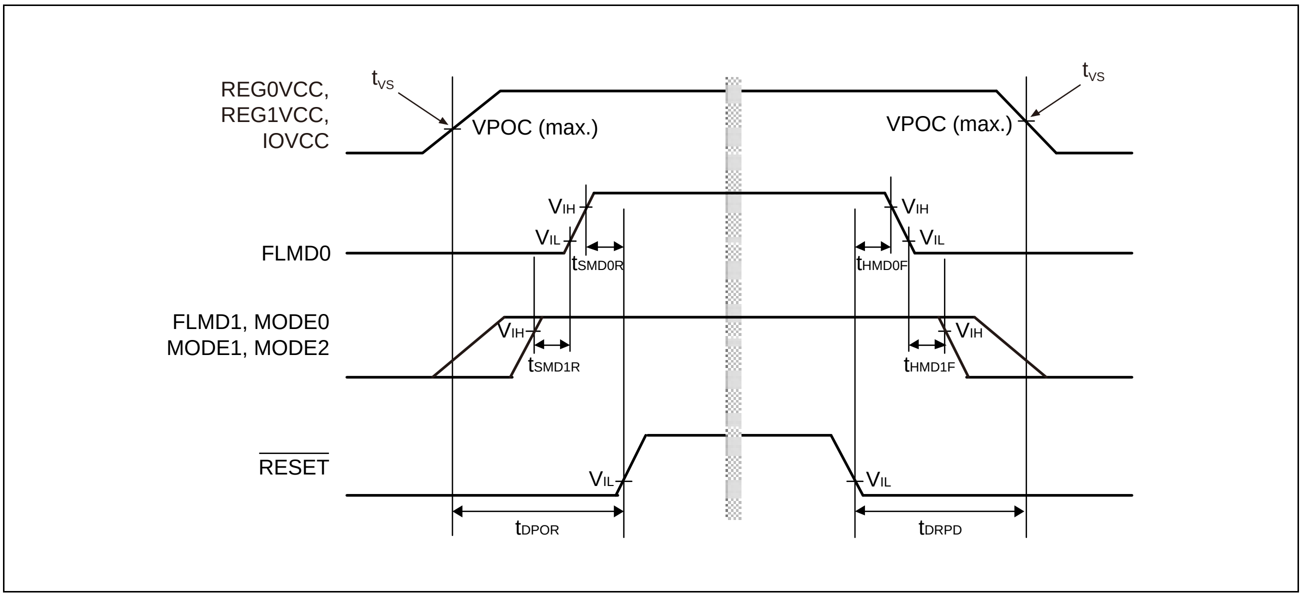
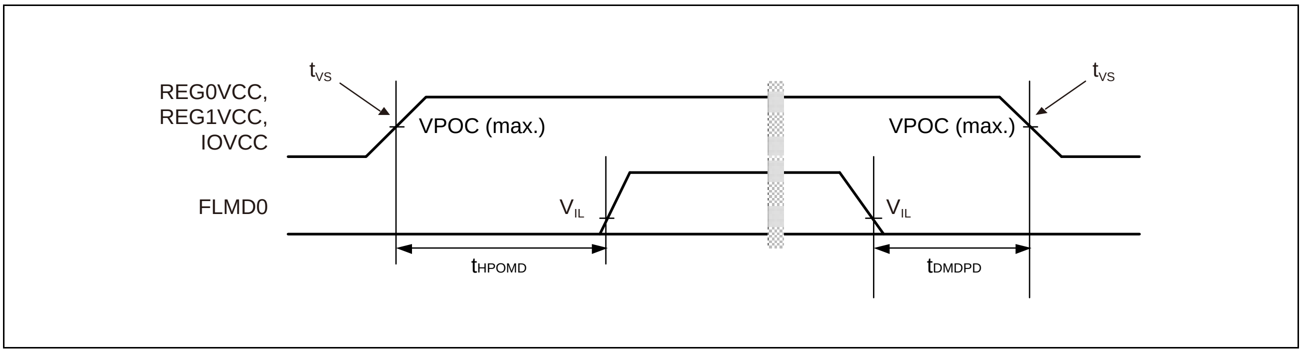


Table 47A.9 In Case the $\overline{\text{RESET}}$ Pin is Not Used and Fixed to High Level by Pull-up*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REG0VCC, REG1VCC and IOVCC*2)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 µs/V)	V/ms
REG0VCC ↑, REG1VCC ↑ and IOVCC*2 ↑ to FLMD0 hold time	t_{HPOMD}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 ↓ to REG0VCC ↓, REG1VCC ↓ and IOVCC*2 ↓ delay time	t_{DMDPD}		1			µs

Note 1. This operating condition is available only in normal operation mode (include self-programming mode). When the device is used in except normal operation mode, please use the $\overline{\text{RESET}}$ pin.

Note 2. IOVCC means EVCC, BVCC, A0VREF and A1VREF.



47A.4.5.4 CPU Reset Release Timing

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47A.10 In Case the $\overline{\text{RESET}}$ Pin is Not Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
REG0VCC ↑ and REG1VCC ↑ to CPU reset release*[1]	t_{DPCRR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms			2.58	ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms			8.3	ms

Note 1. This is reference value.

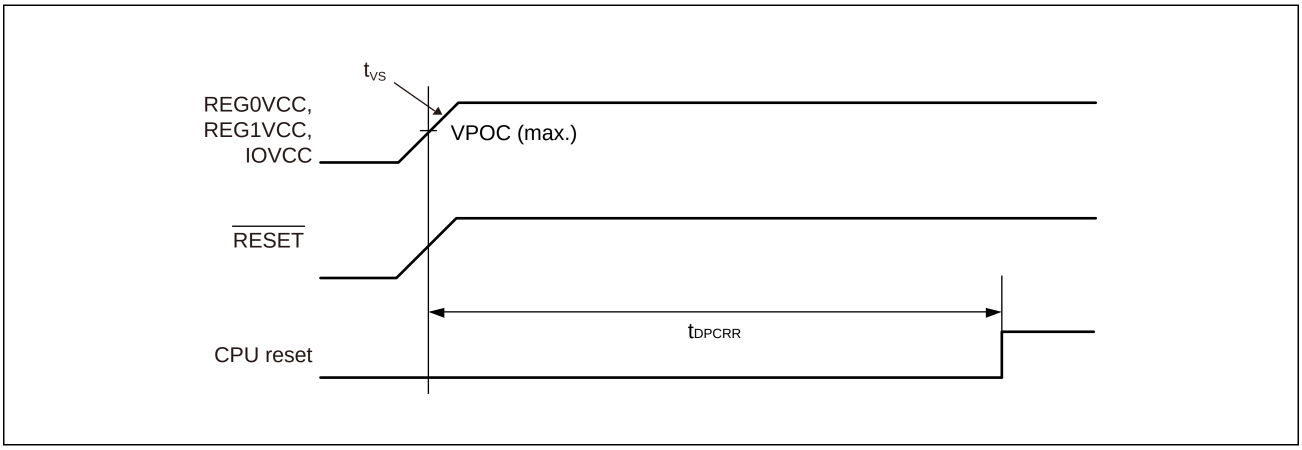
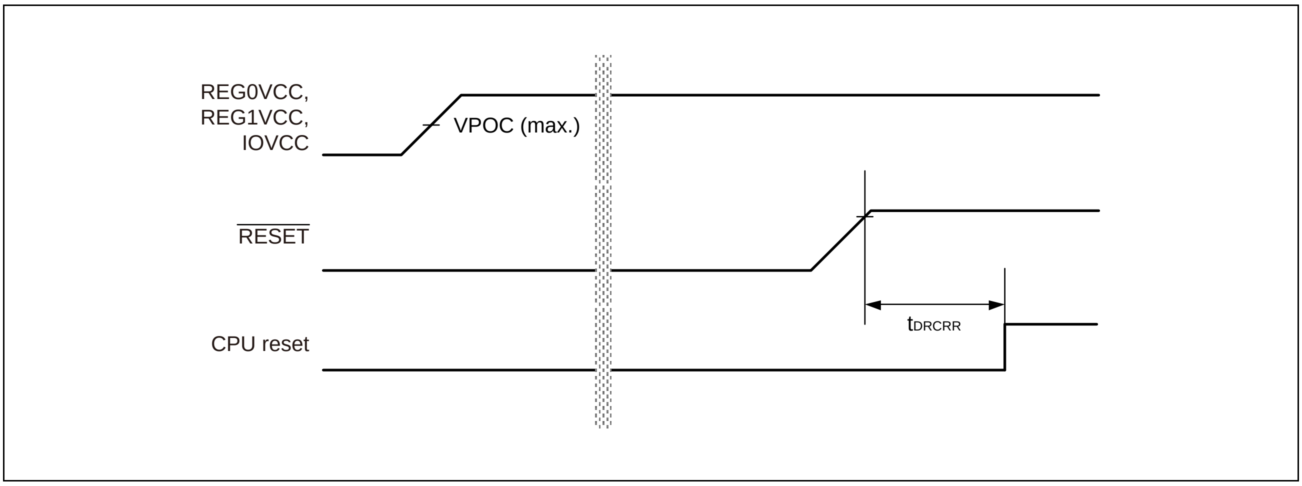


Table 47A.11 In Case the $\overline{\text{RESET}}$ Pin is Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ ↑ to CPU reset release*[1]	t_{DRCRR}				32*[2]	µs

Note 1. This is reference value.

Note 2. In case the time until releasing the $\overline{\text{RESET}}$ pin is longer than t_{DPCRR}.



47A.5 AC Characteristics

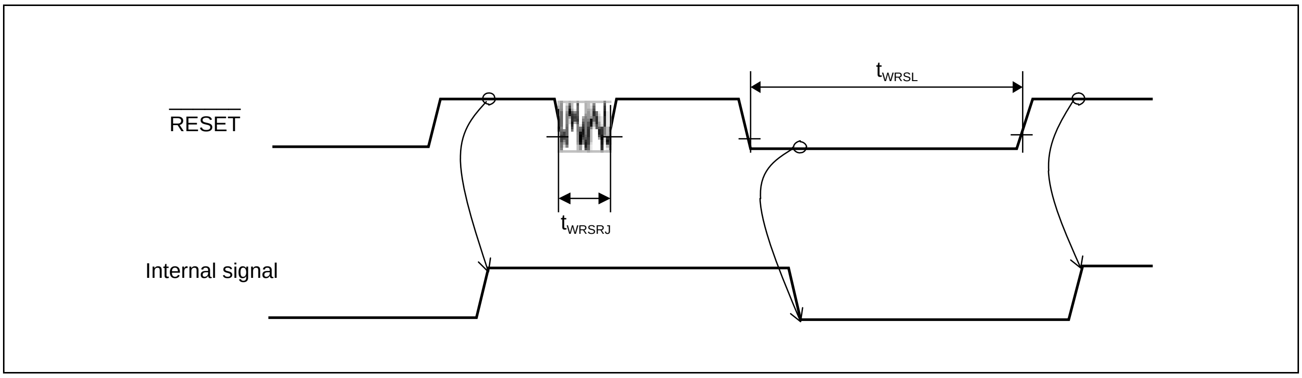
47A.5.1 $\overline{\text{RESET}}$ Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width*[1]	t_{WRSL}	Except power on	600			ns
$\overline{\text{RESET}}$ pulse rejection*[2]	t_{WRSRJ}		100			ns

Note 1. $\overline{\text{RESET}}$ input width is needed to ensure that the internal reset signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



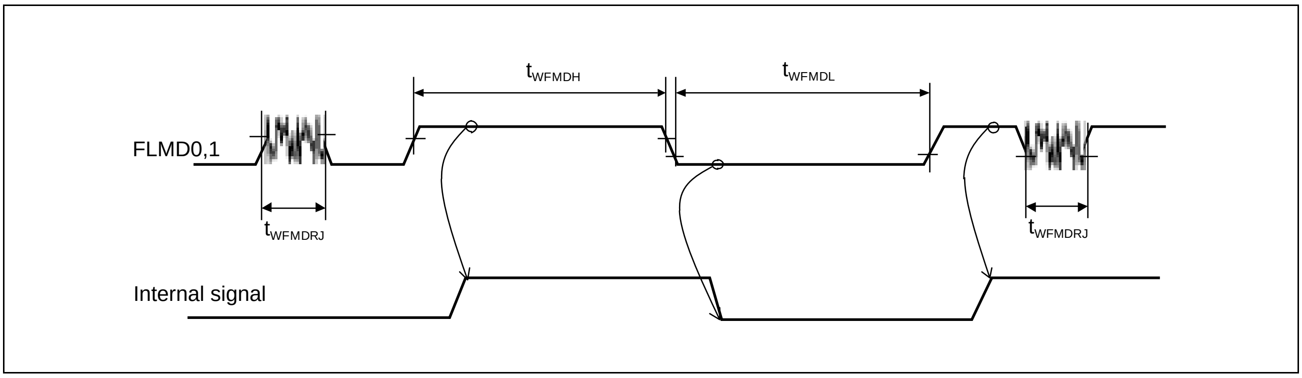
47A.5.2 Mode Timing

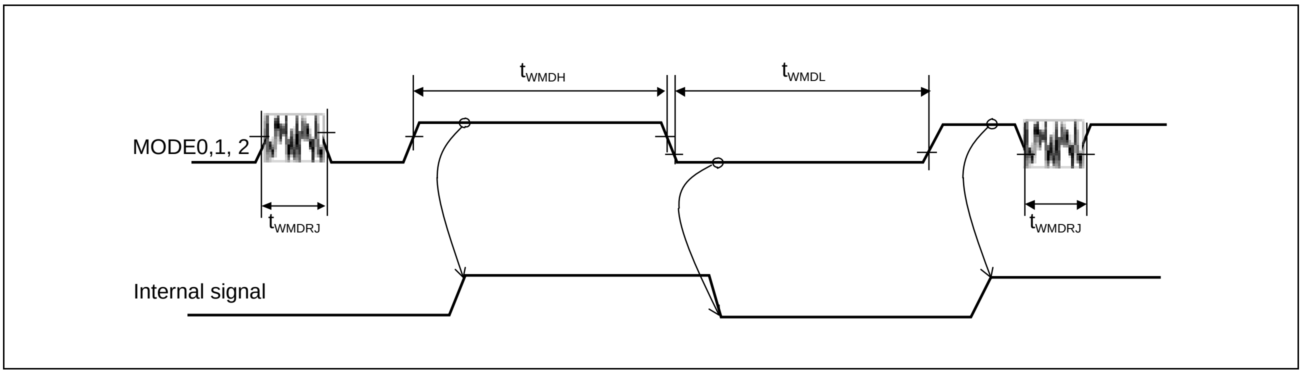
Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0, 1 input high/low level width*[1]	t_{WFMDH}/ t_{WFMDL}		600			ns
FLMD0, 1 pulse rejection*[2]	t_{WFMDRJ}		100			ns
MODE0, 1, 2 input high/low level width*[1]	t_{WMDH}/ t_{WMDL}		600			ns
MODE0, 1, 2 pulse rejection*[2]	t_{WMDRJ}		100			ns

Note 1. FLMD0, 1 and MODE0, 1, 2 input width is needed to ensure that the internal mode signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.





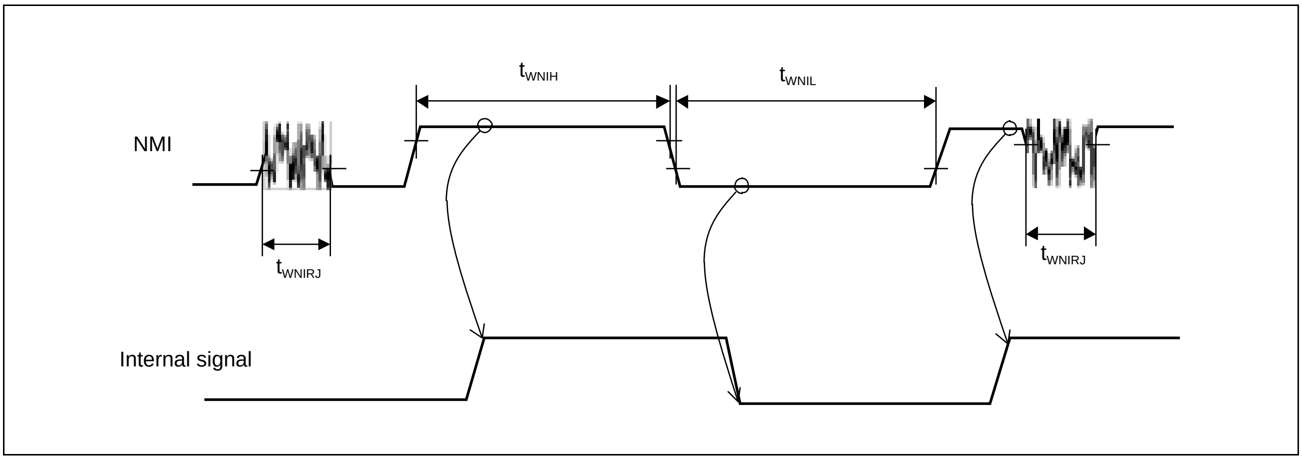
47A.5.3 Interrupt Timing

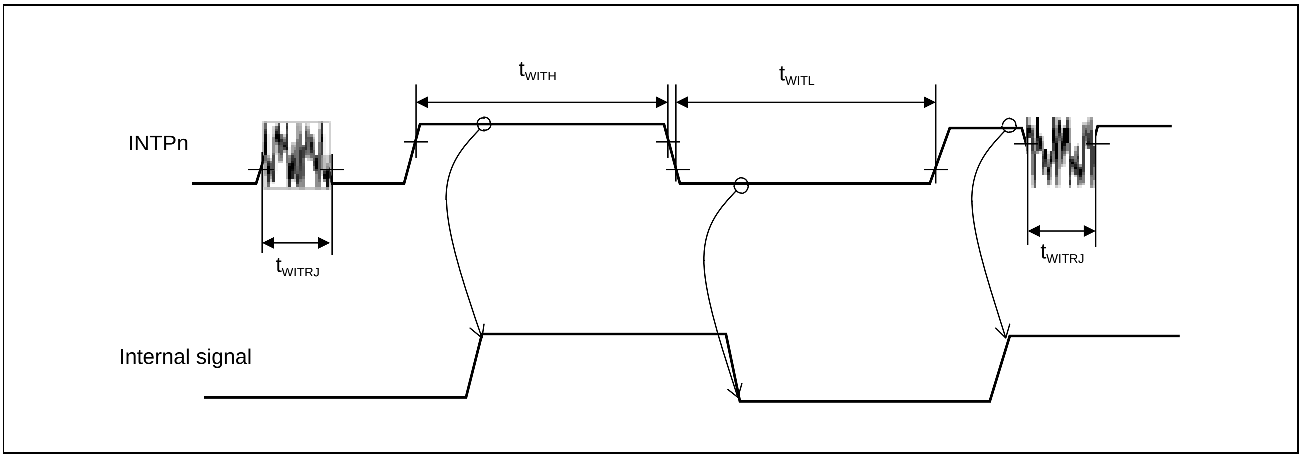
Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
NMI input high/low level width*1	t_{WNIH}/ t_{WNIL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			µs
NMI pulse rejection*2	t_{WNIRJ}		100			ns
INTPn input high/low level width*1	t_{WITH}/ t_{WITL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			µs
INTPn pulse rejection*2	t_{WITRJ}		100			ns

Note 1. NMI and INTPn input width is needed to ensure that the internal interrupt signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



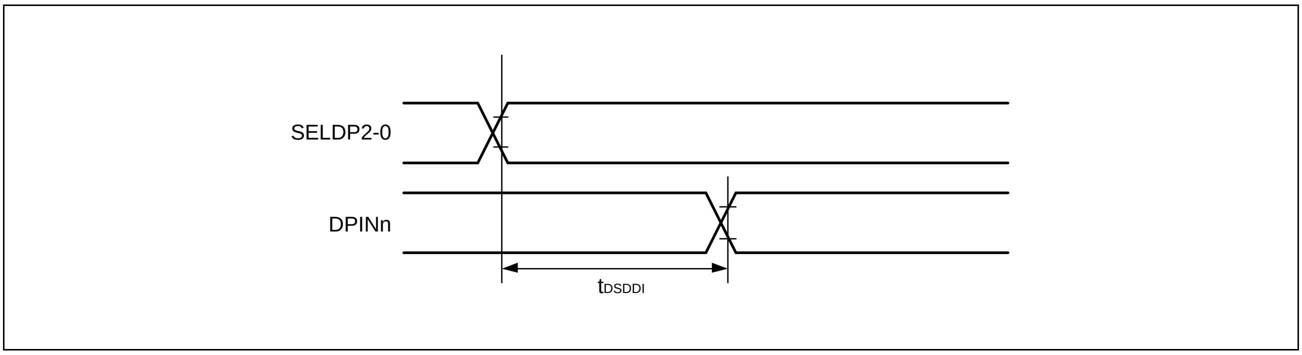


47A.5.4 Low Power Sampler (DPIN input) Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DPINn input delay time (vs SELDP2-0)	t_{DSDDI}				150	ns

Note: n = 7 to 0



47A.5.5 CSCXFOUT Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>

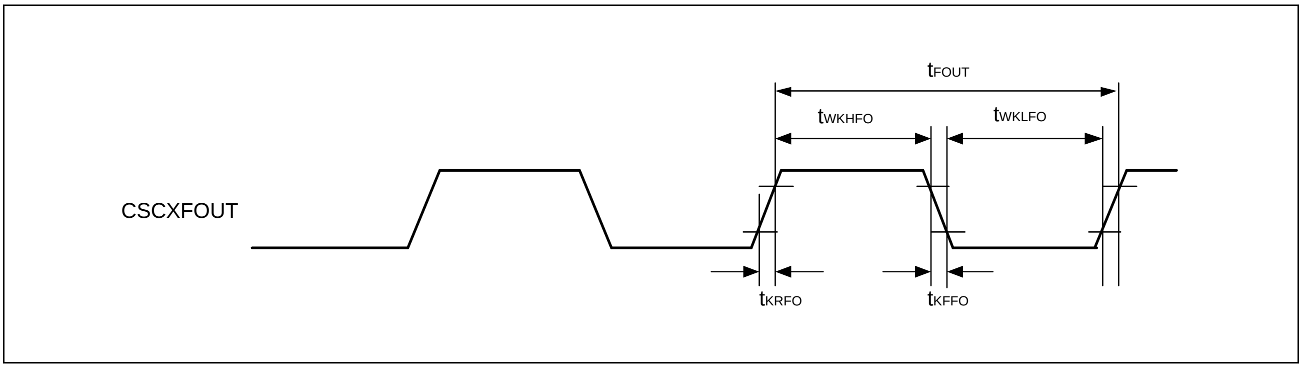
CSCXFOUT: Slow or fast mode (refer to the condition in the following table)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
CSCXFOUT output cycle	t_{FOUT}	Slow mode		100 (max. 10 MHz)			ns
		Fast mode		41.6 (max. 24 MHz)			ns
CSCXFOUT high level width	t_{WKHFO}	Slow mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)[*2,*3]	t_{FOUT} × (N+1) / 2N – 37			ns
		Fast mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)[*2]	t_{FOUT} × (N+1) / 2N – 10			ns
CSCXFOUT low level width	t_{WKLFO}	Slow mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)[*2,*3]	t_{FOUT} × (N–1) / 2N – 37			ns
		Fast mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)[*2]	t_{FOUT} × (N–1) / 2N – 10			ns
CSCXFOUT rise/ fall time	t_{KRFO}/ t_{KFFO}	Slow mode				37	ns
		Fast mode				10	ns

Note 1. When MainOSC, HS IntOSC, LS IntOSC or SubOSC is selected as source clock with the condition of N = 1, the characteristics of output signal depends on the selected source clock. It is recommended to use output signal after evaluation on an actual environment.

Note 2. "N" is the value of "Clock divisor N" defined by FOUTDIV register.

Note 3. The selection of N = 3 is prohibited when slow mode is used.

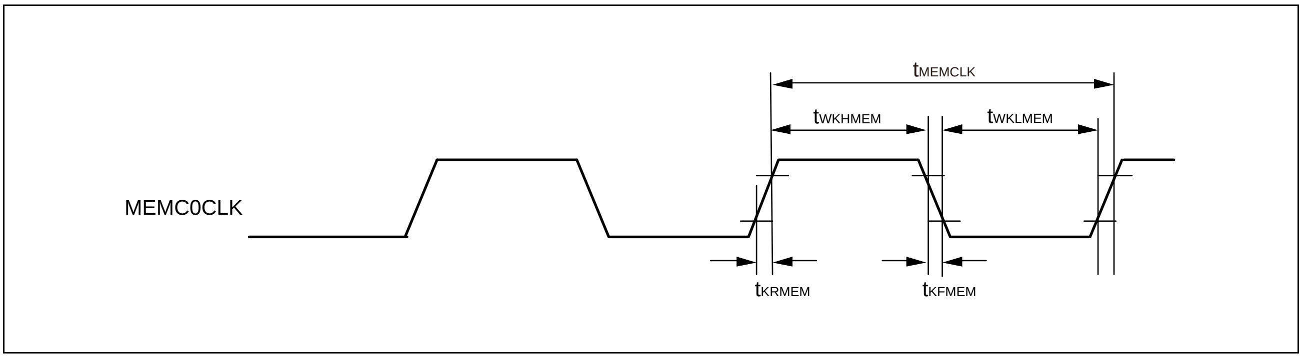


47A.5.6 MEMC0CLK Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>
MEMC0CLK pin: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MEMC0CLK output cycle	t_{MEMCLK}		33.4 (max.30 MHz)			ns
MEMC0CLK high / low level width	t_{WKHMEM} / t_{WKLMEM}		t_{MEMCLK} / 2 – 10			ns
MEMC0CLK rise / fall time	t_{KRMEM} / t_{KFMEM}				10	ns



47A.5.7 External Bus Timing

47A.5.7.1 MEMC0CLK Asynchronous

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>

MEMC0AD0-15, MEMC0A16-23, $\overline{\text{MEMC0CS3-0}}$,
$\overline{\text{MEMC0BEN1-0}}$, $\overline{\text{MEMC0ASTB}}$, $\overline{\text{MEMC0WR}}$, and $\overline{\text{MEMC0RD}}$ pins: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Bus operational period	T		33.4 (max.30 MHz)			ns
Address*4 setup time to $\overline{\text{MEMC0ASTB}}$ ↓	t_{SAST}	<1>	(1 + ASW) × T – 15			ns
Address (MEMC0AD15-0) hold time from $\overline{\text{MEMC0ASTB}}$ ↓	t_{HSTA}	<2>	(1 + AHW) × T – 15			ns
Address (MEMC0AD15-0) float delay time from $\overline{\text{MEMC0RD}}$ ↓	t_{FRDA}*8	<3>			9	ns
Address*5 hold time from $\overline{\text{MEMC0RD}}$ ↑	t_{HRDA}	<4>	–1.5			ns
Data (MEMC0AD15-0) input delay time from $\overline{\text{MEMC0RD}}$ ↓	t_{DRDID}	<5>	9		(1 + w) × T – 35	ns
Data (MEMC0AD15-0) input hold time from $\overline{\text{MEMC0RD}}$ ↑	t_{HRDID}	<6>	0			ns
Delay time from $\overline{\text{MEMC0ASTB}}$ ↓ to $\overline{\text{MEMC0RD}}$ ↓	t_{DSTRD}	<7>	(1 + AHW) × T – 15			ns
Delay time from $\overline{\text{MEMC0ASTB}}$ ↓ to $\overline{\text{MEMC0WR}}$ ↓	t_{DSTWR}	<8>	(1 + AHW) × T – 15			ns
$\overline{\text{MEMC0RD}}$, $\overline{\text{MEMC0WR}}$ low level width	t_{WRDST}	<9>	(1 + w) × T – 10			ns
Data (MEMC0AD15-0) output delay time from $\overline{\text{MEMC0WR}}$ ↓	t_{DWROD}	<10>			11	ns
Address*5 hold time from $\overline{\text{MEMC0WR}}$ ↑	t_{HWRA}	<11>	(1 + DHW) × T – 15			ns
Data (MEMC0AD15-0) output setup time to $\overline{\text{MEMC0WR}}$ ↑	t_{SODWR}	<12>	(1 + w) × T – 15			ns
Data (MEMC0AD15-0) output hold time from $\overline{\text{MEMC0WR}}$ ↑	t_{HWROD}	<13>	(1 + DHW) × T – 15			ns
$\overline{\text{MEMC0WAIT}}$ setting delay from $\overline{\text{MEMC0ASTB}}$ ↓	t_{SSTWT1}	<14>			(AHW + DPW) × T – 24	ns
	t_{SSTWT2}	<15> DEW ≥ 1			(AHW + DPW + DEW) × T – 24	ns
$\overline{\text{MEMC0WAIT}}$ hold time from $\overline{\text{MEMC0ASTB}}$ ↓	t_{HSTWT1}	<16>	(AHW + DPW + DEW – 1) × T – 9			ns
	t_{HSTWT2}	<17> DEW ≥ 1	(AHW + DPW + DEW) × T – 9			ns

Note 1. ASW means the number of address setup wait for multiplex bus.

Note 2. AHW means the number of address hold wait for multiplex bus.

Note 3. DPW means the number of programmable data wait for multiplex bus.
DEW means the number of external data wait for multiplex bus.
“w” means the sum of DPW and DEW.

Note 4. t_{CPUCLK}: CPU clock period.

Note 5. DHW means the number of data hold wait for multiplex bus.

Note 6. Address means MEMC0AD15-0, MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$, and $\overline{\text{MEMC0BEN1-0}}$.
324-pin products support 24-bit address. 233/176-pin products support 23-bit address.

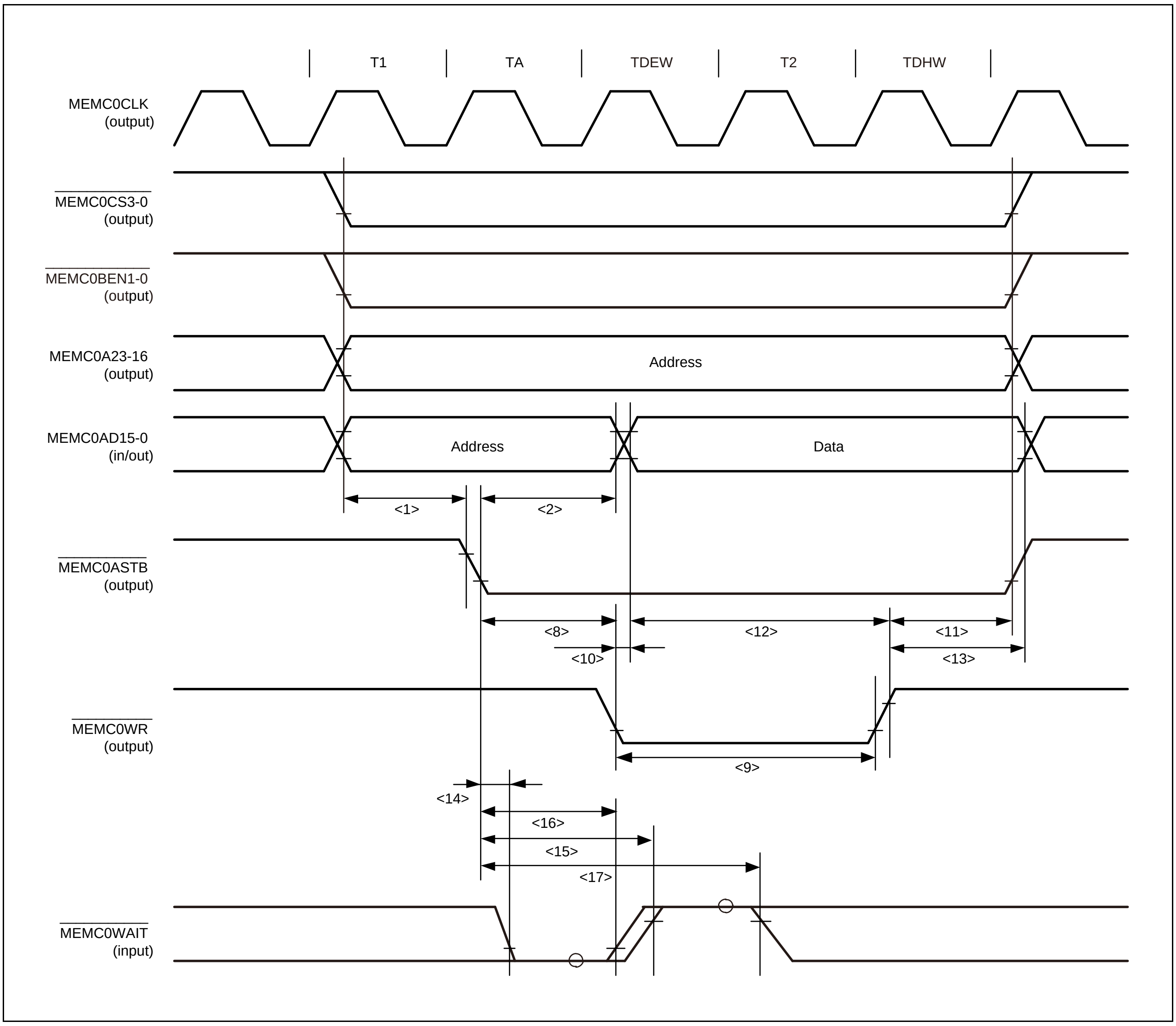
Note 7. Address means MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$, $\overline{\text{MEMC0BEN1-0}}$, and $\overline{\text{MEMC0ASTB}}$.
324-pin products support 24-bit address. 233/176-pin products support 23-bit address.

Note 8. t_{FRDA} means the period from output off to Hi-z for MEMC0AD15-0.

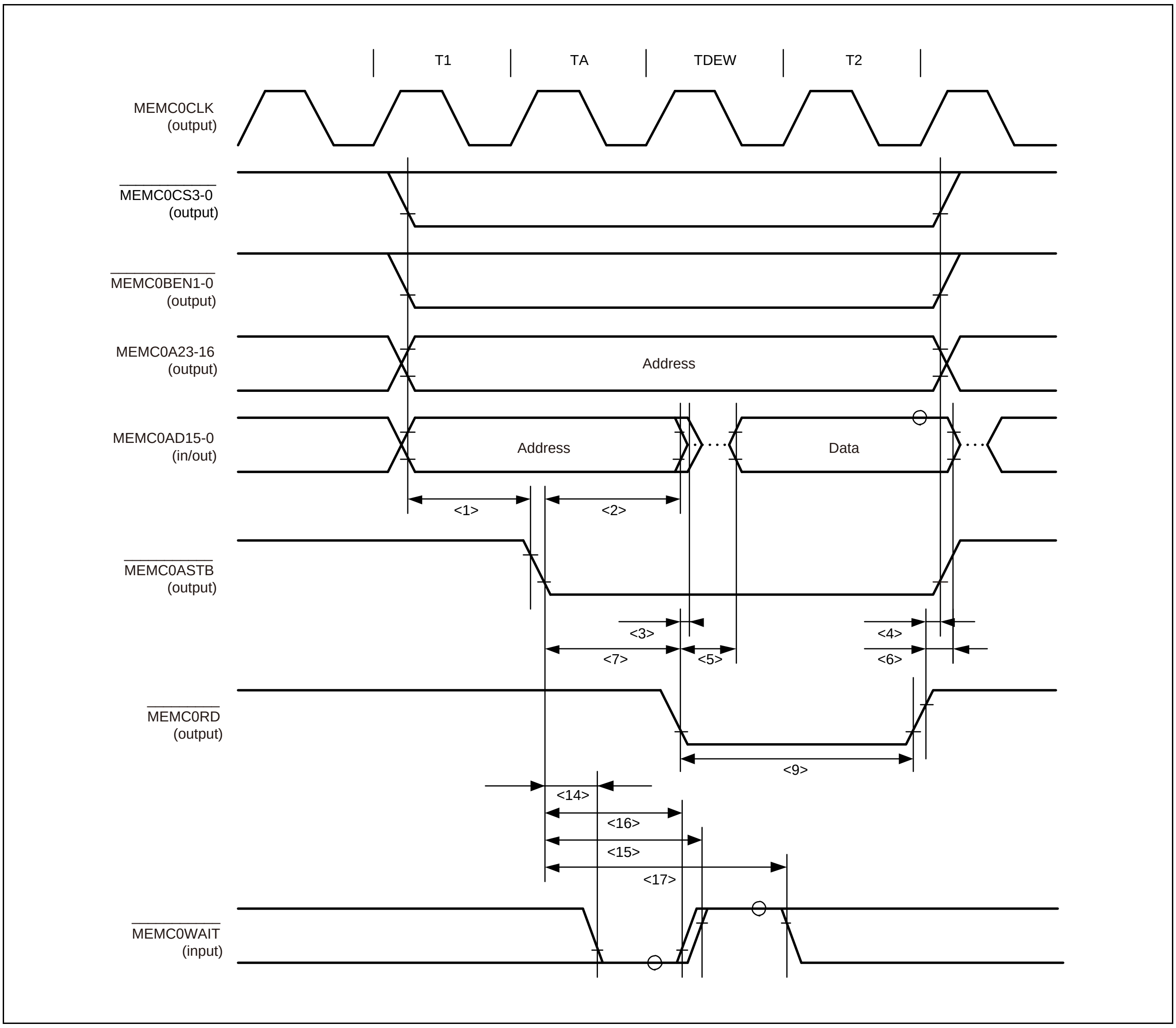
NOTE

When the bus period (T) is shorter than 44 ns, t_{DRDID} spec requires at least 1data wait. (w = 1)

(1) Multiplex Write Cycle (Asynchronous; 1 Data Wait)



(2) Multiplex Read Cycle (Asynchronous; 1 Data Wait)



47A.5.7.2 MEMC0CLK Synchronous

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>

MEMC0AD0-15, MEMC0A16-23, $\overline{\text{MEMC0CS3-0}}$,
$\overline{\text{MEMC0BEN1-0}}$, $\overline{\text{MEMC0ASTB}}$, $\overline{\text{MEMC0WR}}$, and $\overline{\text{MEMC0RD}}$ pins: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Bus operational period	T		33.4 (max.30 MHz)			ns
Delay time from MEMC0CLK ↑ to address[*1]	t_{DKA}	<18>	–0.5		15	ns
Delay time from MEMC0CLK ↑ to address (MEMC0AD15-0) float	t_{FKA}[*2]	<19>	0		12	ns
Delay time from MEMC0CLK ↑ to $\overline{\text{MEMC0ASTB}}$	t_{DKST}	<20>	0		11	ns
Delay time from MEMC0CLK ↑ to $\overline{\text{MEMC0RD}}$ and $\overline{\text{MEMC0WR}}$	t_{DKRDWR}	<21>	–2.5		6	ns
Data (MEMC0AD15-0) input setup time (from MEMC0CLK ↑)	t_{SIDK}	<22>	29			ns
Data (MEMC0AD15-0) input hold time (from MEMC0CLK ↑)	t_{HKID}	<23>	2.5			ns
Data (MEMC0AD15-0) output delay time (from MEMC0CLK ↑)	t_{DKOD}	<24>			15	ns
$\overline{\text{MEMC0WAIT}}$ setup time (to MEMC0CLK ↑)	t_{SWTK}	<25>	T + 22			ns
$\overline{\text{MEMC0WAIT}}$ hold time (from MEMC0CLK ↑)	t_{HKWT}	<26>	–T – 5			ns

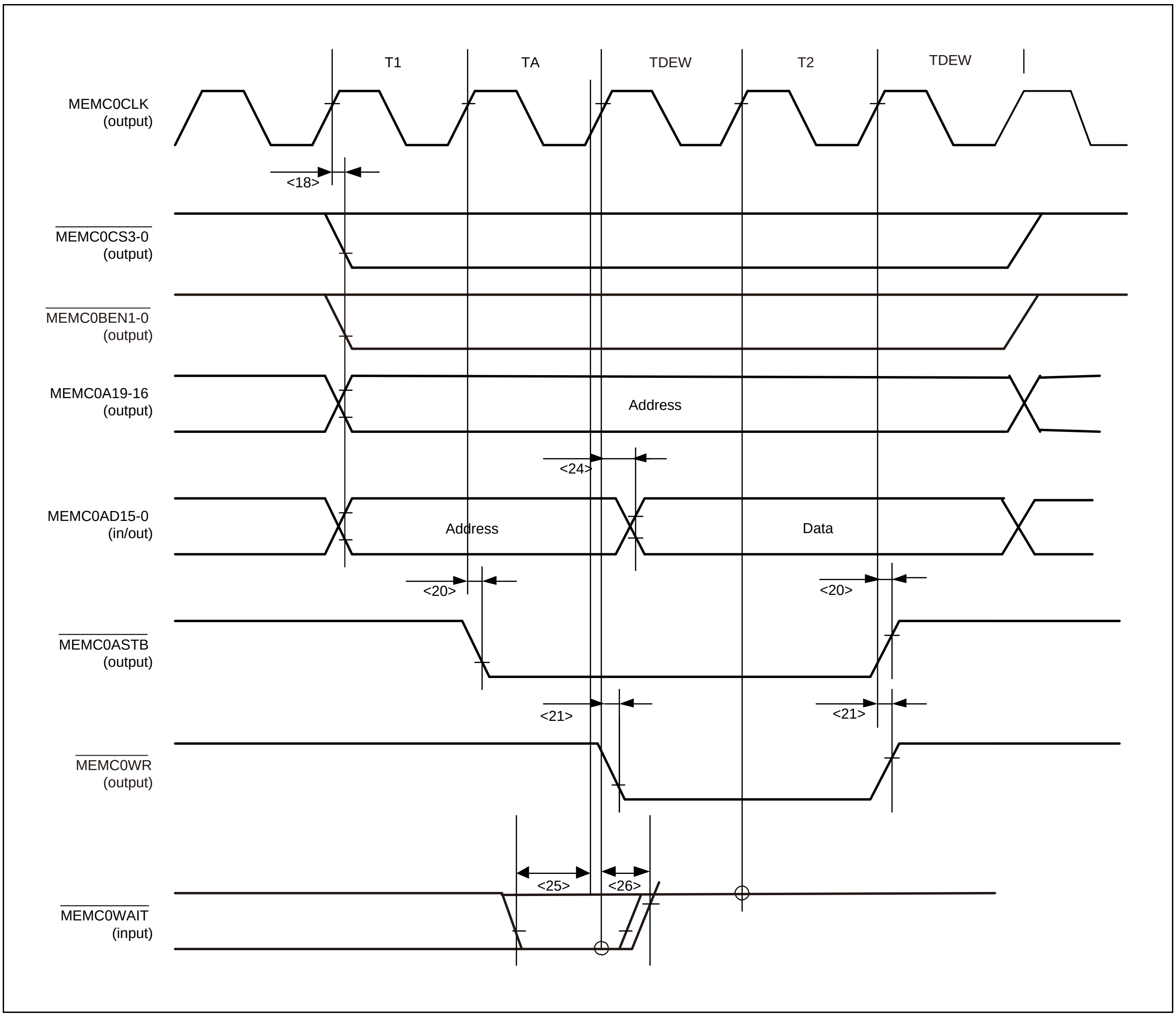
Note 1. Address means MEMC0AD15-0, MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$ and $\overline{\text{MEMC0BEN1-0}}$.
324-pin products support 24-bit address. 233/176-pin products support 23-bit address.

Note 2. t_{FKA} means the period from output off to Hi-z for MEMC0AD15-0.

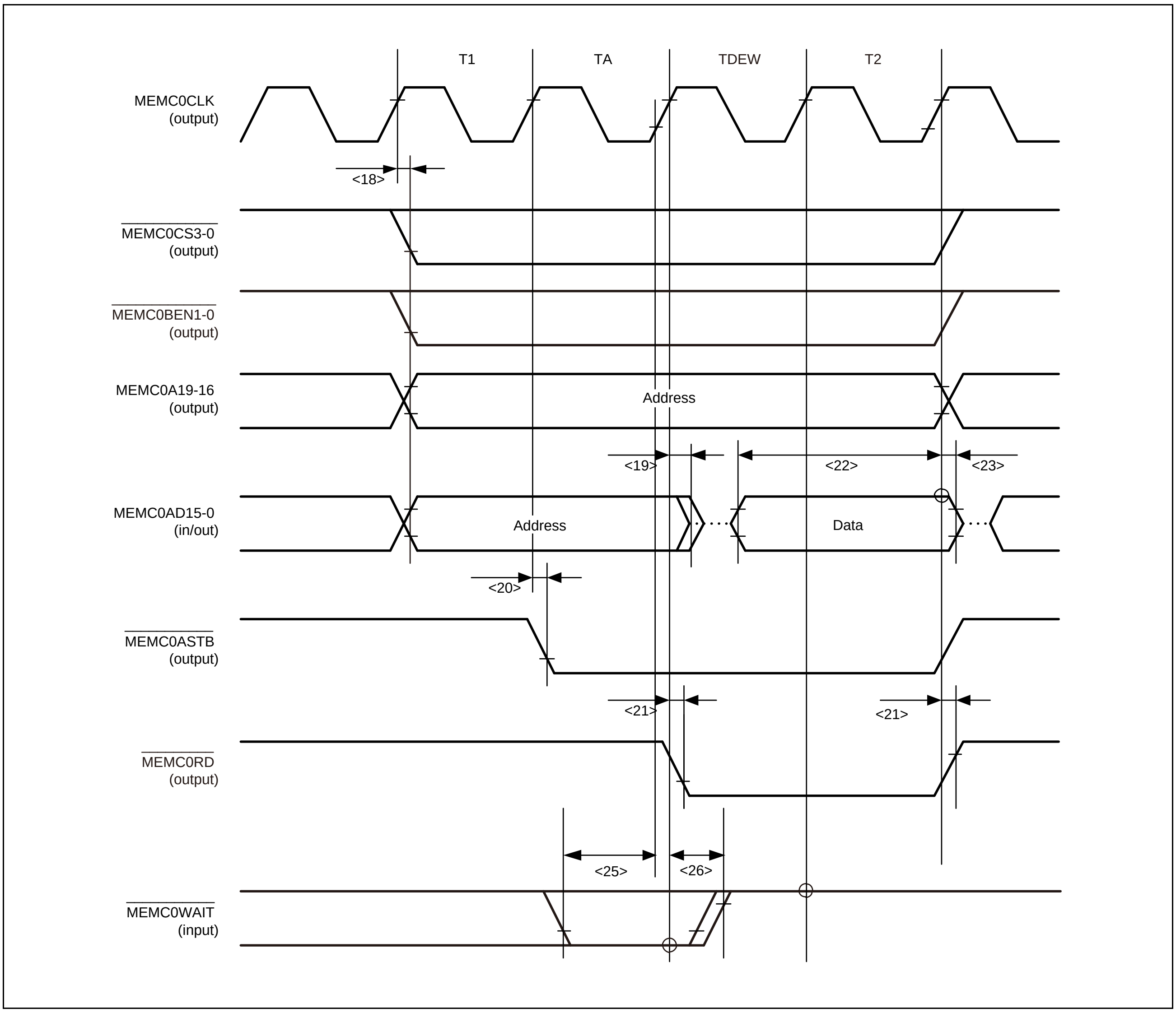
NOTE

When the bus period (T) is shorter than 44 ns, t_{DRDID} spec requires at least 1data wait. (w = 1)

(1) Multiplex Write Cycle (Synchronous; 1 Data Wait)



(2) Multiplex Read Cycle (Synchronous; 1 Data Wait)



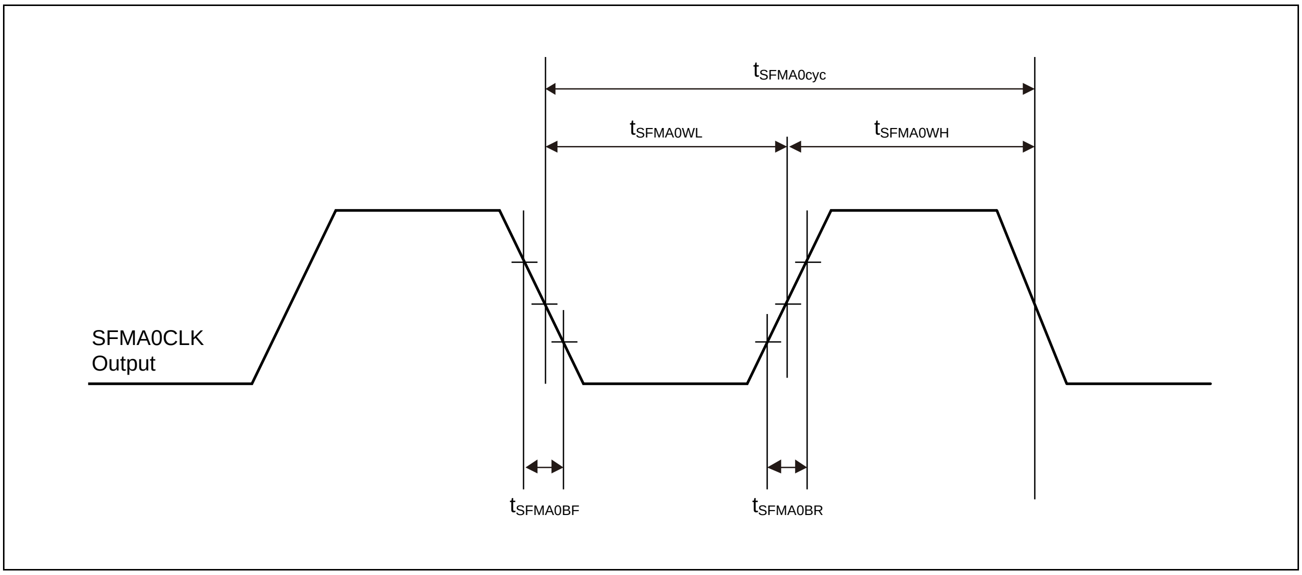
47A.5.8 SFMA Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to 3.6 V, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

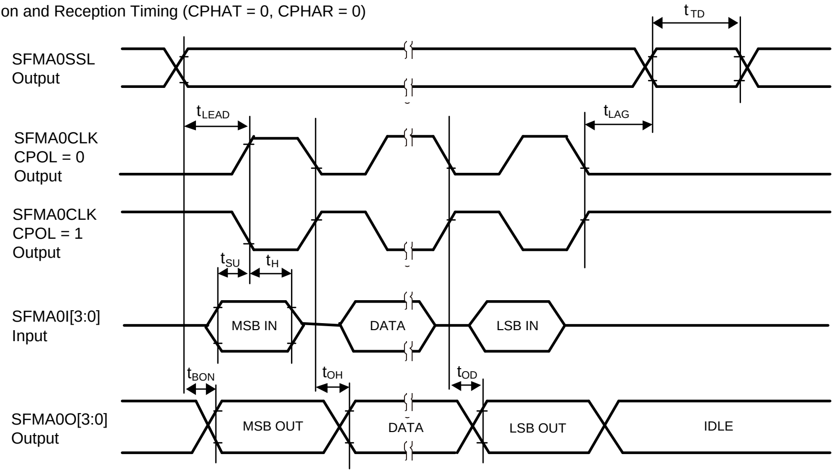
<Output driver strength>

SFMA0CLK, SFMA0SSL, and SFMA0O[3:0] pins: Fast mode

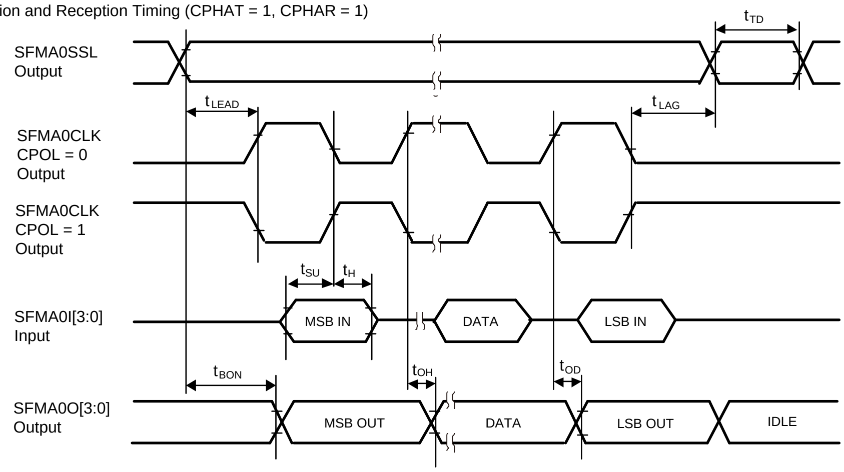
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SFMA0CLK clock cycle	$t_{SFMA0cyc}$		25			ns
SFMA0CLK high pulse width	$t_{SFMA0WH}$		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
SFMA0CLK low pulse width	$t_{SFMA0WL}$		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
SFMA0CLK rise time	t_{SFMA0R}				4.5	ns
SFMA0CLK fall time	t_{SFMA0F}				4.5	ns
Data input setup time	t_{SU}		13.0			ns
Data input hold time	t_{H}		0.0			ns
SFMA0SSL setup time	t_{LEAD}		1 × $t_{SFMA0cyc}$ – 5		8 × $t_{SFMA0cyc}$	ns
SFMA0SSL hold time	t_{LAG}		1.5 × $t_{SFMA0cyc}$		8.5 × $t_{SFMA0cyc}$ + 5	ns
Continuous transfer delay time	t_{TD}		1 × $t_{SFMA0cyc}$		8 × $t_{SFMA0cyc}$	ns
Data output delay time	t_{OD}				3.6	ns
Data output hold time	t_{OH}		–1.6			ns
Data output buffer on time	t_{BON}				3.6	ns
Data output buffer off time	t_{BOFF}		–7.0		0	ns

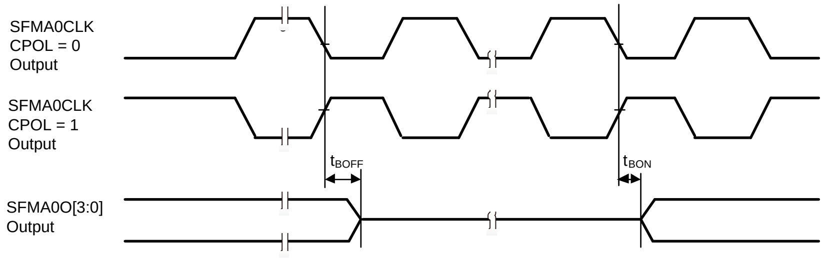


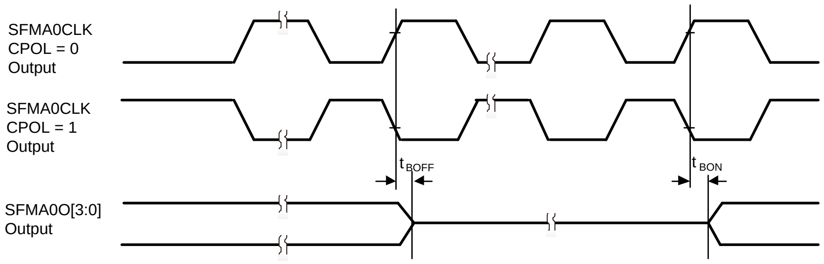
Transmission and Reception Timing (CPHAT = 0, CPHAR = 0)



Transmission and Reception Timing (CPHAT = 1, CPHAR = 1)




Timing for Switching the Buffers on and off (CPHAT = 0, CPHAR = 0)


Timing for Switching the Buffers on and off (CPHAT = 1, CPHAR = 1)

47A.5.9 MMCA Timing

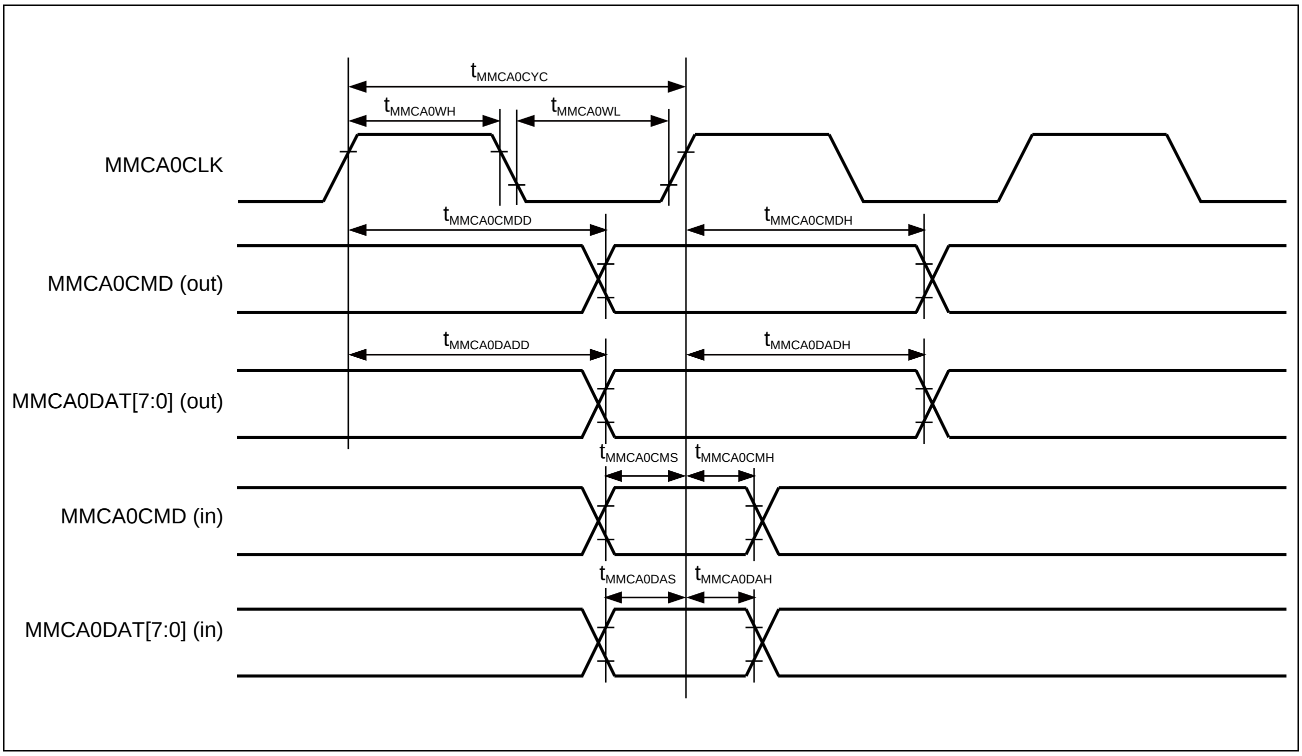
Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>

MMCA0CLK, MMCA0CMD and MMCA0DAT[7:0] pin: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MMCA0CLK clock cycle	$t_{MMCA0CYC}$		2 × t_{CKSCLK_IPERI2}			ns
MMCA0CLK high time	$t_{MMCA0WH}$		10			ns
MMCA0CLK low time	$t_{MMCA0WL}$		10			ns
MMCA0CMD output data delay time	$t_{MMCA0CMDD}$				$t_{MMCA0CYC}$ × 1/2 + 19	ns
MMCA0CMD output data hold time	$t_{MMCA0CMDH}$		4			ns
Data output delay time	$t_{MMCA0DADD}$				$t_{MMCA0CYC}$ × 1/2 + 19	ns
Data output hold time	$t_{MMCA0DADH}$		4			ns
MMCA0CMD input data setup time	$t_{MMCA0CMS}$		10			ns
MMCA0CMD input data hold time	$t_{MMCA0CMH}$		7			ns
Data input setup time	$t_{MMCA0DAS}$		10			ns
Data input hold time	$t_{MMCA0DAH}$		7			ns

Note: t_{CKSCLK_IPERI2} is period of CKSCLK_IPERI2.



47A.5.10 CSI Timing

47A.5.10.1 CSIG Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47A.12 CSIG Timing (Master Mode)

<Output driver strength>

CSIGnSO, CSIGnSC (output): Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYMGn}		100			ns
CSIGnSC high level width	t_{KWHMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSC low level width	t_{KWLMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSIMGn}		30			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSIMGn}		0			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOMGn}				7	ns
CSIGnRYI setup time (vs. CSIGnSC)	t_{SRYIGn}	CSIGnCTL1.CSIGnSIT = x CSIGnCTL1.CSIGnHSE = 1	2 × t_{KCYGn} + 25			ns
CSIGnRYI high level width	t_{WRYIGn}	CSIGnCTL1.CSIGnHSE = 1	t_{KCYGn} + 5			ns

Note: n = 0 to 4

Table 47A.13 CSIG Timing (Slave Mode)

<Output driver strength>

CSIGnSO: Fast mode

CSIGnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYSGn}		200			ns
CSIGnSC high level width	t_{KWHSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSC low level width	t_{KWLSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSISGn}		20			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSISGn}		t_{KCYGn} + 5			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOSGn}				30	ns
CSIGnRYO output delay	t_{SRYOGn}				38	ns
$\overline{\text{CSIGnSSI}}$ setup time (vs. CSIGnSC)	$t_{SSSISGn}$		0.5 × t_{KCYSGn} – 5			ns
$\overline{\text{CSIGnSSI}}$ hold time (vs. CSIGnSC)	$t_{HSSISGn}$		t_{KCYGn} + 5			ns

Note: n = 0 to 4

47A.5.10.2 CSIH Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47A.14 CSIH Timing (Master Mode: 10 Mbps)

<Output driver strength>

CSIHnSO, CSIHnSC (output): Fast mode (CL = 100pF@n = 0 / 50pF@n = 1-4)

CSIHnCSSx: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYMHn}		100			ns
CSIHnSC high level width	t_{KWHMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSC low level width	t_{KWLMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	19			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	14			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	0			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	t_{KCYHn}/2			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOMHn}				7	ns
CSIHnRYI setup time (vs. CSIHnSC)	t_{SRYIHn}	CSIHnCTL1.CSIHnSIT = x CSIHnCTL1.CSIHnHSE = 1	2 × t_{KCYHn} + 25			ns
CSIHnRYI high level width	t_{WRYIHn}	CSIHnCTL1.CSIHnHSE = 1	t_{KCYHn} + 5			ns
CSIHnCSS0-7 inactive width	$t_{WSCSBHn}$		CSIDLE × t_{KCYMHn} – 15			ns
CSIHnCSS0-7 setup time (vs. CSIHnSC)	$t_{SSCSBHn0}$	CSIHnCFGx.CSIHnDAP = 0	CSSETUP × t_{KCYMHn} – 23			ns
	$t_{SSCSBHn1}$	CSIHnCFGx.CSIHnDAP = 1	(CSSETUP + 0.5) × t_{KCYMHn} – 23			ns
CSIHnCSS0-7 hold time (vs. CSIHnSC)	$t_{HSCSBHn0}$	CSIHnCTL1.CSIHnSIT = 0	CSSHOLD × t_{KCYMHn} – 5			ns
	$t_{HSCSBHn1}$	CSIHnCTL1.CSIHnSIT = 1	(CSSHOLD + 0.5) × t_{KCYMHn} – 5			ns

Note: n = 0 to 4

NOTE

CSIDLE: Setting value of CSIHnCFGx.CSIHnIDx[2:0]
CSSETUP: Setting value of CSIHnCFGx.CSIHnSPx[3:0]
CSSHOLD: Setting value of CSIHnCFGx.CSIHnHDx[3:0]
x: Depends on number of the chip select signals.

CAUTION

When the serial clock level is changed during the communication (CSIHnCFGx.CSIHnCKPx) and the IDLE has a setting of 0.5 transmission clock cycles, an inactive width time $t_{WSCSBHn}$ of "0.5 × t_{KCYMHn}" is added.

Table 47A.15 CSIH Timing (Slave Mode: 5 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		200			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		20			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				30	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			38	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$38 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 0 to 4

Table 47A.16 CSIH Timing (Slave Mode: 8 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

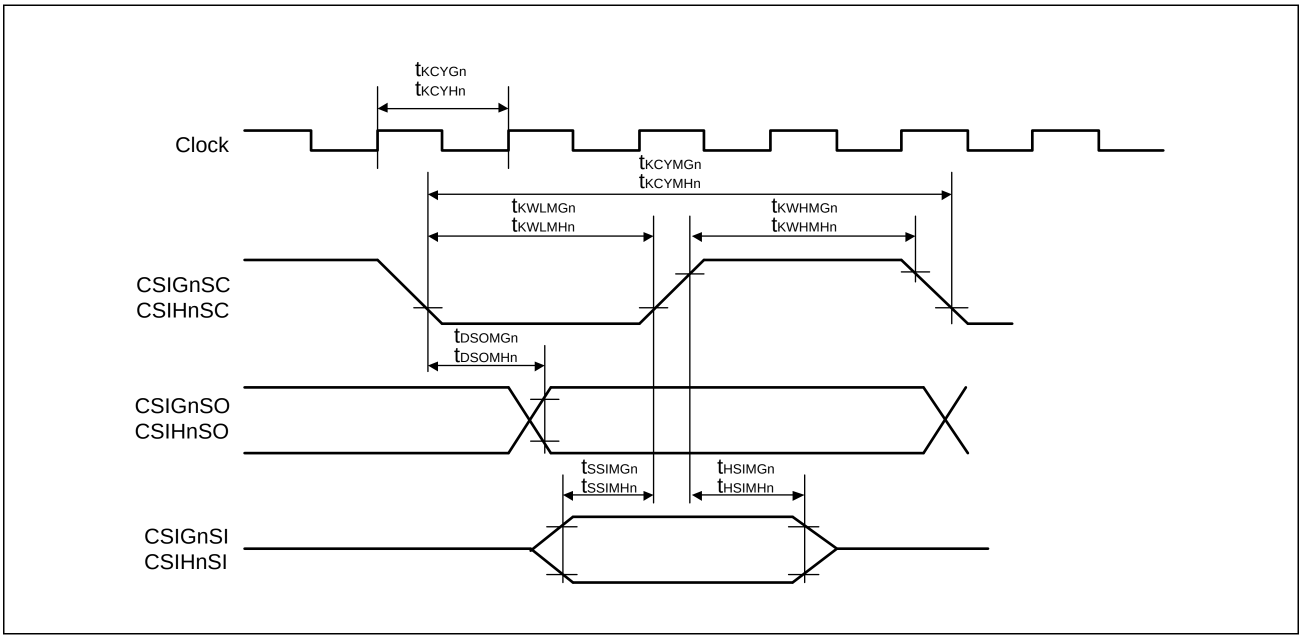
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		125			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		12.5			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				25	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			27	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$27 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 2 (Only for CSIH2)

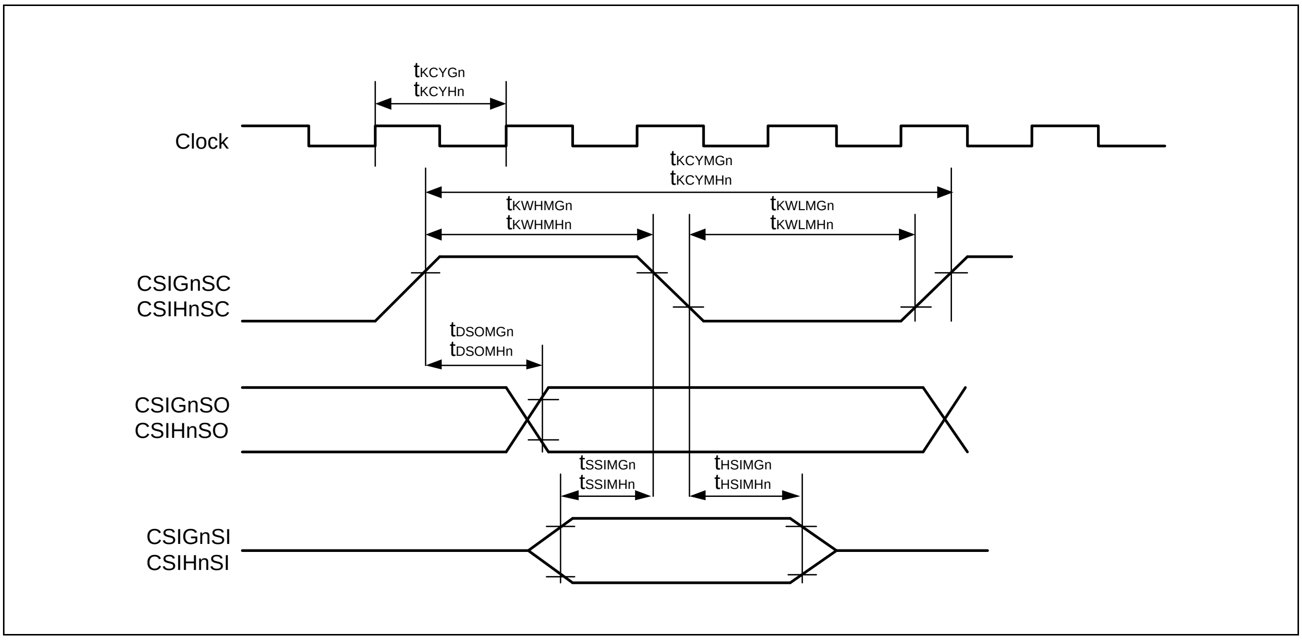
(1) SC/SI/SO

Master mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

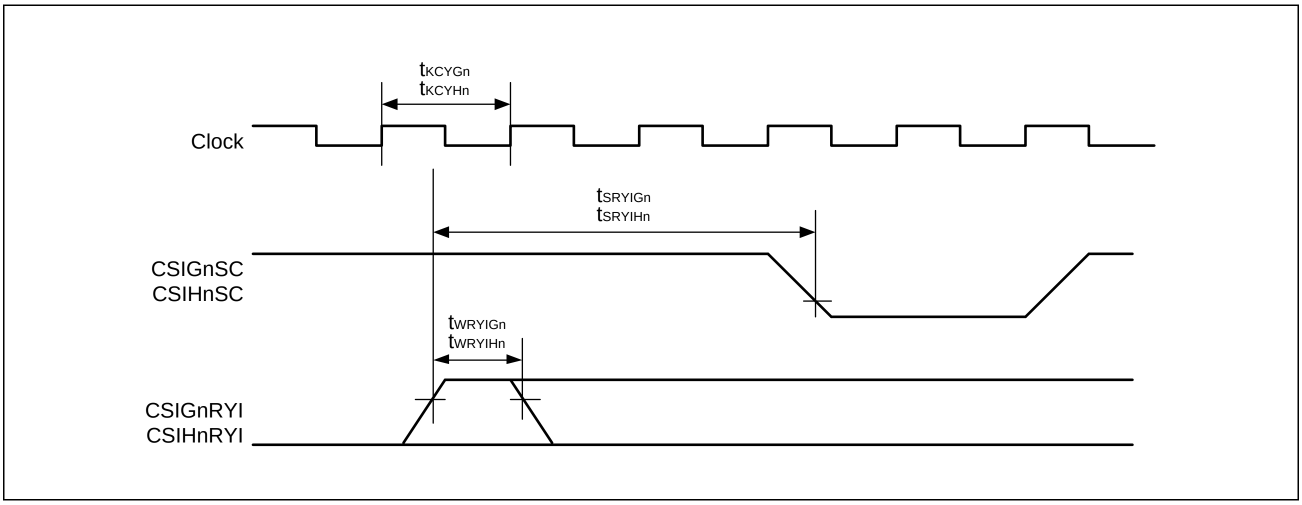


(2) RYI

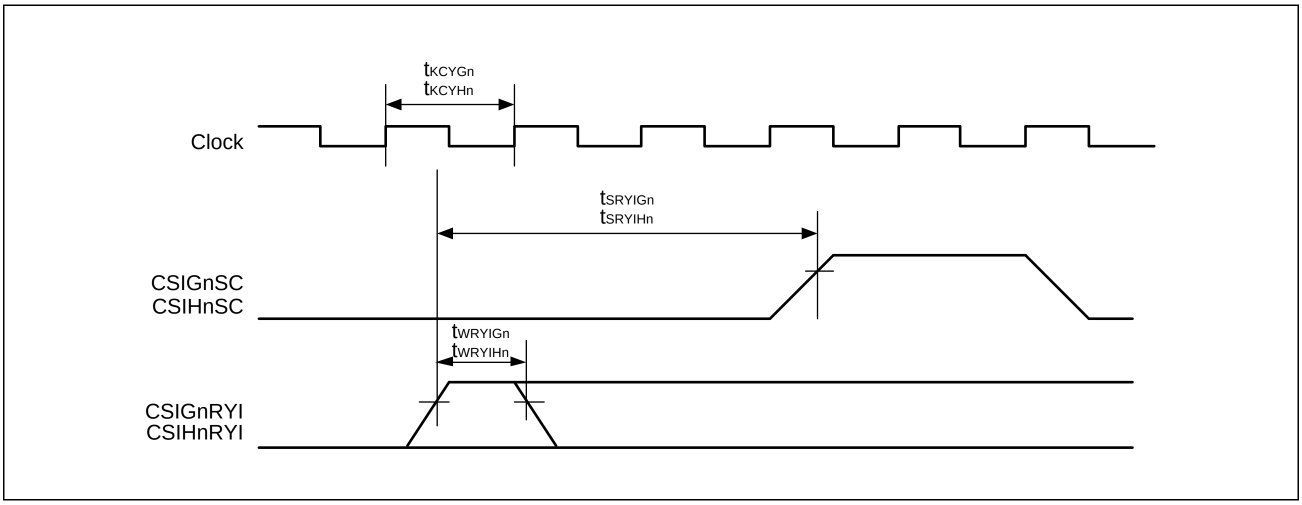
Master mode:

- CSIG: Only master mode (CSIGnCTL1: CSIGnHSE = 1, CSIGnCTL1: CSIGnSIT = 0)
- CSIH: Only master mode (CSIHnCTL1: CSIHnHSE = 1, CSIHnCTL1: CSIHnSIT = 0)

  - CSIG (CSIGnCTL1: CSIGnCKR = 0)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 0)



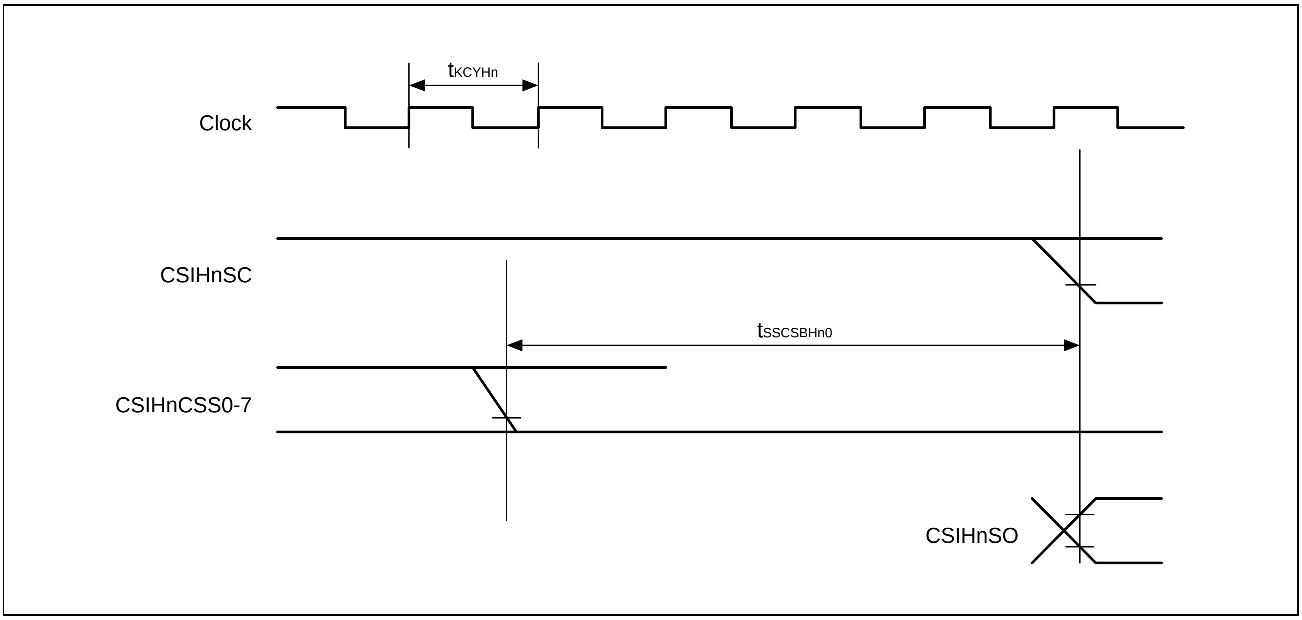
  - CSIG (CSIGnCTL1: CSIGnCKR = 1)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 1)



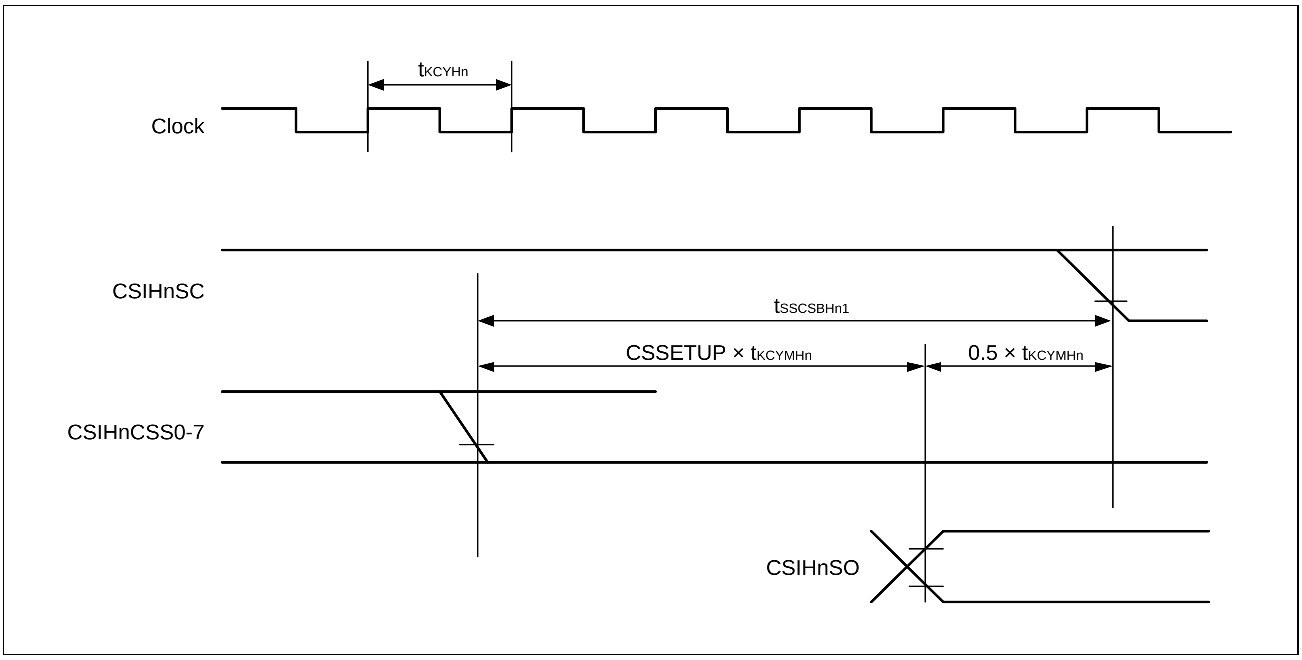
(3) CSSx

Only master mode (setup time):

- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0

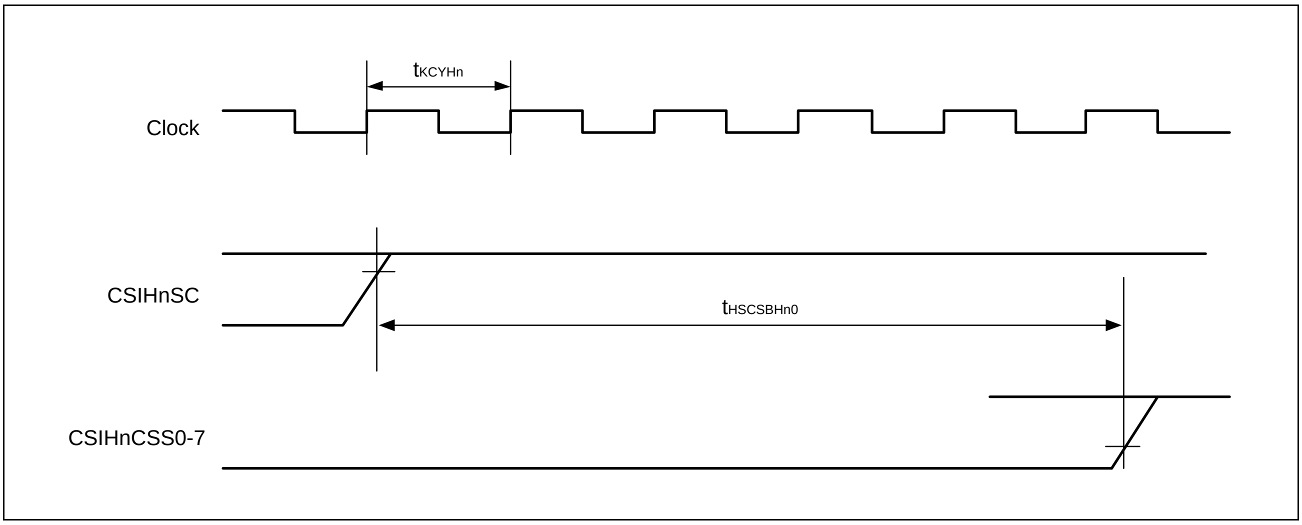


- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 1

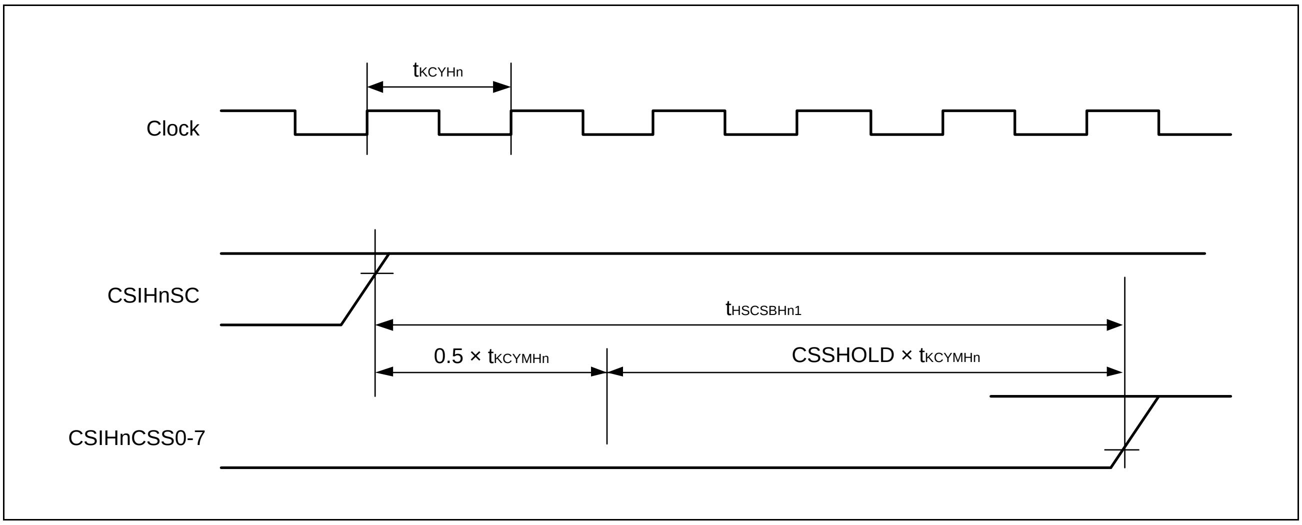


Only master mode (hold time):

- CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



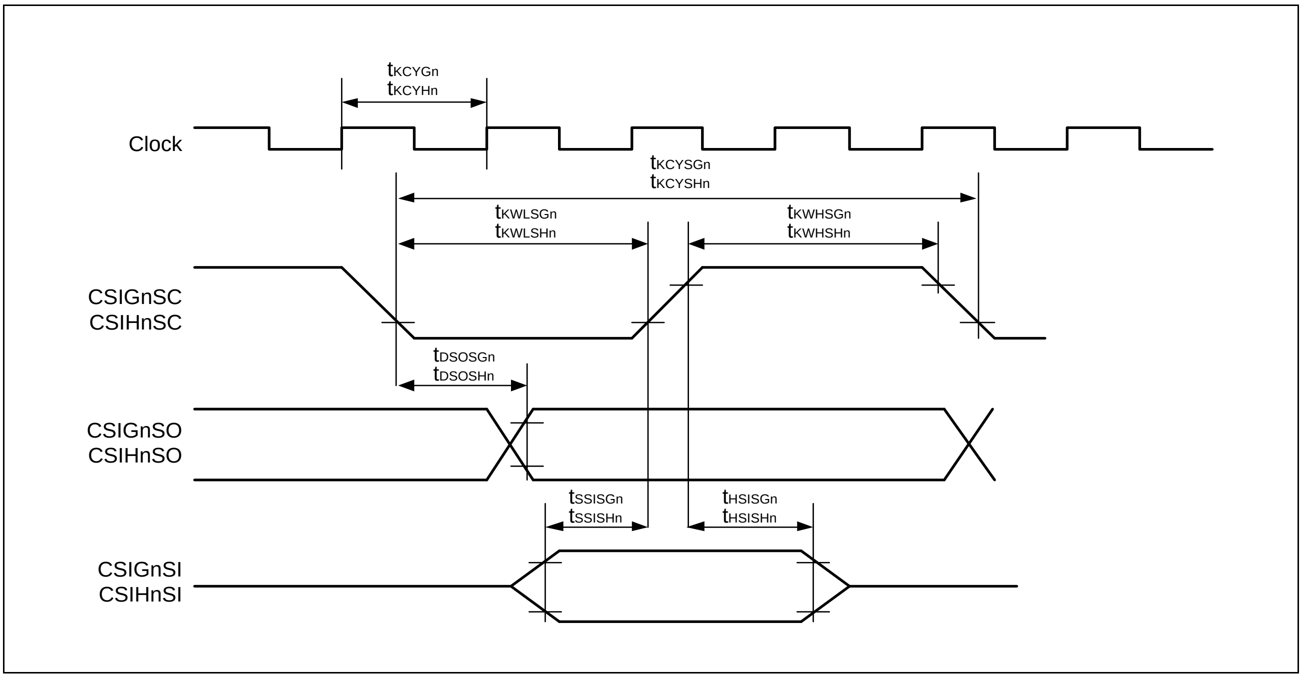
- CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



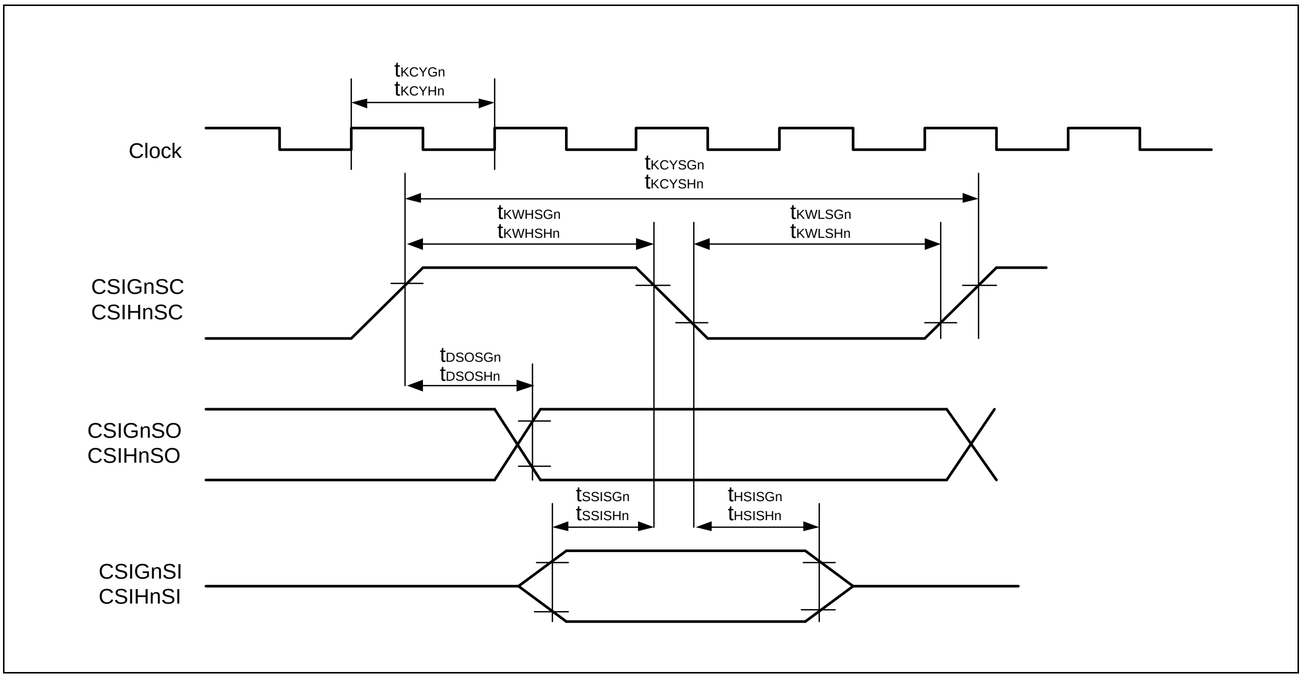
(4) SC/SI/SO

Slave mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)

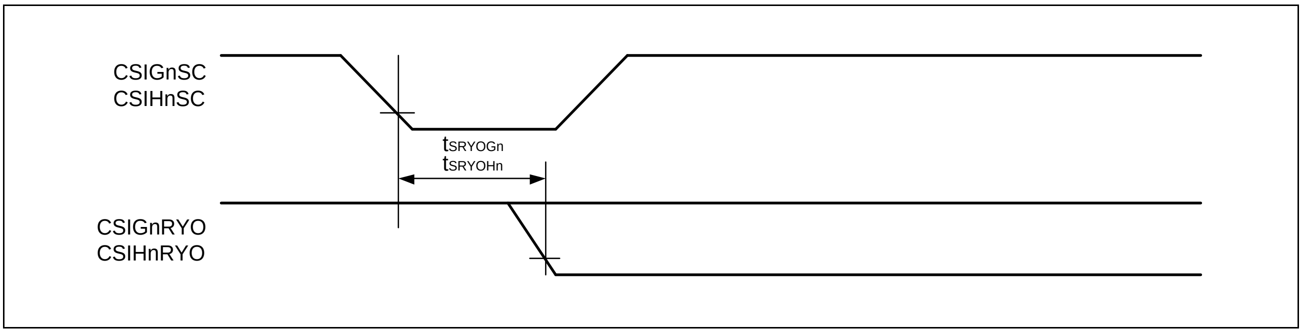


- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

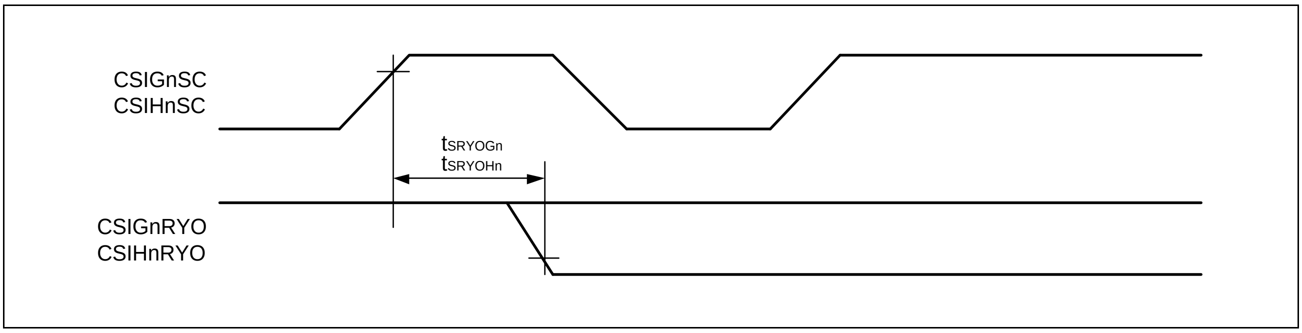


(5) RYO

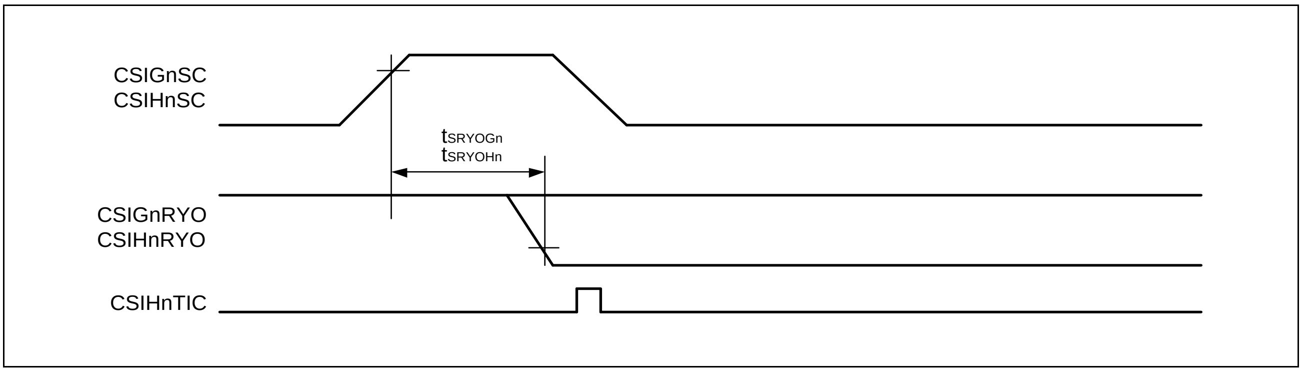
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0)



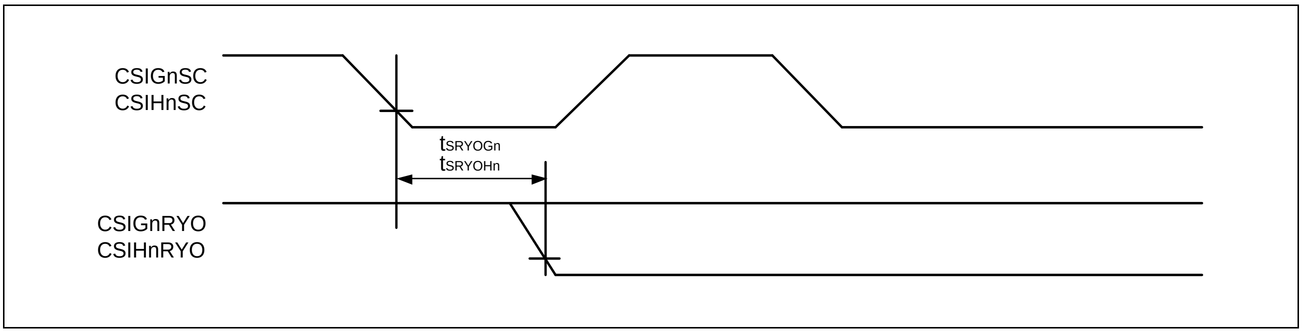
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0)



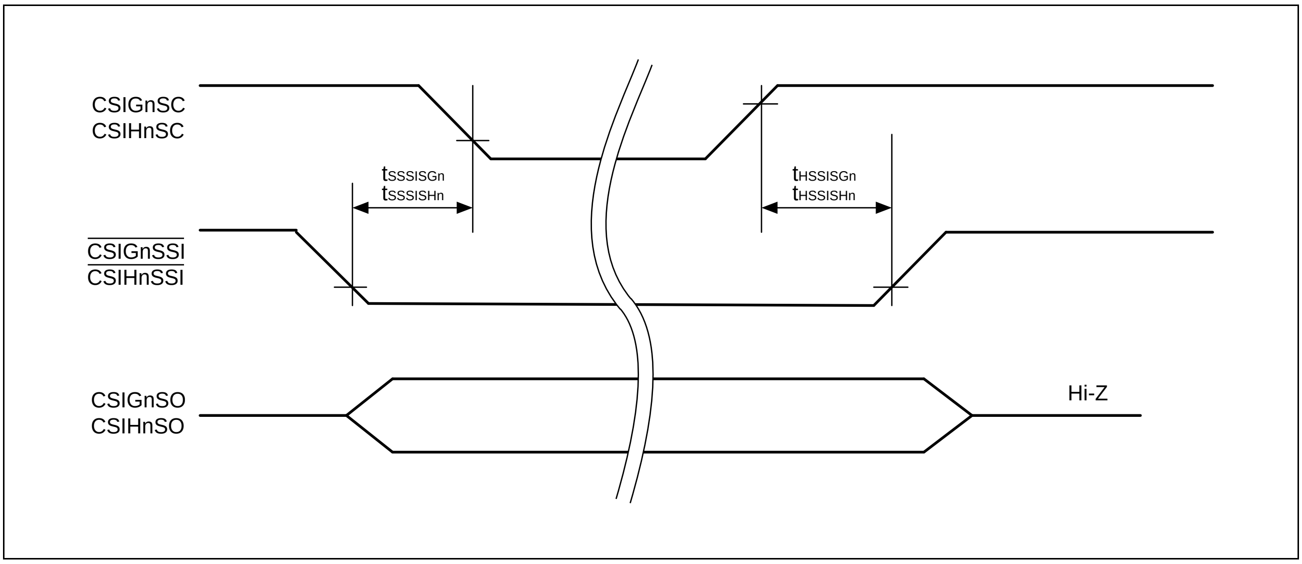
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/1)



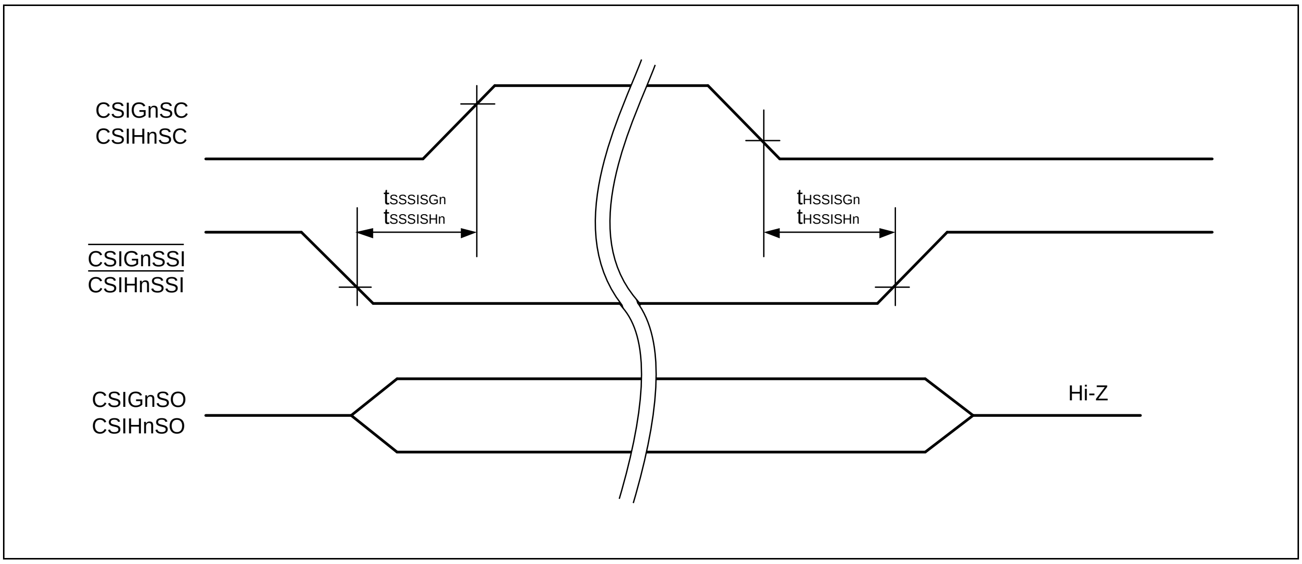
(6) SSI

Slave mode:

- CSIG (CSIGnCTL1: CSIGnSSE = 1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCTL1: CSIHnSSE = 1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnSSE = 1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCTL1: CSIHnSSE = 1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)



47A.5.11 RLIN2/RLIN3 Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RLIN3 transfer rate		LIN specification	1		20	kbps
		LIN extended baud rate	1		115.2*[1]	kbps
		UART function			1.5	Mbps
RLIN2 transfer rate		LIN specification	1		20	kbps

Note 1. The LIN extended baud rate is not part of the LIN standard specification.

47A.5.12 RIIC Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C

Table 47A.17 RIIC Timing (Normal Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				100	kHz
Bus free time (between stop/start condition)	t_{BUF}		4.7			μs
Hold time*[1]	t_{HD}: STA		4.0			μs
RIICnSCL clock low-level width	t_{LOW}		4.7			μs
RIICnSCL clock high-level time	t_{HIGH}		4.0			μs
Setup time for start/restart condition	t_{SU}: STA		4.7			μs
Data hold time	t_{HD}: DAT	CBUS compatible master	5.0			μs
		I^{2}C mode	0*[2]			μs
Data setup time	t_{SU}: DAT		250			ns
Stop condition setup time	t_{SU}: STO		4.0			μs
Capacitance load of each bus line	Cb				400	pF

Remark: n = 0, 1

Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Table 47A.18 RIIC Timing (Fast Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				400	kHz
Bus free time (between stop/start condition)	t_{BUF}		1.3			µs
Hold time*1	t_{HD}: STA		0.6			µs
RIICnSCL clock low-level width	t_{LOW}		1.3			µs
RIICnSCL clock high-level time	t_{HIGH}		0.6			µs
Setup time for start/restart condition	t_{SU}: STA		0.6			µs
Data hold time	t_{HD}: DAT	I^{2}C mode	0*2			µs
Data setup time	t_{SU}: DAT		100*3			ns
Stop condition setup time	t_{SU}: STO		0.6			µs
Pulse width with spike suppressed by input filter	t_{SP}		0		50	ns
Capacitance load of each bus line	Cb				400	pF

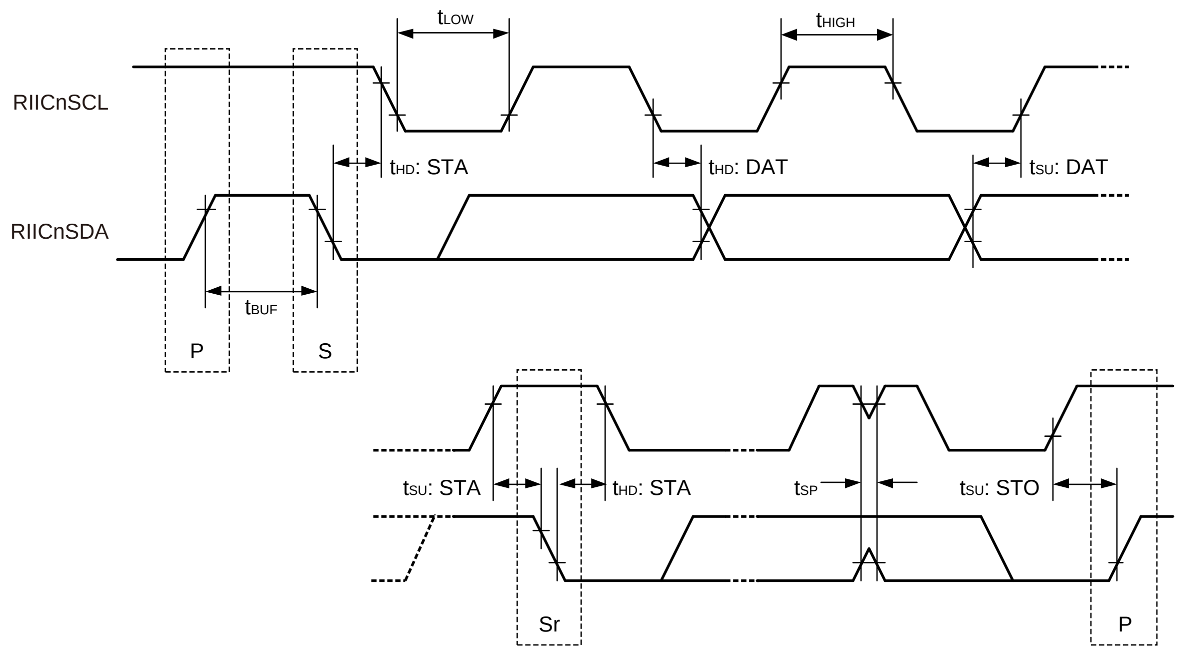
Remark: n = 0, 1

Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Note 3. The fast mode I^{2}C bus can be used in normal mode I^{2}C bus system. In this case, set the fast mode I^{2}C bus so that it meets the following conditions.
- If the system does not extend the RIICnSCL signal's low state hold time: t_{SU}: DAT ≥ 250 ns
- If the system extends the RIICnSCL signal's low state hold time:
Transmit the following data bit to the RIICnSDA line prior to releasing the RIICnSCL line
(1250 ns: Normal mode I^{2}C bus specification).



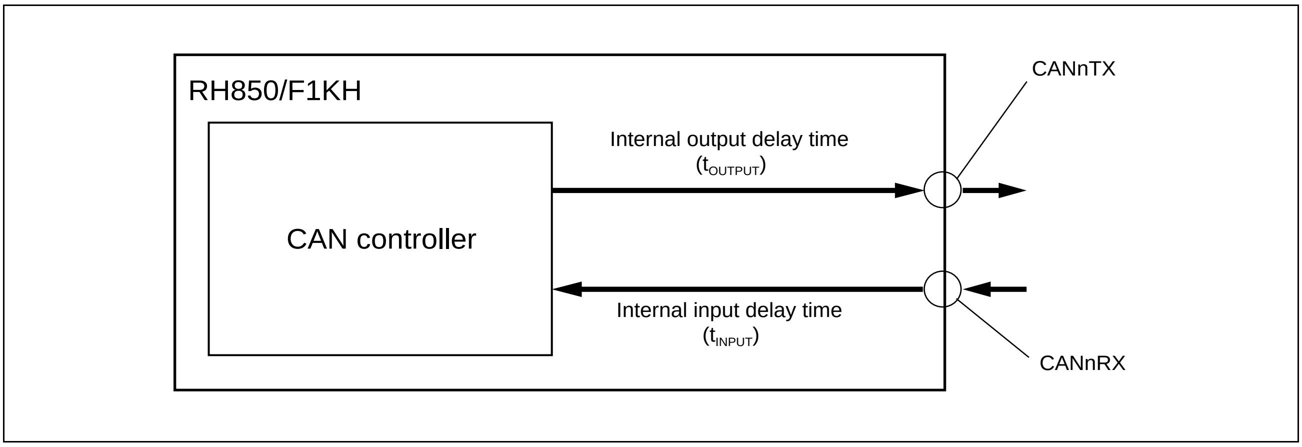
Note: P: Stop condition S: Start condition Sr: Restart condition

47A.5.13 RS-CANFD Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate		Classical CAN mode			1	Mbps
Data bit rate (CAN FD mode)		Nominal bit rate ≤ 500 kbps			5	Mbps
		Nominal bit rate > 500 kbps			2	Mbps
Internal delay time*1	t_{NODE}				50	ns

Note 1. t_{NODE} = Internal input delay time (t_{INPUT}) + Internal output delay time (t_{OUTPUT})



47A.5.14 FlexRay Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Products of CPU frequency 240 MHz max.

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					10	Mbps

Products of CPU frequency 160 MHz max.

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					5	Mbps

47A.5.15 Ethernet Timing

47A.5.15.1 MII Interface

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to 3.6 V, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 15 pF

<Output driver strength>

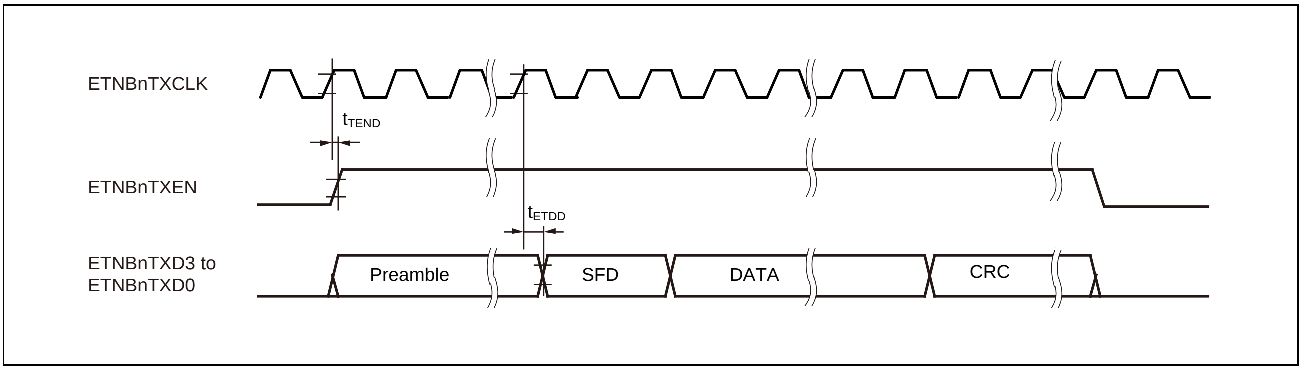
ETNBnTXD3-0 and ETNBnTXEN pins: Fast mode

ETNBnTXCLK pin: TTL type

Table 47A.19 MII Interface (Transmission Interface)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNBnTXCLK clock period	t_{Tcyc}	100 Mbps	40 – 100 ppm	40	40 + 100 ppm	ns
		10 Mbps	400 – 100 ppm	400	400 + 100 ppm	ns
ETNBnTXEN delay vs ETNBnTXCLK ↑	t_{TEND}	CL = 15 pF			18	ns
ETNBnTXD[3:0] delay vs ETNBnTXCLK ↑	t_{ETDD}	CL = 15 pF			18	ns

Note: n = 0, 1



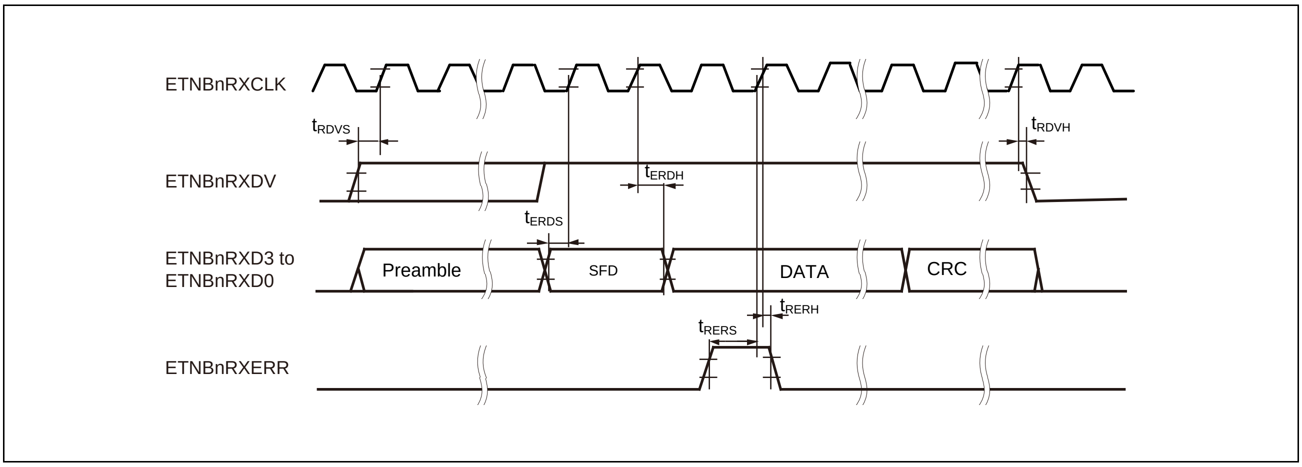
<Input buffer>

ETNBnRXCLK, ETNBnRXDV, ETNBnRXD[3:0], and ETNBnRXER pins: TTL type

Table 47A.20 MII Interface (Reception Interface)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNBnRXCLK clock period	t_{Rcyc}	100 Mbps	40 – 100 ppm	40	40 + 100 ppm	ns
		10 Mbps	400 – 100 ppm	400	400 + 100 ppm	ns
ETNBnRXDV hold time vs ETNBnRXCLK ↑	t_{RDVH}		10			ns
ETNBnRXDV setup time vs ETNBnRXCLK ↑	t_{RDVS}		10			ns
ETNBnRXD[3:0] hold time vs ETNBnRXCLK ↑	t_{ERDH}		10			ns
ETNBnRXD[3:0] setup time vs ETNBnRXCLK ↑	t_{ERDS}		10			ns
ETNBnRXERR hold time vs ETNBnRXCLK ↑	t_{RERH}		10			ns
ETNBnRXERR setup time vs ETNBnRXCLK ↑	t_{RERS}		10			ns

Note: n = 0, 1



47A.5.15.2 Management Interface

Timing of management interface (ETNBnMDC and ETNBnMDIO) depends on software. It is necessary to adjust wait time according to AC specification of PHY.

47A.5.16 RSENT Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Tick Time			1		90	µs

47A.5.17 Timer Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

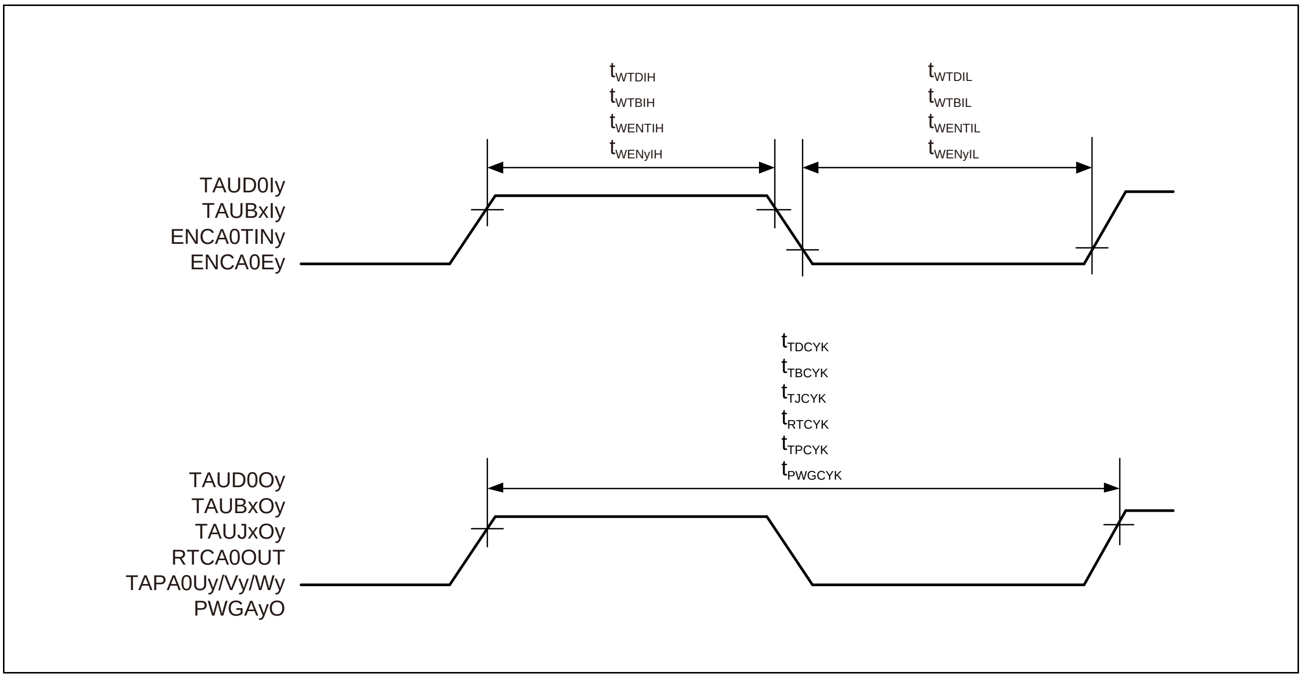
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TAUD0Iy input high/low level width (y = 0 to 15)	t_{WTDIH}/ t_{WTDIL}		n × Tsamp + 20*1,*2			ns
TAUD0Oy output cycle (y = 0 to 15)	t_{TDCYK}	Slow mode			10	MHz
TAUBxIy input high/low level width (x = 0, 1, y = 0 to 15)	t_{WTBIH}/ t_{WTBIL}		n × Tsamp + 20*1,*2			ns
TAUBxOy output cycle (x = 0, 1, y = 0 to 15)	t_{TBCYK}	Slow mode			10	MHz
TAUJxIy input high/low level width*3 (x = 0 to 3, y = 0 to 3)	t_{WTJIH}/ t_{WTJIL}		600			ns
TAUJxIy pulse rejection*4	t_{WTIJRJ}		100			ns
TAUJxOy output cycle (x = 0 to 3, y = 0 to 3)	t_{TJCYK}	Slow mode			10	MHz
RTCA0OUT output cycle	t_{RTCYK}			1		Hz
TAPA0ESO input high/low level width*3	t_{WESIH}/ t_{WESIL}		600			ns
TAPA0ESO pulse rejection*4	t_{WESIRJ}		100			ns
TAPA0Uy/Vy/Wy output cycle (y = P, N)	t_{TPCYK}	Slow mode			10	MHz
ENCA0TINy input high/low level width (y = 0, 1)	t_{WENTIH}/ t_{WENTIL}		n × Tsamp + 20*1			ns
ENCA0Ey input high/low level width (y = 0, 1, C)	t_{WENyIH}/ t_{WENyIL}		n × Tsamp + 20*1			ns
PWGAyO output cycle (y = 0 to 95)	t_{PWGCYK}	Slow mode			10	MHz

Note 1. n: Sampling number of the digital noise filter for each input.
Tsamp: Sampling time of the digital noise filter for each input.

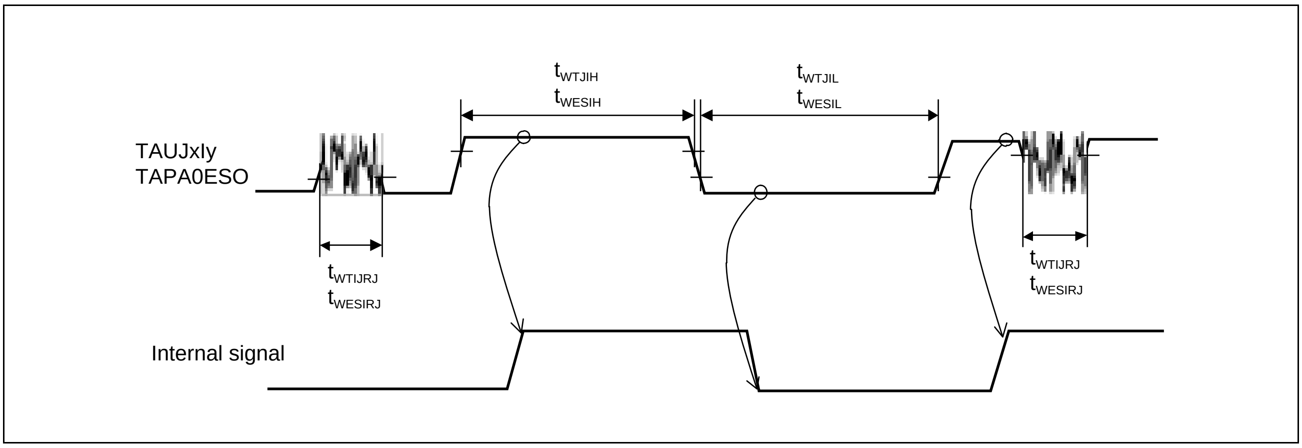
Note 2. Input more than 1 count clock width of each timer counter channel.

Note 3. TAUJxIy and TAPA0ESO input width is needed to ensure that the internal timer input signal is activated.

Note 4. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



t_{WTDIH}
t_{WTBIH}
t_{WENTIH}
t_{WENyIH}
t_{WTDIL}
t_{WTBIL}
t_{WENTIL}
t_{WENyIL}
TAUD0Iy
TAUBxIy
ENCA0TINy
ENCA0Ey
t_{TDCYK}
t_{TBCYK}
t_{TJCYK}
t_{RTCYK}
t_{TPCYK}
t_{PWGCYK}
TAUD0Oy
TAUBxOy
TAUJxOy
RTCA0OUT
TAPA0Uy/Vy/Wy
PWGAyO



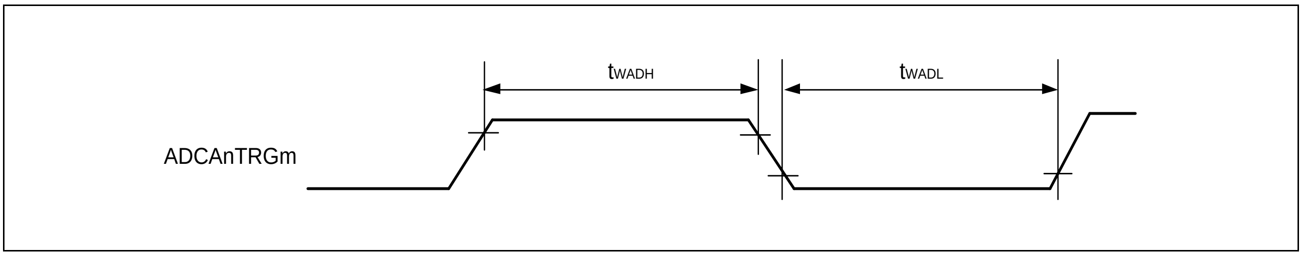
t_{WTJIH}
t_{WESIH}
t_{WTJIL}
t_{WESIL}
TAUJxIy
TAPA0ESO
t_{WTIJRJ}
t_{WESIRJ}
t_{WTIJRJ}
t_{WESIRJ}
Internal signal

47A.5.18 ADTRG Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ADCAnTRGm input high/ low level width	t_{WADH}/t_{WADL}		k × Tsamp + 20*[1]			ns

Note 1. k: Sampling number of the digital noise filter for each input.
Tsamp: Sampling time of the digital noise filter for each input.



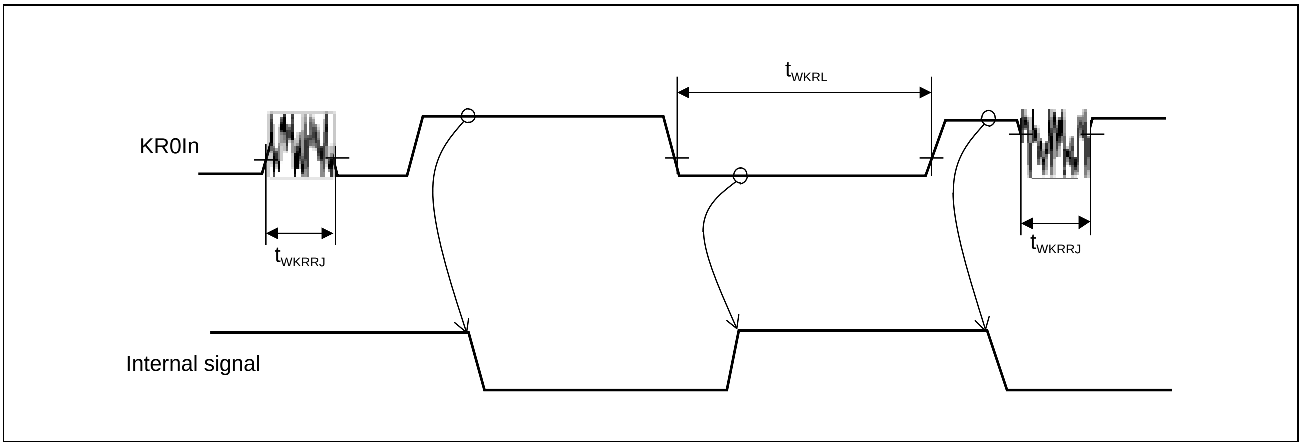
47A.5.19 Key Return Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
KR0In input low level width*[1]	t_{WKRL}		600			ns
KR0In pulse rejection*[2]	t_{WKRRJ}		100			ns

Note 1. KR0In input width is needed to ensure that the internal key input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



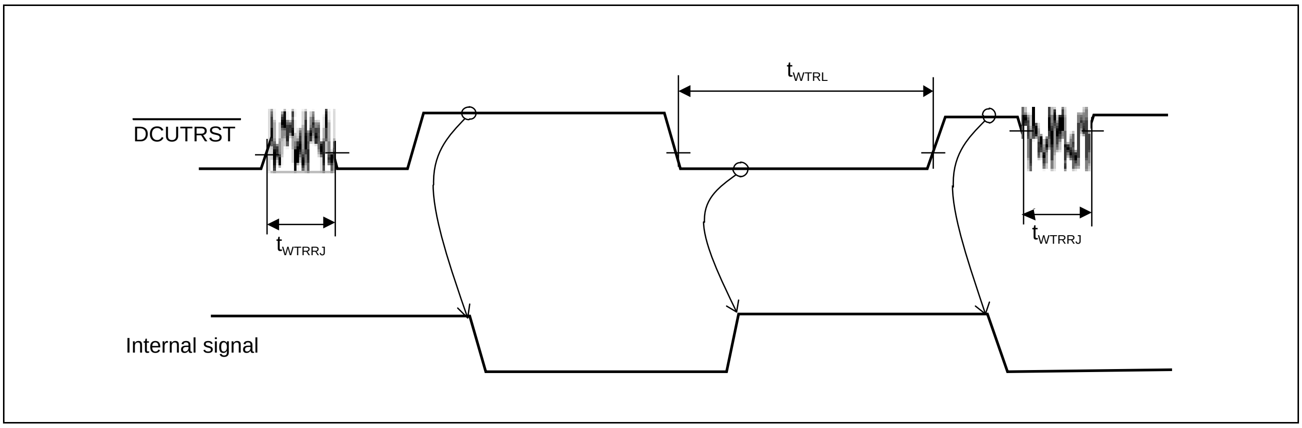
47A.5.20 $\overline{\text{DCUTRST}}$ Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{DCUTRST}}$ input low level width*[1]	t_{WTRL}		600			ns
$\overline{\text{DCUTRST}}$ pulse rejection*[2]	t_{WTRRJ}		100			ns

Note 1. $\overline{\text{DCUTRST}}$ input width is needed to ensure that the internal DCU reset input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



47A.5.21 Debug Interface Characteristics

47A.5.21.1 Nexus Interface Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

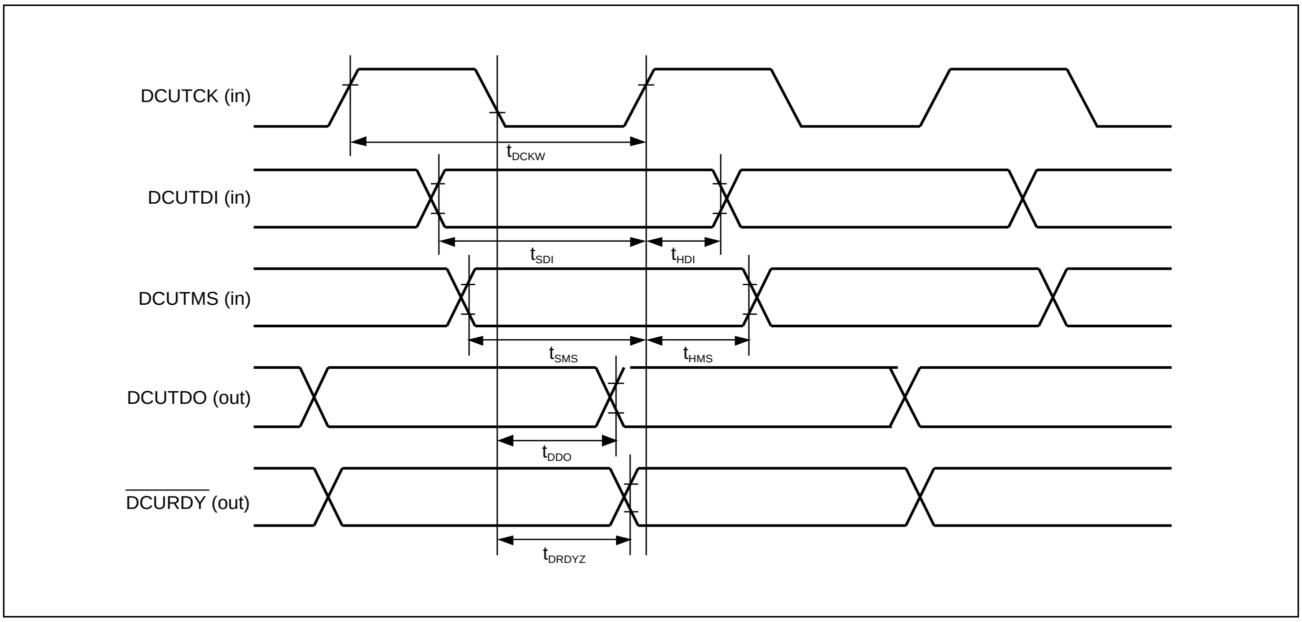
<Input buffer>

DCUTDI, DCUTCK, DCUTMS, $\overline{\text{DCUTRST}}$: TTL

<Output driver strength>

DCUTDO, $\overline{\text{DCURDY}}$: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DCUTCK cycle width	t_{DCKW}		50			ns
DCUTDI setup time (vs DCUTCK ↑)	t_{SDI}		12			ns
DCUTDI hold time (vs DCUTCK ↑)	t_{HDI}		3			ns
DCUTMS setup time (vs DCUTCK ↑)	t_{SMS}		12			ns
DCUTMS hold time (vs DCUTCK ↑)	t_{HMS}		3			ns
DCUTDO delay time (↓ DCUTCK)	t_{DDO}		0		20	ns
$\overline{\text{DCURDY}}$ delay time (↓ DCUTCK)	t_{RDYZ}		0		20	ns



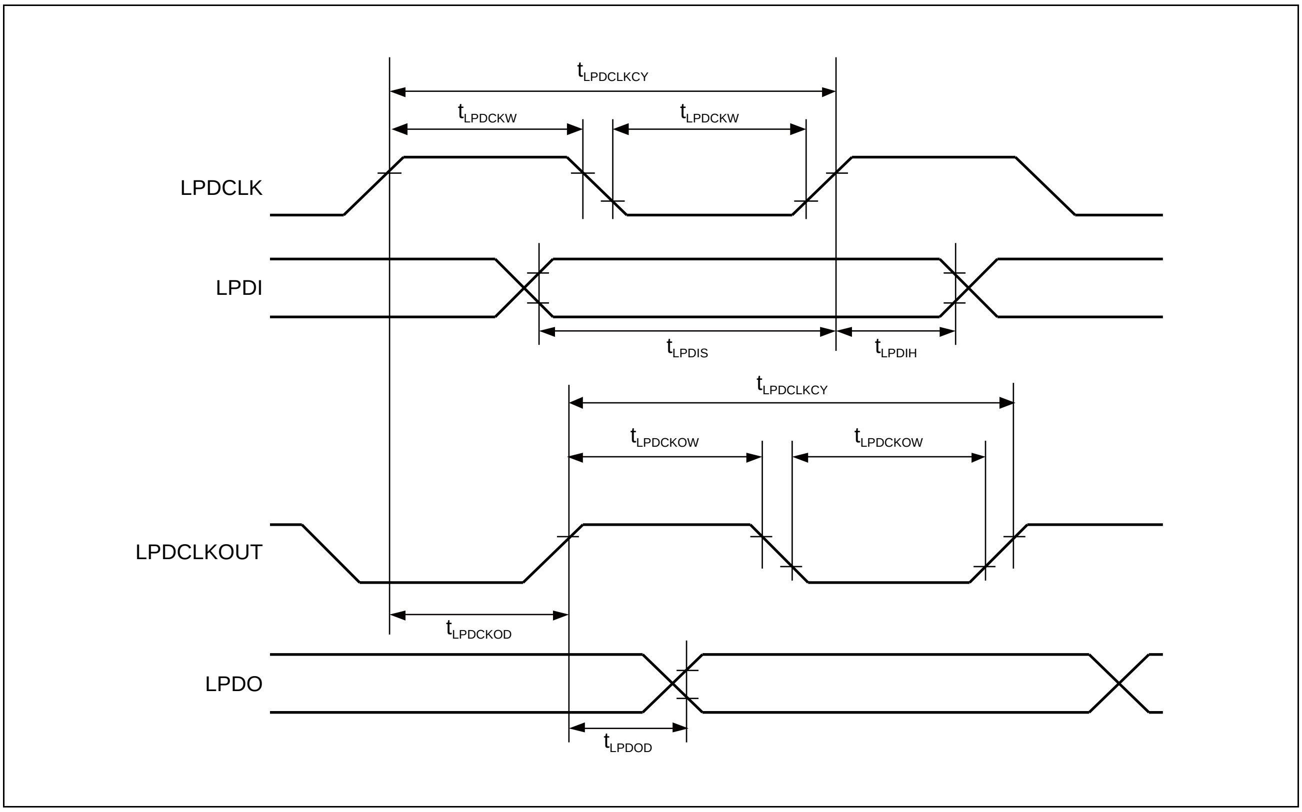
47A.5.21.2 LPD (4 Pins) Interface Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 100 pF

<Input buffer>
LPDCLK, LPDI: TTL

<Output driver strength>
LPDCLKOUT, LPDO: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPDCLK cycle time/ LPDCLKOUT cycle time	$t_{LPDCLKCY}$		83.3 (max.12 MHz)			ns
LPDCLK High-level width/ LPDCLK Low-level width	t_{LPDCKW}		0.5 × $t_{LPDCLKCY}$ – 10			ns
LPDCLKOUT High-level width/ LPDCLKOUT low-level width	$t_{LPDCKOW}$		t_{LPDCKW} – 10			ns
LPDI setup time (LPDCLK ↑)	t_{LPDIS}		41			ns
LPDI hold time (LPDCLK ↑)	t_{LPDIH}		3			ns
LPDCLK to LPDCLKOUT delay time	$t_{LPDCKOD}$				44	ns
LPDO delay time (LPDCLKOUT ↑)	t_{LPDOD}		0		15	ns



47A.5.21.3 LPD (1 Pin) Interface Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 50 pF

<Input buffer>
LPDIO: TTL

<Output driver strength>
LPDIO: Fast mode

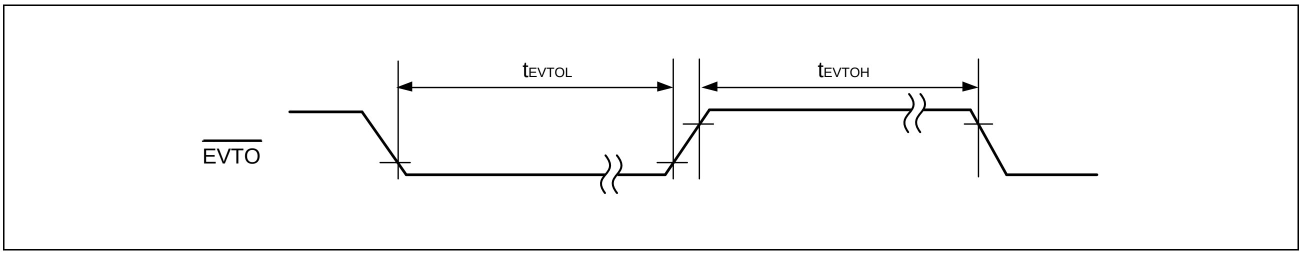
<External pull-up resistor>
LPDIO: 4.7 kΩ to 10 kΩ

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPD (1 pin) baud rate					2.0	Mbps

47A.5.21.4 Debug Event Interface Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 50 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{EVTO}}$ high/low level width	t_{EVTOH}/t_{EVTOL}		50			ns



47A.6 A/D Converter Characteristics

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

(1/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Conversion clock	ADCLKn				8[*3]		40	MHz
Resolution	RESn	12-bit mode			12	12	12	bit
		10-bit mode			10	10	10	bit
Conversion time	t_{CONn}	ADCAnSMPCR.SMPT[7:0] = 12 H (40 cycle) (8M Hz[*3] ≤ ADCLKn ≤ 32 MHz), External MPX is not used			1.25		5	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (46 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz), External MPX is not used			1.15		5.75	µs
		ADCAnSMPCR.SMPT[7:0]=12 H (80 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 32 MHz), External MPX is used			2.5[*4]		10	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (92 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz), External MPX is used			2.3[*4]		11.5	µs
Sampling time	t_{SMP}	ADCAnSMPCR.SMPT[7:0] = 12 H (18 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 32 MHz)			0.56		2.25	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (24cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz)			0.6		3	µs
Overall error[*1]	TOEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±4.0	LSB
				ADCA0I0-5 (T&H used)			±6.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±6.0	LSB
				ADCA0I0-5 (T&H used)			±8.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Analog input voltage	VAIN0SN	ADCAnIm	T&H not used		AnVSS		AnVREF	V
		ADCA0I0-5	T&H used		0.2		A0VREF – 0.2	V
		ADCA0ImS	A0VREF ≥ EVCC		A0VSS		EVCC	V
			A0VREF < EVCC		A0VSS		A0VREF	V
		ADCA1ImS	A1VREF ≥ BVCC		A1VSS		BVCC	V
			A1VREF < BVCC		A1VSS		A1VREF	V
Operation current	IA0VREF IA1VREF	T&H not used				1.1	3.0	mA
		T&H used (max. 6 pins)					[*2]	mA
STOP, DeepSTOP, Cyclic STOP current (@LPS is stopped)	IA0VREFS IA1VREFS					1	10	µA
T&H current	ITH					0.5	1.3	mA/ch
T&H sampling time	t_{THSMP}				450			ns
T&H hold time	t_{THHOLD}						10	µs
Set up time of self diagnosis voltage circuit	t_{BOOT}				500			ns
Set up time of self diagnosis voltage level	t_{OUT}				500			ns
Pull-down resistor for diagnosis of open pins		ADCnIm pins	VI = AnVREF		350	500	650	kΩ
		ADCnImS pins	A0VREF ≥ EVCC: VI = EVCC A0VREF < EVCC: VI = A0VREF		100	215	800	kΩ
			A1VREF ≥ BVCC: VI = BVCC A1VREF < BVCC: VI = A1VREF					

(2/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Accuracy of self-diagnosis function (except diagnosis of open pins)	TESH0SN	12-bit mode	Self-diagnosis voltage level = AnVREF		4015-|TOEn|		4095	—
			Self-diagnosis voltage level = 2/3AnVREF		2651-|TOEn|	2731	2811+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		1968-|TOEn|	2048	2128+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		1285-|TOEn|	1365	1445+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		80+|TOEn|	—
		10-bit mode	Self-diagnosis voltage level = AnVREF		1003-|TOEn|		1023	—
			Self-diagnosis voltage level = 2/3AnVREF		663-|TOEn|	683	703+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		492-|TOEn|	512	532+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		321-|TOEn|	341	361+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		20+|TOEn|	—
Integral nonlinearity error*1	ILEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±2.0	LSB
				ADCA0I0-5 (T&H used)			±3.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Differential nonlinearity error*1	DLEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±1.0	LSB
				ADCA0I0-5 (T&H used)			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0V to 4.5V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Zero scale error (offset error)*1	ZSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Full scale error*1	FSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB

Note: Conversion accuracy when ADCA0ImS terminal is converted in 12-bit mode: Conversion accuracy can be applied if lower 2-bit is ignored from conversion result.

Note 1. This does not include quantization error.

Note 2. 3.0 + 1.3 × (the number of used T&H)

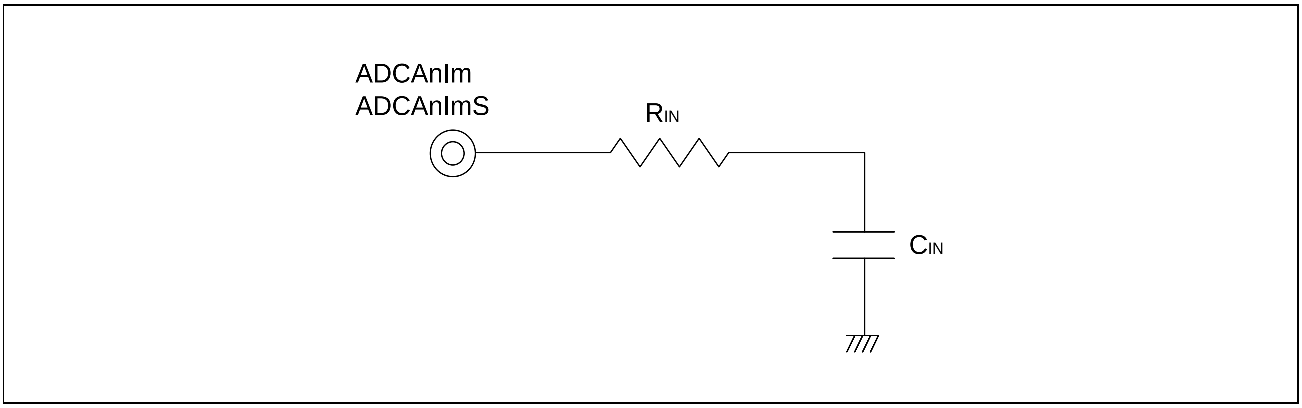
Note 3. Include the oscillation accuracy of HS IntOSC.

Note 4. When the external multiplexer is used, the detailed time of A/D conversion is MPX setup time, sampling time and successive approximation time. MPX setup time is same as “sampling time + successive approximation time”.

CAUTION

When an external digital pulse is applied to AP0, AP1, P8, P9, P18, and P19 pins during an A/D conversion this may lead to an A/D conversion result with a larger conversion error as expected due to the coupling noise of the external digital pulse.
The same behavior may apply when the digital buffer is used as an output pin. For the output port the potential degradation increases with the driven total output current of the port. In addition the conversion resolution may drop if the output current fluctuates at adjacent pins due to the coupling effect of the external circuit connected to these port pins.

47A.6.1 Equivalent Circuit of the Analog Input Block



Terminals	Condition	RIN (kΩ)	CIN (pF)
ADCA0I0 to 5	When T&H is used	14.1	2.1
	When T&H is not used	3.9	2.1
ADCA0I6 to 15	—	3.9	2.1
ADCA0I0S to 3S, 5S to 11S, 14S to 16S	—	5.7	9.2
ADCA0I4S, 17S to 19S	—	10.3	9.2
ADCA1I0 to 15	—	4	2.1
ADCA1I0S to 19S	—	5.5	7.9

CAUTION

This specification is not tested during outgoing inspection. Therefore RIN and CIN are reference values only and not guaranteed. In addition these values are specified as maximum values.

47A.7 Flash Programming Characteristics

47A.7.1 Code Flash

The code flash memory is shipped in the erased state. If the code flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47A.21 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*[3]		5*[4]		30	MHz
Number of rewrites*[1]	CWRT	Data retention of 20 years*[2]	1000			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 1000), the device can be erased “n” times for each block. For example, when a block of 32 KB is erased after 256 bytes of writing have been performed for different addresses 128 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the code flash memory.

Note 3. f_{PCLK} = 1/8 f_{CPUCLK_H}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47A.22 Programming Characteristic

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		f_{PCLK} ≥ 20 MHz CWRT < 100 times	256 B		0.4*1	6*1	ms
			8 KB		20	90	ms
			32 KB		80	360	ms
			256 KB		0.6	2.7	s
			384 KB		0.9	4.1	s
			512 KB		1.2	5.4	s
			768 KB		1.7	8.1	s
			1 MB		2.3	10.8	s
			1.5 MB		3.4	16.2	s
			2 MB		4.5	21.5	s
			3 MB		6.8	32.3	s
			4 MB		9	43	s
			6 MB		13.5	64.5	s
			8 MB		18	85.9	s
		f_{PCLK} ≥ 20 MHz CWRT ≥ 100 times	256 B		0.5*1	7.2*1	ms
			8 KB		24	108	ms
			32 KB		96	432	ms
			256 KB		0.7	3.3	s
			384 KB		1.1	4.9	s
			512 KB		1.4	6.5	s
			768 KB		2.1	9.8	s
			1 MB		2.7	13	s
			1.5 MB		4.1	19.5	s
			2 MB		5.4	26	s
			3 MB		8.1	39	s
			4 MB		10.8	52	s
			6 MB		16.2	78	s
			8 MB		21.6	104	s

Table 47A.22 Programming Characteristic

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Erase time		$f_{PCLK} \geq 20$ MHz CWRT < 100 times	8 KB		39	120	ms
			32 KB		141	480	ms
			256 KB		1.2	3.5	s
			384 KB		1.7	5.3	s
			512 KB		2.3	7	s
			768 KB		3.4	10.5	s
			1 MB		4.5	14	s
			1.5 MB		6.8	21	s
			2 MB		9	28	s
			3 MB		13.5	42	s
			4 MB		18	56	s
			6 MB		27	84	s
			8 MB		36	112	s
		$f_{PCLK} \geq 20$ MHz CWRT ≥ 100 times	8 KB		47	144	ms
			32 KB		169	576	ms
			256 KB		1.4	4.2	s
			384 KB		2.1	6.3	s
			512 KB		2.7	8.4	s
			768 KB		4.1	12.6	s
			1 MB		5.4	16.8	s
			1.5 MB		8.1	25.2	s
			2 MB		10.8	33.6	s
			3 MB		16.2	50.4	s
			4 MB		21.6	67.2	s
			6 MB		32.4	100.8	s
			8 MB		43.2	134.4	s

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

47A.7.2 Data Flash

The data flash memory is shipped in the erased state. If the data flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REG0VCC = EVCC = VPOC to 5.5 V, REG1VCC = VPOC to 3.6 V, REG1VCC ≤ REG0VCC,
BVCC = VPOC to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V,
AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47A.23 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*[3]		5*[4]		30	MHz
Number of rewrites*[1]	CWRT	Data retention 20 years*[2]	125 k			times
		Data retention 3 years*[2]	250 k			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 125000), the device can be erased “n” times for each block. For example, when a block of 64 bytes is erased after 4 bytes of writing have been performed for different addresses 168 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the data flash memory.

Note 3. f_{PCLK} = 1/8 f_{CPUCLK_H}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47A.24 Programming Characteristics

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		f_{PCLK} ≥ 20 MHz	4 B		0.16*[1]	1.7*[1]	ms
			32 KB		1.4	6.8	s
			64 KB		2.79	13.44	s
			128 KB		5.58	26.88	s
			256 KB		11.16	53.74	s
Erase time		f_{PCLK} ≥ 20 MHz	64 B		1.7*[1]	10*[1]	ms
			32 KB		0.9	5.2	s
			64 KB		1.74	10.24	s
			128 KB		3.48	20.48	s
			256 KB		6.95	40.94	s
Blank check time		f_{PCLK} ≥ 20 MHz	4 B			30*[1]	μs
			64 B			100*[1]	μs
			32 KB			35.2	ms
			64 KB			70.4	ms
			128 KB			140.8	ms
			256 KB			281.6	ms

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

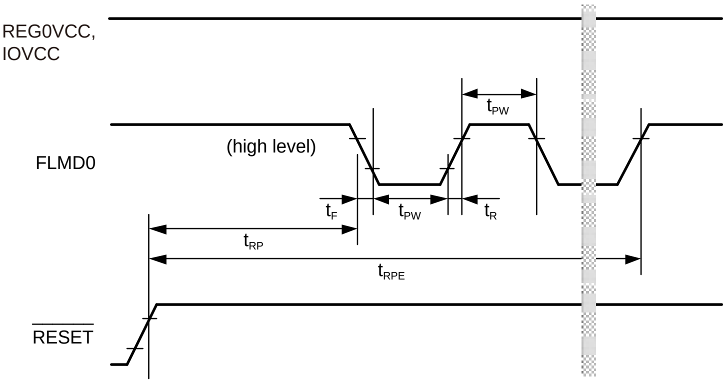
47A.7.3 Serial Programming Interface

47A.7.3.1 Serial Programmer Setup Timing

Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0 pulse input start time	t_{RP}		1.5			ms
FLMD0 pulse input end time	t_{RPE}				101.5	ms
FLMD0 low/high level width	t_{PW}		3.2			µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns

Note: IOVCC: EVCC = BVCC = A0VREF = A1VREF



47A.7.3.2 Flash Programming Interface

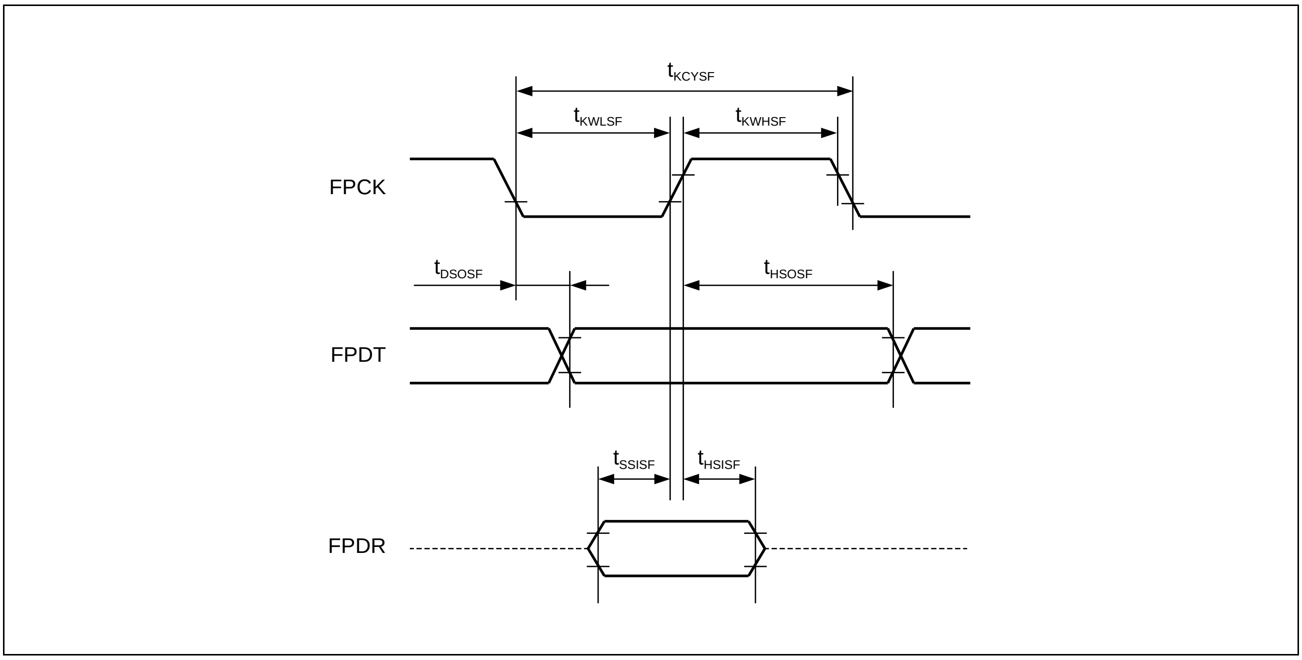
Condition: REG0VCC = EVCC = 3.0 V to 5.5 V, REG1VCC = 3.0 V to 3.6 V, REG1VCC ≤ REG0VCC, BVCC = 3.0 V to REG0VCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Flash Programming transfer rate		1-wired UART mode			1	Mbps
		2-wired UART mode (Products of CPU frequency 240 MHz max.)			1.5	Mbps
		2-wired UART mode (Products of CPU frequency 160 MHz max.)			1	Mbps
FPCK cycle time	t_{KCYSF}	3-wired clock sync mode	200*1			ns
FPCK high level width	t_{KWHSF}	3-wired clock sync mode	$t_{KCYSF} / 2 – 15$			ns
FPCK low level width	t_{KWLSF}	3-wired clock sync mode	$t_{KCYSF} / 2 – 15$			ns
FPDR setup time (vs. FPCK)	t_{SSISF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns
FPDR hold time (vs. FPCK)	t_{HSISF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns
FPDT output delay (vs. FPCK)	t_{DSOSF}	3-wired clock sync mode Not continuous transfer (data: 1st bit)			0	ns
		3-wired clock sync mode Not continuous transfer (data: except 1st bit)			$–t_{KWHSF} + 3 \times t_{Pcyc} + 36$	ns
FPDT hold time (vs. FPCK)	t_{HSOSF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns

Note 1. Input an external clock that is more than 6 clocks of PCLK.

NOTE

t_{Pcyc} is period of PCLK.



47A.8 Thermal Characteristics

47A.8.1 Parameters

Package	Item	Symbol	Estimate	Unit	Note
324-pin FPBGA	Thermal Resistance	Θja	17.4	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	9.8		
233-pin FPBGA	Thermal Resistance	Θja	17.9	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	9.9		
176-pin LQFP	Thermal Resistance	Θja	31.9	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	23.8		

NOTE

The thermal resistance and characterization parameters depend on the usage environment.

47A.8.2 Board

Conforming to JESD51-7 (4 layers)

	Board Size (mm)		
	X	Y	Area (mm^2)
Board	76.2	114.3	8709.66
Remaining copper rates	Thickness of conductors		
50-95-95-50%	70-35-35-70 µm		

Section 47B Electrical Characteristics of RH850/F1KM-S4

47B.1 Overview

The electrical spec of this device is guaranteed by the following operational condition. But, this condition is different depends on each characteristics, so refer to each chapter for more detail.

47B.1.1 Pin Groups

47B.1.1.1 272-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR	REGVCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P3, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P13, P18, P19, P21, P22
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47B.1.1.2 233-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR	REGVCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P3, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P13, P18, P19
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47B.1.1.3 176-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR	REGVCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P2, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P18
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47B.1.1.4 144-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR	REGVCC, AWOVSS	X1, X2, XT1, XT2/IP0_0
PgE	EVCC, EVSS	Related ports: JP0, P0, P1, P8, P9, P20 Related pins: $\overline{\text{RESET}}$, FLMD0
PgB	BVCC, BVSS	Related ports: P10, P11, P12, P18
PgA0	A0VREF, A0VSS	Related port: AP0
PgA1	A1VREF, A1VSS	Related port: AP1

47B.1.1.5 100-Pin Version

Symbol	Pin Group Supplied by	Related Pins/Ports
PgR	REGVCC, AWOVSS	X1, X2
PgE	EVCC, EVSS	Related ports: JP0, P0, P8, P9, P10, P11 Related pins: $\overline{\text{RESET}}$, FLMD0
PgA0	A0VREF, A0VSS	Related port: AP0

47B.1.2 General Measurement Conditions

47B.1.2.1 Common Conditions

- Power supply
  - REGVCC = EVCC = VPOC*[1] to 5.5 V
  - BVCC = VPOC*[1] to REGVCC
  - A0VREF = 3.0 V to 5.5 V
  - A1VREF = 3.0 V to 5.5 V
  - AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V
- Capacitance of the internal regulator
  - CAWOVCL: 0.1 µF ±30%
  - CISOVCL: 0.1 µF ±30% per pin
- Operating temperature
  - Tj = –40 to +130°C@R7F7016xx3ABG*[2]
  - Tj = –40 to +150°C@R7F7016xx4ABG*[2]
    @R7F7016yy3AFP*[2]

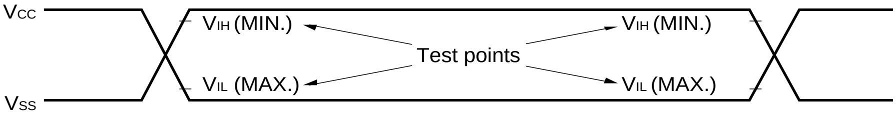
  xx = 50, 51, 52, 53
  yy = 44, 45, 46, 47, 48, 49
- Load conditions
  - CL = 30 pF

Note 1. “VPOC” means POC (power-on clear) detection voltage. For more detail, see **Section 47B.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

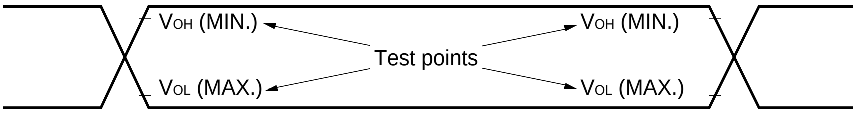
Note 2. Regarding operation temperature of each product, see **Section 1B.3, RH850/F1KM Product Lineup**.

47B.1.2.2 AC Characteristic Measurement Condition

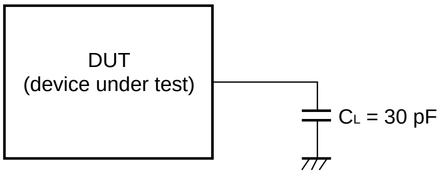
(1) AC Test Input Measurement Points



(2) AC Test Output Measurement Points



(3) Load Conditions



CAUTION

If the load capacitance exceeds 30pF due to the circuit configuration, it is recommended to insert a buffer in order to reduce capacitance to less than 30pF.

47B.2 Absolute Maximum Ratings

CAUTIONS

1. Do not directly connect outputs (or input/outputs) to each other, power supply and ground.
2. Even momentarily exceeding the absolute maximum rating for just one item creates a threat of failure in the reliability of the products. That is, the absolute maximum ratings are the levels that raise a threat of physical damage to the products. Be sure to use the products only under conditions that do not exceed the ratings. The quality and normal operation of the product are guaranteed under the standards and conditions given as DC and AC characteristics.
3. When designing an external circuit ensure that the connections don't conflict with the port state of this device.

47B.2.1 Supply Voltages

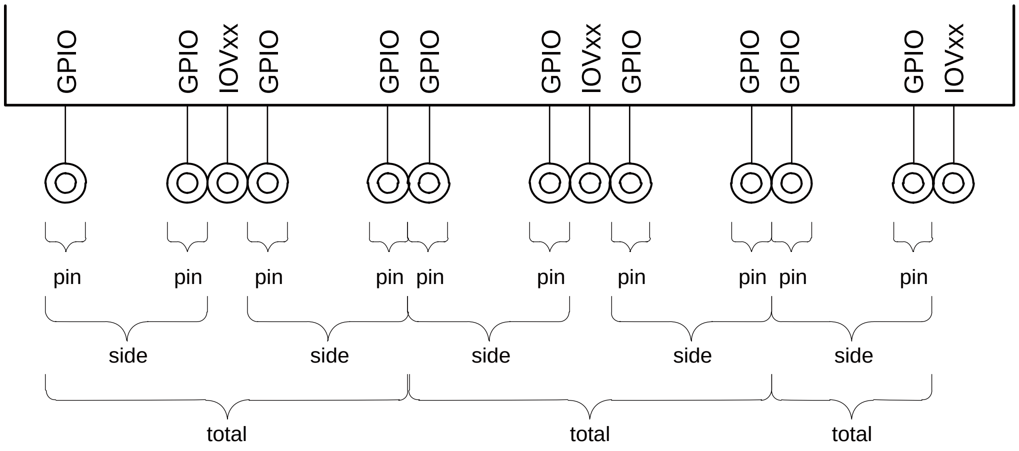
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
System supply voltage	REGVCC		–0.5		6.5	V
	AWOVSS		–0.5		0.5	V
	ISOVSS		–0.5		0.5	V
Port supply voltage	EVCC		–0.5		6.5	V
	BVCC		–0.5		6.5	V
	EVSS		–0.5		0.5	V
	BVSS		–0.5		0.5	V
A/D-converter supply voltage	A0VREF		–0.5		6.5	V
	A1VREF		–0.5		6.5	V
	A0VSS		–0.5		0.5	V
	A1VSS		–0.5		0.5	V

47B.2.2 Port Voltages

Item	Pin Group*[1]	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	PgR	VI		–0.5		REGVCC + 0.5 (Do not exceed 6.5 V)	V
	PgE			–0.5		EVCC + 0.5 (Do not exceed 6.5 V)	V
	PgB			–0.5		BVCC + 0.5 (Do not exceed 6.5 V)	V
	PgA0			–0.5		A0VREF + 0.5 (Do not exceed 6.5 V)	V
	PgA1			–0.5		A1VREF + 0.5 (Do not exceed 6.5 V)	V

Note 1. The characteristics of the alternative-function pins are the same as those of the port pins unless otherwise specified.

47B.2.3 Port Current



Definition of the condition:

- Per pin: Output current of one GPIO
- Per side: Total output current of all GPIO pins on one side of one IOVxx
- Total: Total output current of both sides of one IOVxx

Note:

- GPIO: General-purpose I/O pin (JP0, P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20, P21, P22, AP0, AP1)
- IOVxx: Power supply pin for I/O pins (EVCC/EVSS, BVCC/BVSS, A0VREF/A0VSS, A1VREF/A1VSS)

47B.2.3.1 272-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_10 to P20_14)			–48	mA
			Per side (total of P20_0 to P20_9)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4,P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Per side (total of P3_1 to P3_10)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P18_0 to P18_7)			–48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			–48	mA
			Per side (total of P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			–48	mA
			Per side (total of P21_0, P21_2 to P21_4, P22_0 to P22_2)			–48	mA
			Per side (total of P22_3 to P22_8)			–48	mA
			Per side (total of P21_1, P22_9 to P22_15)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA

(272-pin version)

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_10 to P20_14)			48	mA
			Per side (total of P20_0 to P20_9)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Per side (total of P3_1 to P3_10)			48	mA
			Total (EVCC)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			48	mA
			Per side (total of P21_0, P21_2 to P21_4, P22_0 to P22_6)			48	mA
			Per side (total of P21_1, P22_7 to P22_15)			48	mA
			Total (BVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VREF)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VREF)			48	mA

47B.2.3.2 233-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P18_0 to P18_7)			–48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			–48	mA
			Per side (total of P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA

(233-pin version)

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Total (EVCC)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P18_8 to P18_15, P19_0 to P19_3)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7)			48	mA
			Total (BVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VREF)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VREF)			48	mA

47B.2.3.3 176-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P10_6 to P10_9, P18_0 to P18_7)			–48	mA
			Per side (total of P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2)			–48	mA
			Per side (total of P10_0 to P10_2)			–30	mA
			Per side (total of P10_3 to P10_5)			–30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_0 to P20_5)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P2_2, P2_3)			48	mA
			Per side (total of P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Total (EVSS)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_7)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2)			48	mA
			Per side (total of P10_0 to P10_2)			30	mA
			Per side (total of P10_3 to P10_5)			30	mA
			Per side (total of P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5)			48	mA
			Total (BVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VSS)			48	mA

47B.2.3.4 144-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (total of P9_0 to P9_4, P20_4, P20_5)			–48	mA
			Per side (total of P0_0 to P0_3)			–40	mA
			Per side (total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P8_2, P8_10 to P8_12)			–48	mA
			Per side (total of JP0_0 to JP0_2, P1_8 to P1_11)			–48	mA
			Per side (total of JP0_6, P0_7 to P0_10, P1_4, P1_5, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Total (EVCC)			–60	mA
		PgB	Per pin			–10	mA
			Per side (total of P10_6 to P10_9, P18_0 to P18_3)			–48	mA
			Per side (total of P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2)			–48	mA
			Per side (total of P10_0 to P10_5, P10_15, P11_0, P11_8 to P11_12)			–48	mA
			Total (BVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
		PgA1	Per pin			–10	mA
			Total (A1VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (total of P9_0 to P9_4, P20_4, P20_5)			48	mA
			Per side (total of P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3)			48	mA
			Per side (total of JP0_0 to JP0_5, P1_8 to P1_11, P8_2, P8_10 to P8_12)			48	mA
			Per side (total of JP0_6, P0_7 to P0_10)			48	mA
			Per side (total of P1_4, P1_5, P8_0, P8_1, P8_3 to P8_9)			48	mA
			Total (EVSS)			60	mA
		PgB	Per pin			10	mA
			Per side (total of P18_0 to P18_3)			48	mA
			Per side (total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (total of P11_3 to P11_7, P11_15, P12_0 to P12_2)			48	mA
			Per side (total of P10_0 to P10_5, P10_15, P11_0, P11_8 to P11_12)			48	mA
			Total (BVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA
		PgA1	Per pin			10	mA
			Total (A1VSS)			48	mA

47B.2.3.5 100-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (Total of P9_0 to P9_4)			–48	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5)			–48	mA
			Per side (Total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P8_2, P8_10 to P8_12)			–48	mA
			Per side (Total of JP0_0 to JP0_2)			–30	mA
			Per side (Total of P0_7 to P0_10, P8_3 to P8_9)			–48	mA
			Per side (Total of P10_6 to P10_9)			–40	mA
			Per side(Total of P10_10 to P10_14, P11_1 to P11_7)			–48	mA
			Per side (Total of P10_0 to P10_2)			–30	mA
			Total (EVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (Total of P9_0 to P9_4)			48	mA
			Per side (Total of P0_0 to P0_6, P0_11 to P0_14, P10_3 to P10_5)			48	mA
			Per side (Total of JP0_0 to JP0_5, P8_2, P8_10 to P8_12)			48	mA
			Per side (Total of P0_7 to P0_10)			40	mA
			Per side (Total of P8_3 to P8_9)			48	mA
			Per side(Total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (Total of P11_3 to P11_7)			48	mA
			Per side (Total of P10_0 to P10_2)			30	mA
			Total (EVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA

47B.2.4 Temperature Condition

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Storage temperature	Tstg		–55		150	°C
Junction temperature	Tj	R7F7016xx3ABG	–40		130	°C
		R7F7016xx4ABG R7F7016yy3AFP	–40		150	°C

Note: xx = 40, 41, 42, 43, 50, 51, 52, 53
yy = 34, 35, 36, 37, 38, 39, 44, 45, 46, 47, 48, 49

Regarding operation temperature of each product, see **Section 1B.3, RH850/F1KM Product Lineup**.

47B.3 Operational Condition

47B.3.1 Recommended Operating Conditions

Products of CPU frequency 240 MHz max.

(1/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CPU clock frequency	f_{CPUCLK_H}	CKDIVMD = 1			240	MHz
		CKDIVMD = 0			120	MHz
	f_{CPUCLK_M}				120	MHz
	f_{CPUCLK_L}	for OSTMn			60	MHz
		for MEMC*5				
	f_{CPUCLK_UL}				30	MHz
Peripheral clock (clock domain) frequency*1	f_{CKSCLK_AWDTA}	for WDTA0			240*2	kHz
	f_{CKSCLK_ATAUJ}	for TAUJ0			40	MHz
		for TAUJ2				
	f_{CKSCLK_ARTCA}	for RTCA0			4	MHz
	f_{CKSCLK_AADCA}	for ADCA0			40	MHz
	f_{CKSCLK_AFOUT}	for FOUT			24	MHz
	$f_{CKSCLK\_ICPUCLK}$	for CPU subsystem			240 / 120	MHz
	f_{CKSCLK_IPERI1}	for TAUD0			80	MHz
		for TAUJ1				
		for TAUJ3				
		for ENCA0				
		for TAPA0				
		for PIC0				
		for SFMA0				
	f_{CKSCLK_IPERI2}	for TAUBn			40	MHz
		for RCFDCn (clkc)				
		for RSENTn				
		for PWBAn				
		for PWGAn				
		for PWSAn				
	f_{CKSCLK_ILIN}	for RLIN24n			40	MHz
		for RLIN3n				
	f_{CKSCLK_IADCA}	for ADCA1			40	MHz
	f_{CKSCLK_ICAN}	for RCFDCn (PCLK)			80	MHz
	$f_{CKSCLK\_ICANOSC}$	for RCFDCn (clk_xincan)			24	MHz
	f_{CKSCLK_ICSI}	for CSIGn			80	MHz
		for CSIHn				
	f_{CKSCLK_IIIC}	for RIICn			40	MHz
	$f_{LS\ IntOSC}$	for WDTA1			240*2	kHz
	f_{EMCLK}	for LPSn			8	MHz

(2/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Power supply	REGVCC	REGVCC = EVCC	VPOC*3		5.5	V
	EVCC					
	BVCC		VPOC*3		REGVCC	V
	A0VREF		3.0		5.5	V
	A1VREF					
Normal operation voltage	AWOVCL		1.1	1.25	1.35	V
	ISOVCL					
Limited operation voltage*4	AWOVCL		1.35		1.43	V
	ISOVCL					

Note 1. For clock specification of peripherals, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**.

Note 2. This frequency depends on the internal oscillator (LS IntOSC).

Note 3. “VPOC” means POC (power-on clear) detection voltage (TYP. 2.85 V). For detail, see **Section 47B.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.
In addition, the guaranteed operation in DC characteristic.
And AC characteristic is guaranteed when more than 3.0 V.
When the power supply voltage is VPOC to 3.0 V, the device does not malfunction.

Note 4. Reliability restrictions from 1.35 V to 1.43 V.

Note 5. Devided by 2 on MEMC internal.

Products of CPU frequency 160 MHz max.

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CPU clock frequency	f_{CPUCLK_H}				160	MHz
	f_{CPUCLK_M}				80	MHz
	f_{CPUCLK_L}	for OSTMn			40	MHz
		for MEMC*5				
	f_{CPUCLK_UL}				20	MHz
Peripheral clock (clock domain) frequency*1	f_{CKSCLK_AWDTA}	for WDTA0			240*2	kHz
	f_{CKSCLK_ATAUJ}	for TAUJ0			40	MHz
		for TAUJ2				
	f_{CKSCLK_ARTCA}	for RTCA0			4	MHz
	f_{CKSCLK_AADCA}	for ADCA0			40	MHz
	f_{CKSCLK_AFOUT}	for FOUT			24	MHz
	$f_{CKSCLK\_ICPUCLK}$	for CPU subsystem			160	MHz
	f_{CKSCLK_IPERI1}	for TAUD0			80	MHz
		for TAUJ1				
		for TAUJ3				
		for ENCA0				
		for TAPA0				
		for PIC0				
		for SFMA0				
	f_{CKSCLK_IPERI2}	for TAUBn			40	MHz
		for RCFDCn (clkc)				
		for RSENTn				
		for PWBAn				
		for PWGAn				
		for PWSAn				
	f_{CKSCLK_ILIN}	for RLIN24n			40	MHz
		for RLIN3n				
	f_{CKSCLK_IADCA}	for ADCA1			40	MHz
	f_{CKSCLK_ICAN}	for RCFDCn (pclk)			80	MHz
	$f_{CKSCLK\_ICANOSC}$	for RCFDCn (clk_xincan)			24	MHz
	f_{CKSCLK_ICSI}	for CSIGn			80	MHz
		for CSIHn				
	f_{CKSCLK_IIIC}	for RIICn			40	MHz
	$f_{LS\ IntOSC}$	for WDTA1			240*2	kHz
	f_{EMCLK}	for LPSn			8	MHz
Power supply	REGVCC	REGVCC = EVCC	VPOC*3		5.5	V
	EVCC					
	BVCC		VPOC*3		REGVCC	V
	A0VREF		3.0		5.5	V
	A1VREF					
Normal operation voltage	AWOVCL		1.1	1.25	1.35	V
	ISOVCL					
Limited operation voltage*4	AWOVCL		1.35		1.43	V

Note 1. For clock specification of peripherals, see **Section 12AB, Clock Controller of RH850/F1KH-D8, RH850/F1KM-S4**.

Note 2. This frequency depends on the internal oscillator (LS IntOSC).

Note 3. “VPOC” means POC (power-on clear) detection voltage (TYP. 2.85 V). For detail, see **Section 47B.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.
In addition, the guaranteed operation in DC characteristic.
And AC characteristic is guaranteed when more than 3.0 V.
When the power supply voltage is VPOC to 3.0 V, the device does not malfunction.

Note 4. Reliability restrictions from 1.35 V to 1.43 V.

Note 5. Devided by 2 on MEMC internal.

47B.3.2 Oscillator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C

(1) MainOSC (In Case of Using a Crystal/Ceramic)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MainOSC frequency*[3]	f_{MOSC}		8		24	MHz
MainOSC current consumption	I_{MOSC}	After stabilization		1.9*[2]	2.3*[2]	mA
MainOSC oscillation start point	V_{MOSCSP}		VPOC			V
MainOSC oscillation operating point	V_{MOSCOP}			0.5 × REGVCC*[2]		V
MainOSC oscillation amplitude	$V_{MOSCAMP}$		0.4 × REGVCC – 0.2*[2]			V
MainOSC oscillation stabilization time	t_{MSTB}			2*[1],*[2]		ms
MainOSC transconductance	g_{m_MOSC}	MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 00		11.1*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 01		10.6*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 10		9.3*[1]		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 11		7.8*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 00		8.6*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 01		7.8*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 10		6.1*[1]		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 11		4.0*[1]		mA/V

Note 1. Oscillator stabilization time is time until being set (“1”) in MOSCS.MOSCCLKACT bit after MOSCE.MOSCENTRG bit is written “1”, and depends on the setting value of MOSCST register. Please decide appropriate oscillation stabilization time by matching test with resonator and oscillation circuit.

Note 2. This is reference value.

Note 3. The following four crystal/ceramic resonator frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

(2) MainOSC (In Case of External Clock Input to X1)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
X1 clock Input frequency*[1]	f_{EX}		8		24	MHz
X1 clock Input cycle time	t_{EXCYC}		41.7		125	ns
X1 High level Input voltage	V_{IH}		0.7 × REGVCC		REGVCC + 0.5	V
		@Flash Programing Interface*[2]	0.8 × REGVCC		REGVCC + 0.5	V
X1 Low level Input voltage	V_{IL}		–0.5		0.3 × REGVCC	V
		@Flash Programing Interface*[2]	–0.5		0.2 × REGVCC	V
X1 Input leakage current	I_{LIH}	VI = REGVCC			0.5	μA
	I_{LIL}	VI = 0 V			–0.5	μA
X1 clock Input low-level pulse width	t_{EXL}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input high-level pulse width	t_{EXH}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input period jitter			–0.3		0.3	ns

Note 1. The following four external clock input frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

Note 2. X2 should be open and its parasitic capacitance should be less than 5 pF.

(3) SubOSC

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SubOSC frequency	f_{SOSC}	Crystal	30	32.768	38	kHz
SubOSC current consumption	I_{SOSC}	After stabilization		1.5*[2]	4*[2]	μA
SubOSC DC operating point	$V_{SOSCDCOP}$			0.65*[2]		V
SubOSC oscillation stabilization time	t_{SSTB}			*[1]		s

Note 1. Oscillator stabilization time is time until being set (“1”) in SOSCS.SOSCCLKACT bit after SOSCE.SOSCENTRG bit is written “1”, and depends on the setting value of SOSCST register. Please decide appropriate oscillation stabilization time by matching test with resonator and oscillation circuit.

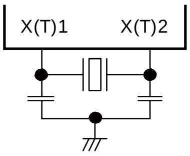
Note 2. This is reference value.

CAUTION

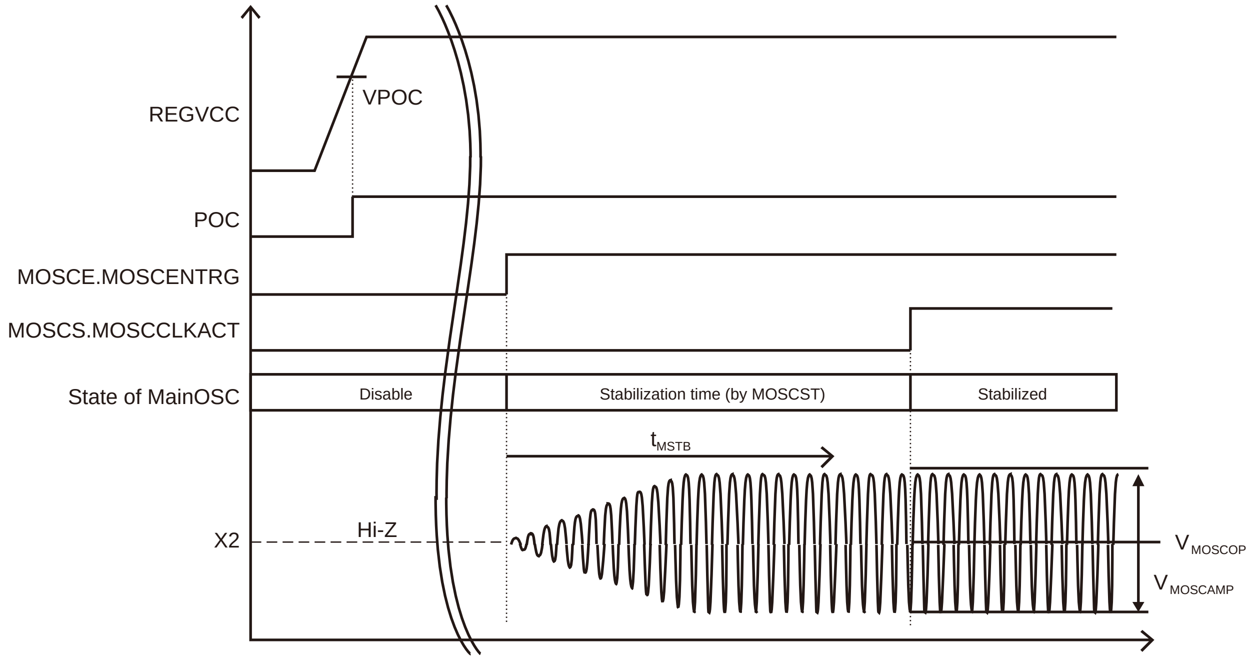
The oscillation stabilization time differs according the matching with the external resonator circuit. It is recommended to determine the oscillation stabilization time by an oscillator matching test.

NOTE

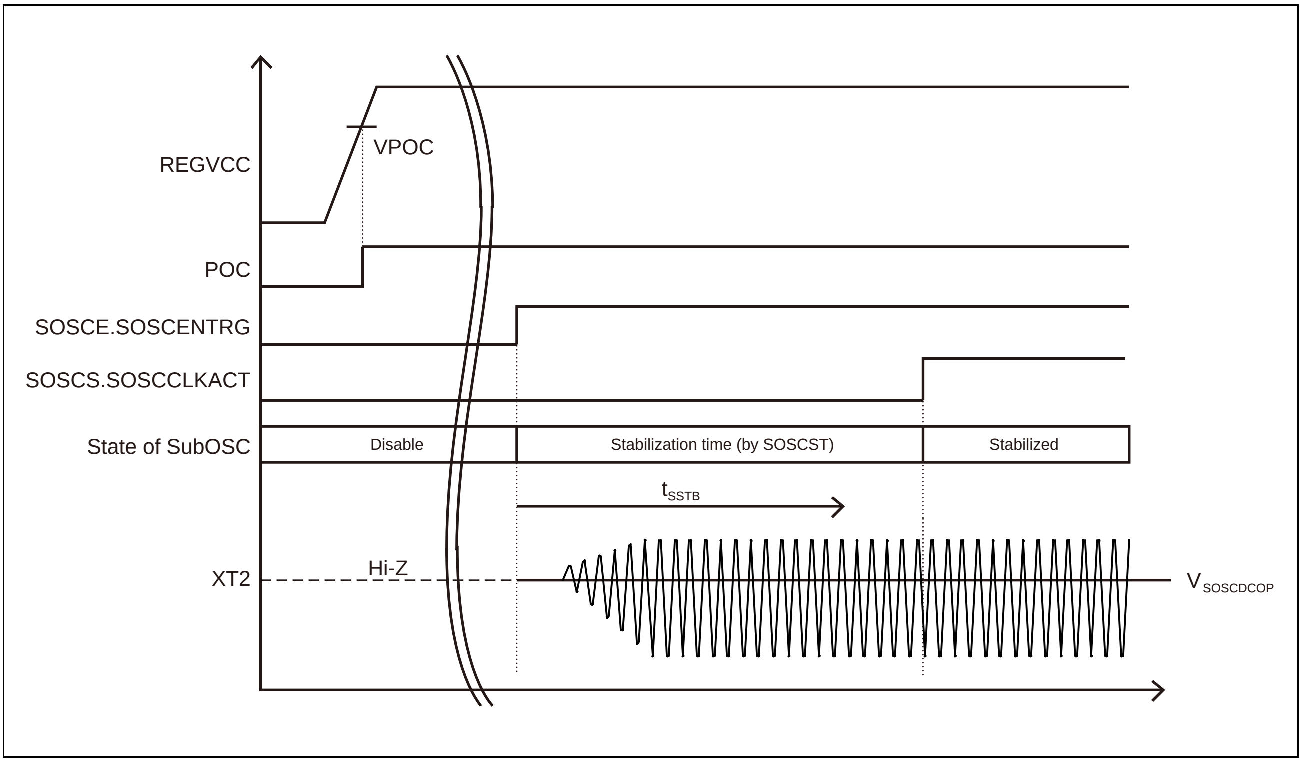
Recommended oscillator circuit is shown below.



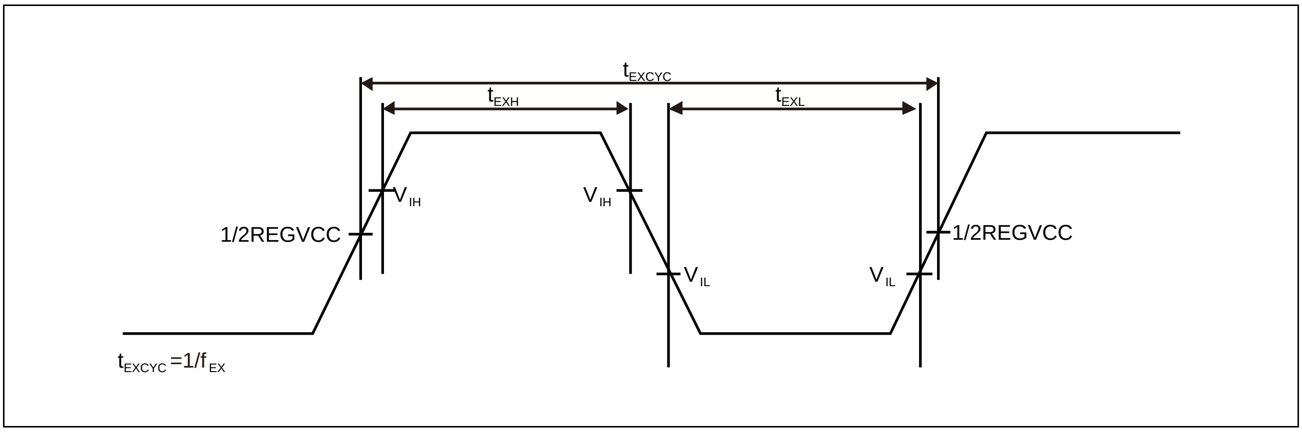
MainOSC



SubOSC



External clock



47B.3.3 Internal Oscillator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LS IntOSC frequency	f_{RL}		220.8	240	259.2	kHz
HS IntOSC frequency*[3]	f_{RH}		7.6	8	8.4	MHz
		After user trimming @ trimming temp*[2]	7.92	8	8.08	MHz
HS IntOSC current consumption	I_{RH}	After stabilization			170*[1]	μA
HS IntOSC oscillation stabilization time	t_{RHSTB}				54.4	μs

Note 1. This is reference value.

Note 2. The HS IntOSC frequency may not meet the specification range (8.00 MHz ±0.08 MHz after user trimming @ trimming temp) in the while writing/erasing the code/data flash.

Note 3. The HS IntOSC frequency may not meet the specification range in the Cyclic STOP/Cyclic RUN mode.

47B.3.4 PLL Characteristics

47B.3.4.1 PLL0 (for CPU, with SSCG) Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Input frequency	$f_{PLL0CLKIN}$	MainOSC			8		24	MHz
		HS IntOSC*3			7.6	8.0	8.4	MHz
		HS IntOSC After user trimming @ trimming temp*3			7.92	8.0	8.08	MHz
Output frequency	$f_{CPLL0OUT}$	SSCG mode	MainOSC	Products of CPU frequency 240 MHz max.	105.8		240	MHz
				Products of CPU frequency 160 MHz max.	105.8		160	MHz
			HS IntOSC*3		67		84	MHz
			HS IntOSC After user trimming @ trimming temp*3		69.8		80.8	MHz
Modulation frequency	f_{MOD}				20		100	kHz
Frequency dithering range*2	f_{DIT}				0.82	1.0	1.18	%
					1.64	2.0	2.36	%
					2.46	3.0	3.54	%
					3.28	4.0	4.72	%
					4.10	5.0	5.90	%
					4.92	6.0	7.08	%
					6.56	8.0	9.44	%
					8.20	10.0	11.80	%
Lock time*1	t_{LCK0}	SSCG mode	PLL0ST = 0000 1B80$_H$		814.9	880	956.6	µs

Note 1. Lock time is time until being set (“1”) in PLL0S.PLL0CLKACT bit after PLL0E.PLL0ENTRG bit is written “1”.

Note 2. "Frequency dithering range” is set by PLL0ADJ[2:0] bits of PLL0C registers.

Note 3. The HS IntOSC has a frequency deviation. When the HSIntOSC is used the frequency deviation should be considered for the customer application as it affects peripheral functions (e.g. TAUx, ADCAn, etc.).

47B.3.4.2 PLL1 (for CPU/Peripheral) Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input frequency	$f_{PLL1CLKIN}$	MainOSC	8		24	MHz
		HS IntOSC*[3]	7.6	8.0	8.4	MHz
		HS IntOSC After user trimming @ trimming temp*[3]	7.92	8.0	8.08	MHz
Output frequency	$f_{CPLL1OUT}$	MainOSC	80		120	MHz
		HS IntOSC*[3]	76	80	84	MHz
	$f_{PPLLOUT}$		76	80	84	MHz
Output period jitter*[1]	t_{CPJ1}		–100		100	ps
Long term jitter*[1]	t_{LTJ}	term = 1 μs	–500		500	ps
		term = 10 μs	–1		1	ns
		term = 20 μs	–2		2	ns
Lock time*[2]	t_{LCK1}		104	112.3	122.1	μs

Note 1. This is reference value.

Note 2. Lock time is time until being set (“1”) in PLL1S.PLL1CLKACT bit after PLL1E.PLL1ENTRG bit is written “1”.

Note 3. The HS IntOSC has a frequency deviation. When the HSIntOSC is used the frequency deviation should be considered for the customer application as it affects peripheral functions (e.g. TAUx, ADCAn, etc.).

47B.4 DC Characteristics

47B.4.1 Capacitance

Condition: REGVCC = EVCC = BVCC = A0VREF = A1VREF = AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, Ta = 25°C

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input capacitance	CI*[1]	f = 1 MHz 0 V for non measurement pins			10	pF
Input/output capacitance	CIO*[2]				10	pF

Note 1. CI: Capacitance between the input pin and ground

Note 2. CIO: Capacitance between the input/output pin and ground

47B.4.2 Pin Characteristics

Condition: Some of the conditions mentioned in this chapter can be selected by software and described in the hardware user's manual.

(1/5)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
$\overline{\text{RESET}}$	—	—	✓	—	—	—	—	—	—
FLMD0	—	✓	—	—	—	—	—	✓	✓
AP0_0	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_1	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_2	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_3	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_4	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_5	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_6	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_7	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_8	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_9	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_10	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_11	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_12	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_13	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_14	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_15	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_0	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_1	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_2	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_3	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_4	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_5	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_6	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_7	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_8	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_9	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_10	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_11	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_12	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_13	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_14	✓	—	—	—	—	✓	Slow	—	✓*1
AP1_15	✓	—	—	—	—	✓	Slow	—	✓*1
IP0_0	—	—	—	—	—	—	—	—	—
JP0_0	—	✓	—	✓	✓	—	Slow	✓	✓
JP0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
JP0_2	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_3	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_4	—	—	—	✓	—*5	—	Slow	✓	✓
JP0_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(2/5)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
JP0_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_2	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_3	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_5	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P1_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_0	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P10_1	—	✓	—	✓	✓*6	—	Slow/Fast*3	✓	✓
P10_2	—	✓	—	✓	✓*6	—	Slow/Fast*3	✓	✓
P10_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_4	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P10_5	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P10_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓

(3/5)

	Port Input Buffer Function						Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P10_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_2	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_3	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_7	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_10	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P11_11	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P11_12	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P11_15	—	✓	—	✓	✓*6	—	Slow/Fast	✓	✓
P12_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P12_4	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P12_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_2	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_3	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_5	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
P13_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P13_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P18_0	—	✓	—	✓	✓*6	✓	Slow/Fast	✓	✓*4
P18_1	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_2	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_3	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_4	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_5	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_6	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_7	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_8	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_9	—	✓	—	✓	✓	✓	Slow/Fast	✓	✓*4
P18_10	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_11	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_12	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_13	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4

(4/5)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P18_14	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P18_15	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_0	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_1	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_2	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P19_3	—	✓	—	✓	—	✓	Slow/Fast	✓	✓*4
P2_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P2_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P20_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P21_0	—	✓	—	✓	—	—	Slow	✓	✓
P21_1	—	✓	—	✓	—	—	Slow	✓	✓
P21_2	—	✓	—	✓	—	—	Slow	✓	✓
P21_3	—	✓	—	✓	—	—	Slow	✓	✓
P21_4	—	✓	—	✓	—	—	Slow	✓	✓
P22_0	—	✓	—	✓	—	—	Slow	✓	✓
P22_1	—	✓	—	✓	—	—	Slow	✓	✓

(5/5)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P22_2	—	✓	—	✓	—	—	Slow	✓	✓
P22_3	—	✓	—	✓	—	—	Slow	✓	✓
P22_4	—	✓	—	✓	—	—	Slow	✓	✓
P22_5	—	✓	—	✓	—	—	Slow	✓	✓
P22_6	—	✓	—	✓	—	—	Slow	✓	✓
P22_7	—	✓	—	✓	—	—	Slow	✓	✓
P22_8	—	✓	—	✓	—	—	Slow	✓	✓
P22_9	—	✓	—	✓	—	—	Slow	✓	✓
P22_10	—	✓	—	✓	—	—	Slow	✓	✓
P22_11	—	✓	—	✓	—	—	Slow	✓	✓
P22_12	—	✓	—	✓	—	—	Slow	✓	✓
P22_13	—	✓	—	✓	—	—	Slow	✓	✓
P22_14	—	✓	—	✓	—	—	Slow	✓	✓
P22_15	—	✓	—	✓	—	—	Slow	✓	✓
P3_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_2	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P3_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P8_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_4	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_5	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_6	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_7	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_8	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_9	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_10	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_11	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_12	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_4	—	✓	—	✓	—	✓	Slow	✓	✓*4

Note 1. Pull-down resistor for ADC diagnostic purpose. Control via ADC self-diagnostic register.

Note 2. Supports Cload: 100 pF

Note 3. Supports Cload: 50 pF

Note 4. Pull-down resistors for ADC diagnostic and internal pull-down purposes. For ADC diagnostic, control via ADC self-diagnostic register. For internal pull-down, control via PD register.

Note 5. TTL is selected for Boundary scan mode or Nexus in normal operating mode.

Note 6. Only available for 176-pin, 233-pin and 272-pin devices.

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

(1/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High level input voltage	VIH	CMOS		0.65 × IOVCC		IOVCC + 0.3	V
		SHMT1*[3]		0.65 × IOVCC		IOVCC + 0.3	V
		SHMT2		0.75 × IOVCC		IOVCC + 0.3	V
		SHMT4		0.8 × IOVCC		IOVCC + 0.3	V
		TTL	IOVCC = VPOC to 3.6 V	2.0		IOVCC + 0.3	V
			IOVCC = 3.6 V to 5.5 V	2.2		IOVCC + 0.3	V
		IP0_0 pin		0.7 × REGVCC		REGVCC	V
Low level input voltage	VIL	CMOS		–0.3		0.35 × IOVCC	V
		SHMT1		–0.3		0.35 × IOVCC	V
		SHMT2		–0.3		0.25 × IOVCC	V
		SHMT4		–0.3		0.5 × IOVCC	V
		TTL		–0.3		0.8	V
		IP0_0 pin		0		0.3 × REGVCC	V
Input hysteresis for Schmitt	VH	SHMT1		0.3			V
		SHMT2		0.2 × IOVCC			V
		SHMT4		0.1			V
Input leakage current	ILIH	IP0_0 pin, VI = REGVCC				0.5	μA
		$\overline{\text{RESET}}$, FLMD0, JP0, P0, P1, P2, P3, P8, P9, P20 pin, VI = EVCC*[2]				0.5	μA
		P10, P11, P12, P13, P18, P19, P21, P22 pin, VI = BVCC*[2]				0.5	μA
		AP0 pin, VI = A0VREF*[2], Tj ≤ 130°C				0.3	μA
		AP0 pin, VI = A0VREF*[2]				0.5	μA
		AP1 pin, VI = A1VREF*[2], Tj ≤ 130°C				0.3	μA
		AP1 pin, VI = A1VREF*[2]				0.5	μA
	ILIL	IP0_0 pin, VI = 0 V				–0.5	μA
		$\overline{\text{RESET}}$, FLMD0, JP0, P0, P1, P2, P3, P8, P9, P20 pin, VI = 0 V*[2]				–0.5	μA
		P10, P11, P12, P13, P18, P19, P21, P22 pin, VI = 0V*[2]				–0.5	μA
		AP0 pin, VI = 0 V*[2], Tj ≤ 130°C				–0.3	μA
		AP0 pin, VI = 0 V*[2]				–0.5	μA
		AP1 pin, VI = 0 V*[2], Tj ≤ 130°C				–0.3	μA
		AP1 pin, VI = 0 V*[2]				–0.5	μA
Internal pull-up resistance	RU	except FLMD0 pin, VI = 0 V		20 (275 μA)	40	100	kΩ
		FLMD0 pin, VI = 0V*[3]		4 (1375 μA)		36	kΩ
Internal pull-down resistance	RD	except FLMD0 pin, VI = IOVCC		20 (275 μA)	40	100	kΩ
		FLMD0 pin, VI = EVCC		4 (1375 μA)		36	kΩ

(2/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High level output voltage	VOH	Fast mode	IOH = –5 mA (6 pins)*4	IOVCC – 1.0			V
			IOH = –3 mA (10 pins)*4	IOVCC – 1.0			V
			IOH = –1 mA (16 pins)*4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins)*4	IOVCC – 0.5			V
		Slow mode	IOH = –1 mA (16 pins)*4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins)*4	IOVCC – 0.5			V
Low level output voltage	VOL	Fast mode	IOL = 5 mA (6 pins)*4			0.4	V
			IOL = 3 mA (10 pins)*4			0.4	V
			IOL = 1 mA (16 pins)*4			0.4	V
		Slow mode	IOL = 1 mA (16 pins)*4			0.4	V
Rise/Fall time	t_{KRP}/t_{KFP}	Fast mode (except below pins)*5	CL = 30 pF			7	ns
			CL = 50 pF			12	ns
			CL = 100 pF			24	ns
		Fast mode (P0_5, P0_6, P10_1, P10_2, P11_2, P11_3, P11_6, P11_7)*6	CL = 50 pF			6	ns
		Fast mode (P0_2, P0_3)*6	CL = 100 pF			6.15	ns
		Slow mode*5	CL = 30 pF			37	ns
			CL = 50 pF			62	ns
			CL = 100 pF			124	ns
Output frequency	f_O	Fast mode	CL = 30 pF			40	MHz
		Slow mode	CL = 30 pF			10	MHz
			CL = 50 pF			6	MHz
			CL = 100 pF			3	MHz

Note 1. “IOVCC” means the pins are assigned to the power supply (EVCC, BVCC, A0VREF and A1VREF).

Note 2. Not select the analog input function of ADCn.

Note 3. When the internal pull-up resistor of FLMD0 pin is applied by FLMDCNT register, please connect 86 kΩ or more as external pull-down resistor.

Note 4. The number of pin indicates simultaneous ON.

Note 5. Measurement point: 0.1 × IOVCC to 0.9 × IOVCC

Note 6. Measurement point: 0.2 × IOVCC to 0.8 × IOVCC

47B.4.2.1 Output Current

(1) 272-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_10 to P20_14			–30	mA
				P20_0 to P20_9			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
				P3_1 to P3_10			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P18_0 to P18_7			–30	mA
				P18_8 to P18_15, P19_0 to P19_3			–30	mA
				P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			–30	mA
				P21_0, P21_2 to P21_4, P22_0 to P22_2			–7	mA
				P22_3 to P22_8			–4	mA
				P21_1, P22_9 to P22_15			–8	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA

(272-pin version)

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_10 to P20_14			30	mA
				P20_0 to P20_9			30	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
				P3_1 to P3_10			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P18_8 to P18_15, P19_0 to P19_3			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			30	mA
				P21_0, P21_2 to P21_4, P22_0 to P22_6			11	mA
				P21_1, P22_7 to P22_15			10	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47B.2.3, Port Current**.

(2) 233-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15, P3_0			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P18_0 to P18_7			–30	mA
				P18_8 to P18_15, P19_0 to P19_3			–30	mA
				P10_6 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			30	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6 to P2_12			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P2_13 to P2_15 P3_0, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P18_8 to P18_15, P19_0 to P19_3			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2, P13_0, P13_1			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5, P13_2 to P13_7			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47B.2.3, Port Current**.

(3) 176-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			–30	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11, P2_0, P2_1			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P1_14, P1_15, P2_2 to P2_5, P8_0, P8_1, P8_3 to P8_9			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P10_6 to P10_9, P18_0 to P18_7			–30	mA
				P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2			–30	mA
				P10_0 to P10_2			–15	mA
				P10_3 to P10_5			–15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_0 to P20_5			11	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P1_12, P1_13, P2_6			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P2_0, P2_1, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10, P2_2, P2_3			30	mA
				P1_4, P1_5, P1_14, P1_15, P2_4, P2_5, P8_0, P8_1, P8_3 to P8_9			30	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_7			30	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2			30	mA
				P10_0 to P10_2			15	mA
				P10_3 to P10_5			15	mA
				P10_15, P11_0, P11_8 to P11_12, P12_3 to P12_5			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47B.2.3, Port Current**.

(4) 144-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4, P20_4, P20_5			–15	mA
				P0_0 to P0_3			–20	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P1_0 to P1_3, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2, P1_8 to P1_11			–30	mA
				JP0_6, P0_7 to P0_10, P1_4, P1_5, P8_0 to P8_1, P8_3 to P8_9			–30	mA
			Total (EVCC)				–60	mA
		PgB	Per side	P10_6 to P10_9, P18_0 to P18_3			–30	mA
				P10_10 to P10_14, P11_1 to P11_7, P11_15, P12_0 to P12_2			–30	mA
				P10_0 to P10_5, P10_15, P11_0, P11_8 to P11_12			–30	mA
			Total (BVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
		PgA1	Total (A1VREF)				–8	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4, P20_4 to P20_5			15	mA
				P0_0 to P0_6, P0_11 to P0_14, P1_0 to P1_3			30	mA
				JP0_0 to JP0_5, P1_8 to P1_11, P8_2, P8_10 to P8_12			30	mA
				JP0_6, P0_7 to P0_10			25	mA
				P1_4 to P1_5, P8_0, P8_1, P8_3 to P8_9			19	mA
			Total (EVCC)				60	mA
		PgB	Per side	P18_0 to P18_3			20	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7, P11_15, P12_0 to P12_2			30	mA
				P10_0 to P10_5, P10_15, P11_0, P11_8 to P11_12			30	mA
			Total (BVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA
		PgA1	Total (A1VREF)				8	mA

Note: For detail of the definition of “side” and “total”, see **Section 47B.2.3, Port Current**.

(5) 100-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_4			–5	mA
				P0_0 to P0_3, P10_3 to P10_5			–25	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2			–11	mA
				P0_7 to P0_10, P8_3 to P8_9			–27	mA
				P10_6 to P10_9			–20	mA
				P10_10 to P10_14, P11_1 to P11_7			–30	mA
				P10_0 to P10_2			–15	mA
			Total (EVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_4			5	mA
				P0_0 to P0_6, P0_11 to P0_14, P10_3 to P10_5			30	mA
				JP0_0 to JP0_5, P8_2, P8_10 to P8_12			26	mA
				P0_7 to P0_10			20	mA
				P8_3 to P8_9			7	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7			25	mA
				P10_0 to P10_2			15	mA
			Total (EVCC)				60	mA
		PgA0	Total (A0VREF)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47B.2.3, Port Current**.

47B.4.3 Power Supply Currents

Condition: REGVCC, EVCC, BVCC, A0VREF and A1VREF total current. But the I/O buffer is stopped.

Products of CPU frequency 240 MHz max.

Item	Symbol	Condition				MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2				
RUN mode current	IDDR	Run (240 MHz)	Run	–40 to 150°C	Run (#1)		70	185	mA
				25°C	Stop (#1)		64		mA
RUN mode current (During data/code flash programming)	IDDR3	Run (240 MHz)	Run	–40 to 150°C	Run (#2)		90	205	mA
RUN mode current (With code flash background operation)	IDDRBGO	Run (240 MHz)	Run	–40 to 150°C	Run (#6)		90	205	mA
RUN mode current (HALT state)	IDDH	Run (240 MHz)	Run	–40 to 150°C	Run (#3)		67	183	mA

Products of CPU frequency 160 MHz max.

Item	Symbol	Condition				MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2				
RUN mode current	IDDR	Run (160 MHz)	Run	–40 to 150°C	Run (#1)		60	173	mA
				25°C	Stop (#1)		54		mA
RUN mode current (During data/code flash programming)	IDDR3	Run (160 MHz)	Run	–40 to 150°C	Run (#2)		80	193	mA
RUN mode current (With code flash background operation)	IDDRBGO	Run (160 MHz)	Run	–40 to 150°C	Run (#6)		80	193	mA
RUN mode current (HALT state)	IDDH	Run (160 MHz)	Run	–40 to 150°C	Run (#3)		57	171	mA

Products of CPU frequency 240 MHz max, 160 MHz max.

Item	Symbol	Condition				MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2				
STOP mode current	IDDS	Stop	Stop	–40 to 90°C	Stop (#2)		1.3	22	mA
				110°C	Stop (#2)			42	mA
				135°C	Stop (#2)			66	mA
DeepSTOP mode current	IDDDS	Power off	Power off	–40 to 85°C	Stop (#3)		50	700	µA
				105°C	Stop (#3)			1280	µA
				125°C	Stop (#3)			1840	µA
Cyclic RUN mode current	IDDCR	Run (HS IntOSC)	Stop	–40 to 90°C	Run (#4)		6.1	28	mA
				115°C	Run (#4)			47	mA
				135°C	Run(#4)			71	mA
Cyclic STOP mode current	IDDCS	Stop	Stop	–40 to 90°C	Run (#5)		1.4	23	mA
				110°C	Run (#5)			42	mA
				135°C	Run (#5)			66	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- REGVCC = EVCC = BVCC = A0VREF = A1VREF = 5.0 V
- AWOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V

Note 2. Operating condition of each peripheral function is shown in the table of next page.

Caution: It must be ensured that the junction temperature in the Ta range remains below Tj ≤ 150°C and does not exceed its limit under application conditions (thermal resistance, power supply current, peripheral current (if not included in power supply current), port output current and injection current).

Function		Run						Stop		
		(#1)	(#2)	(#3)	(#4)	(#5)	(#6)	(#1)	(#2)	(#3)
AWO	MainOSC	Run	Run	Run	Stop	Stop	Run	Run	Stop	Stop
	SubOSC	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	HS IntOSC	Run	Run	Run	Run	Stop	Run	Run	Stop	Stop
	FOUT	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	LPS	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	RRAM	Read/Write	Read/Write	No access	Fetch	No access	Read/Write	Read/Write	No access	No access
	WDTA0	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	TAUJ0, TAUJ2	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Run	Stop	Stop	Stop
	RTCA0	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Run	Stop	Stop	Stop
	CLMA0	Run	Run	Run	Run	Stop	Run	Stop	Stop	Stop
	CLMA1	Run	Run	Run	Stop	Stop	Run	Stop	Stop	Stop
	ADCA0	Run*1	Run*1	Run*1	Stop	Stop	Run*1	Stop	Stop	Stop
ISO	CPU	Run (PLL0)	Run (PLL0)	HALT (PLL0)	Run (HS IntOSC)	Stop	Run (PLL0)	Run (PLL0)	Stop	Power off
	ICUMD	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	DMA	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	PLL0	Run	Run	Run	Stop	Stop	Run	Run	Stop	
	PLL1	Run	Run	Run	Stop	Stop	Run	Run	Stop	
	Code flash (FLI0)	Fetch	Fetch	No access	No access	No access	Fetch	Fetch	No access	
	Code flash (FLI1)	Fetch	Fetch	No access	No access	No access	Write/Erase	Fetch	No access	
	Data flash	Read	Write/Erase	No access	No access	No access	No access	Read	No access	
	LRAM	Read/Write	Read/Write	No access	No access	No access	Read/Write	Read/Write	No access	
	GRAM	Read/Write*2	Read/Write*2	No access	No access	No access	Read/Write*2	Read/Write*2	No access	
	OSTMn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	WDTA1	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	TAUD0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAUBn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAUJ1, TAUJ3	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	TAPA, PIC	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	ENCA0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	PWM-diag	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RLIN3n	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RLIN24n	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	RCFDCn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	CSIGn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	CSIHn	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	RIICn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	FlexRay	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	ETNBn	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	SFMA0	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	KR	Wait	Wait	Wait	Stop	Stop	Wait	Stop	Stop	
	RSENTn	Run	Run	Run	Stop	Stop	Wait	Stop	Stop	
	CLMA2	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	CLMA3	Run	Run	Run	Stop	Stop	Run	Stop	Stop	
	ADCA1	Run	Run	Run	Stop	Stop	Run	Stop	Stop	

Note 1. T&H used.

Note 2. GRZF not used.

47B.4.4 Injection Currents

Table 47B.1 Definition of Pin Group (272-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR	REGVCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P3, P20
PgB	BVCC, BVSS	P10, P11, P12, P13, P21, P22
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18, P19
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47B.2 Definition of Pin Group (233-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR	REGVCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P3, P20
PgB	BVCC, BVSS	P10, P11, P12, P13
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18, P19
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47B.3 Definition of Pin Group (176-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR	REGVCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P2, P20
PgB	BVCC, BVSS	P10, P11, P12
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47B.4 Definition of Pin Group (144-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgR	REGVCC, AWOVSS	IP0_0
PgE	EVCC, EVSS	JP0, P0, P1, P20
PgB	BVCC, BVSS	P10, P11, P12
PgE’	EVCC, EVSS	P8, P9
PgB’	BVCC, BVSS	P18
PgA0	A0VREF, A0VSS	AP0
PgA1	A1VREF, A1VSS	AP1

Table 47B.5 Definition of Pin Group (100-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgE	EVCC, EVSS	JP0, P0, P10, P11
PgE’	EVCC, EVSS	P8, P9
PgA0	A0VREF, A0VSS	AP0

47B.4.4.1 Absolute Maximum Ratings

(1) 272/233/176/144-Pin Versions

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgB	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgB’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
		PgA1	Per pin			10	mA
			Total			60	mA
		PgR	Per pin			10	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgB	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgB’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA
		PgA1	Per pin			–10	mA
			Total			–60	mA
		PgR	Per pin			–10	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0 and PgA1, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

(2) 100-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

47B.4.4.2 DC Characteristics for Overload Current

(1) 272/233/176/144-Pin Versions

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgB	Per pin			2	mA
			Total			50	mA
		PgE’	Per pin			3	mA
			Total			20	mA
		PgB’	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
		PgA1	Per pin			3	mA
			Total			20	mA
		PgR	Per pin			2	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgB	Per pin			–2	mA
			Total			–50	mA
		PgE’	Per pin			–3	mA
			Total			–20	mA
		PgB’	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA
		PgA1	Per pin			–3	mA
			Total			–20	mA
		PgR	Per pin			–2	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

(2) 100-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgE'	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgE'	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

47B.4.5 Power Management Characteristics

47B.4.5.1 Regulator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	REGVCC		VPOC*[1]		5.5	V
Output voltage	AWOVCL	AWOVCL pin	1.15	1.25	1.35	V
	ISOVCL	ISOVCL pin	1.15	1.25	1.35	V
Capacitance	CAWOVCL	AWOVCL pin	0.07	0.10	0.13	μF
	CISOVCL	ISOVCL pin	0.07	0.10	0.13	μF
Equivalent series resistance for load capacitance	RVRAWO	for CAWOVCL			40*[2]	mΩ
	RVRISO	for CISOVCL			40*[2]	mΩ
Inrush current during power-on					250	mA

Note 1. “VPOC” means POC (power-on clear) detection voltage (typ. 2.85 V). For detail, see **Section 47B.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

Note 2. This is reference value.

47B.4.5.2 Voltage Detector (POC, LVI, VLVI, CVM) Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Detection voltage (REGVCC)	VPOC	POC			2.7	2.85	3.0	V
	VLVI0	LVI	Rise		3.87	4.0	4.13	V
			Fall		3.9	4.0	4.1	V
	VLVI1		Rise		3.57	3.7	3.83	V
			Fall		3.6	3.7	3.8	V
	VLVI2		Rise		3.37	3.5	3.63	V
			Fall		3.4	3.5	3.6	V
	VVLVI	VLVI			1.8	1.9	2.0	V
Detection voltage (ISOVCL)	VCVMH	CVM	High voltage$^{\text{Caution}}$		1.35	1.39	1.43	V
	VCVML*8		Low voltage$^{\text{Caution}}$		1.10	1.15	1.20	V
Response time	t_{D_POC1}*6	POC	At power-on (Rise)	*1			2	ms
				*2			6.3	ms
			After power-on (Rise)	*3			2	ms
				*4			5	ms
	t_{D_POC2}*7		After power-on (Fall)	*5			5	µs
	t_{D_LVI}	LVI					2	ms
	t_{D_VLVI}	VLVI		*3			2	ms
				*4			5	ms
	t_{D_CVM}	CVM			0.2		10	µs
Setup time	t_{S_LVI}	LVI	LVICNT0,1 bits are set to 1 (except 00_B), then LVI is ready to operate				80	µs
REGVCC minimum width	t_{W_POC}	POC			0.2			ms
	t_{W_LVI}	LVI			0.2			ms
	t_{W_VLVI}	VLVI			0.2			ms

Note 1. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms

Note 2. Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms

Note 3. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 20 V/ms

Note 4. Voltage slope (t_{VS}): 20 V/ms < t_{VS} ≤ 500 V/ms

Note 5. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 500 V/ms

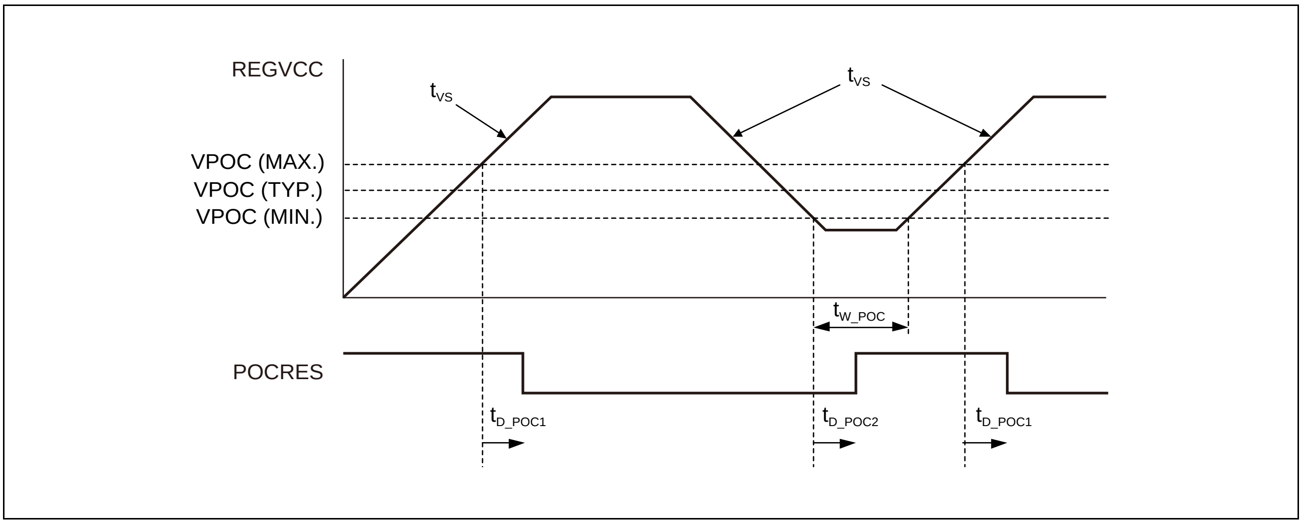
Note 6. t_{D_POC1} is the time from detection voltage to release of reset signal.

Note 7. t_{D_POC2} is the time from detection voltage to occurrence of reset signal.

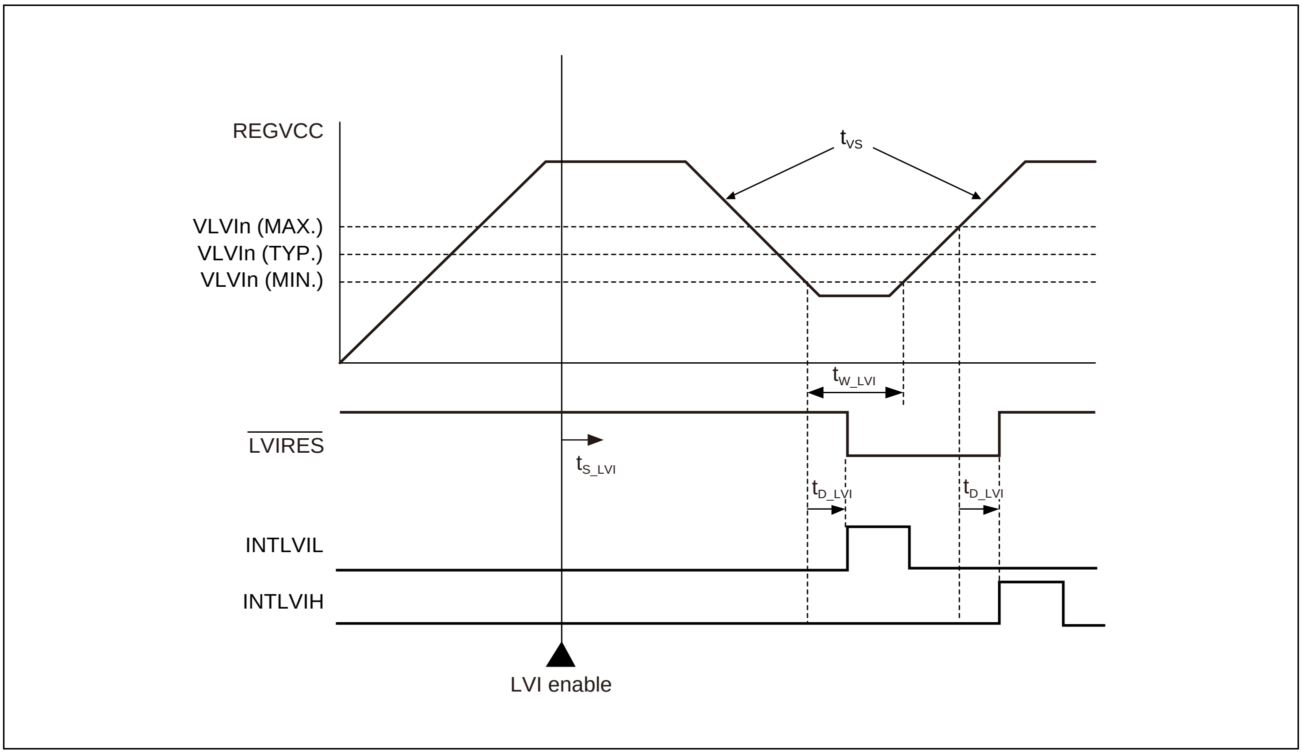
Note 8. The CVM monitors the internal voltage regulator output to ensure that ISOVCL is upper than specified minimum level.

Caution: A detection of the voltage ISOVCL outside the specified level of VCVMH and VCVML is not ensured by CVM.

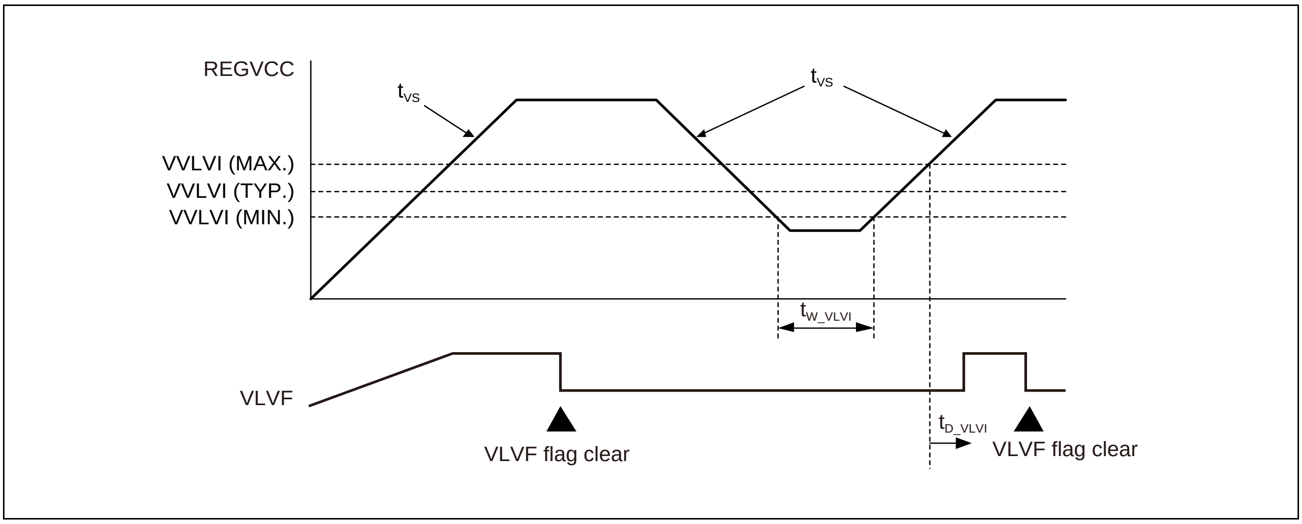
POC


REGVCC
t_{VS}
t_{VS}
VPOC (MAX.)
VPOC (TYP.)
VPOC (MIN.)
t_{W_POC}
POCRES
t_{D_POC1}
t_{D_POC2}
t_{D_POC1}

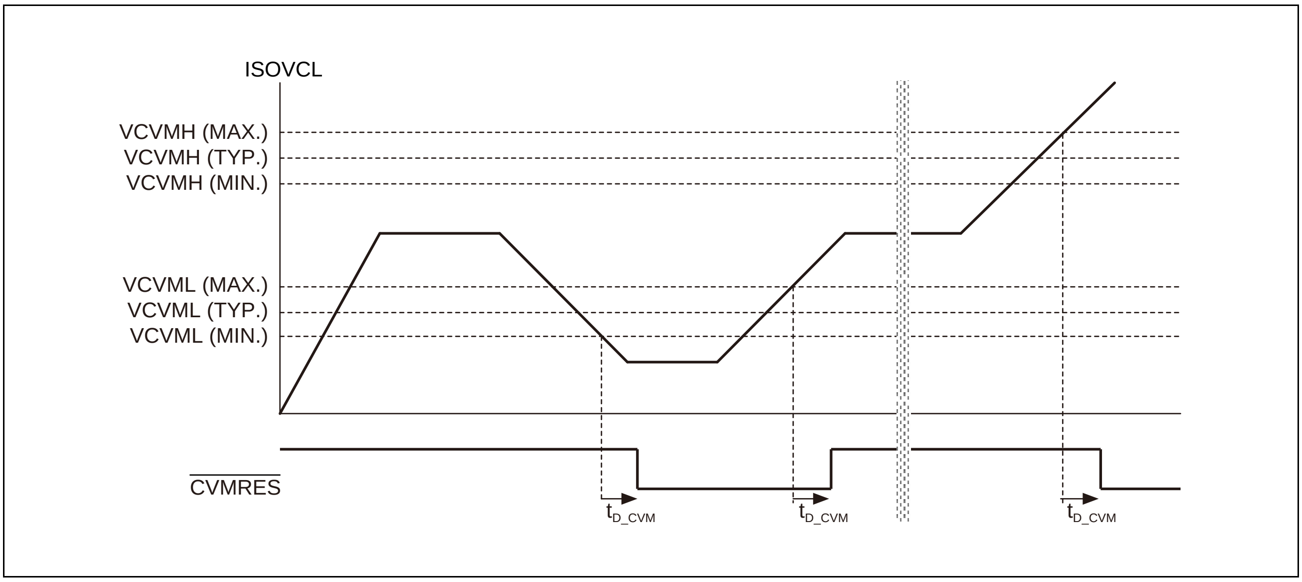
LVI


REGVCC
t_{VS}
VLVIn (MAX.)
VLVIn (TYP.)
VLVIn (MIN.)
t_{W_LVI}
$\overline{\text{LVIRES}}$
t_{S_LVI}
t_{D_LVI}
t_{D_LVI}
INTLVIL
INTLVIH
LVI enable

VLVI


REGVCC
tVS
tVS
VVLVI (MAX.)
VVLVI (TYP.)
VVLVI (MIN.)
tW_VLVI
VLVF
VLVF flag clear
tD_VLVI
VLVF flag clear

CVM


ISOVCL
VCVMH (MAX.)
VCVMH (TYP.)
VCVMH (MIN.)
VCVML (MAX.)
VCVML (TYP.)
VCVML (MIN.)
CVMRES
tD_CVM
tD_CVM
tD_CVM

47B.4.5.3 Power Up/Down Timing

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47B.6 In Case the $\overline{\text{RESET}}$ Pin is Used (for Normal Operating Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*1)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 µs/V)	V/ms
REGVCC ↑ and IOVCC*1 ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↑)	t_{HMDR}		1			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↓)	t_{SMDF}		0			µs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*1 ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

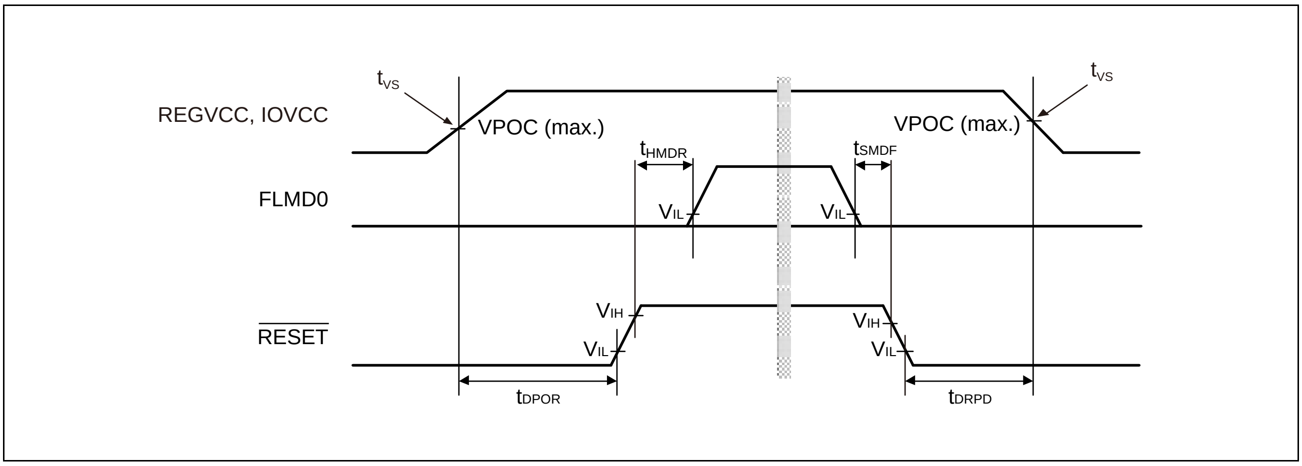


Table 47B.7 In Case the $\overline{\text{RESET}}$ Pin is Used (for Serial Programming Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*1)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*1 ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*1 ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

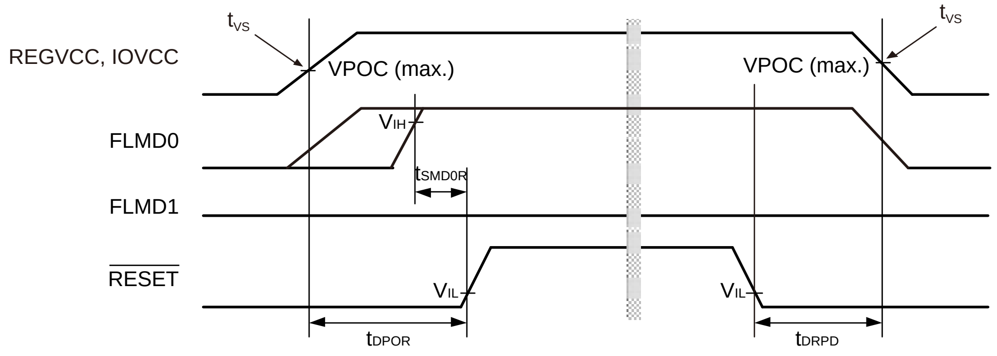


Table 47B.8 In Case the $\overline{\text{RESET}}$ Pin is Used (for Boundary Scan Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*^{1})	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 µs/V)	V/ms
REGVCC ↑ and IOVCC*^{1} ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1,MODE0,MODE1 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			µs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			µs
FLMD1, MODE0, MODE1, MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			µs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*^{1} ↓ delay time	t_{DRPD}		0			ms
$\overline{\text{DCUTRST}}$ input delay time (vs $\overline{\text{RESET}}$ ↑)	t_{DRTRST}		1			ms
$\overline{\text{RESET}}$ hold time (vs $\overline{\text{DCUTRST}}$ ↓)	t_{HRTRST}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

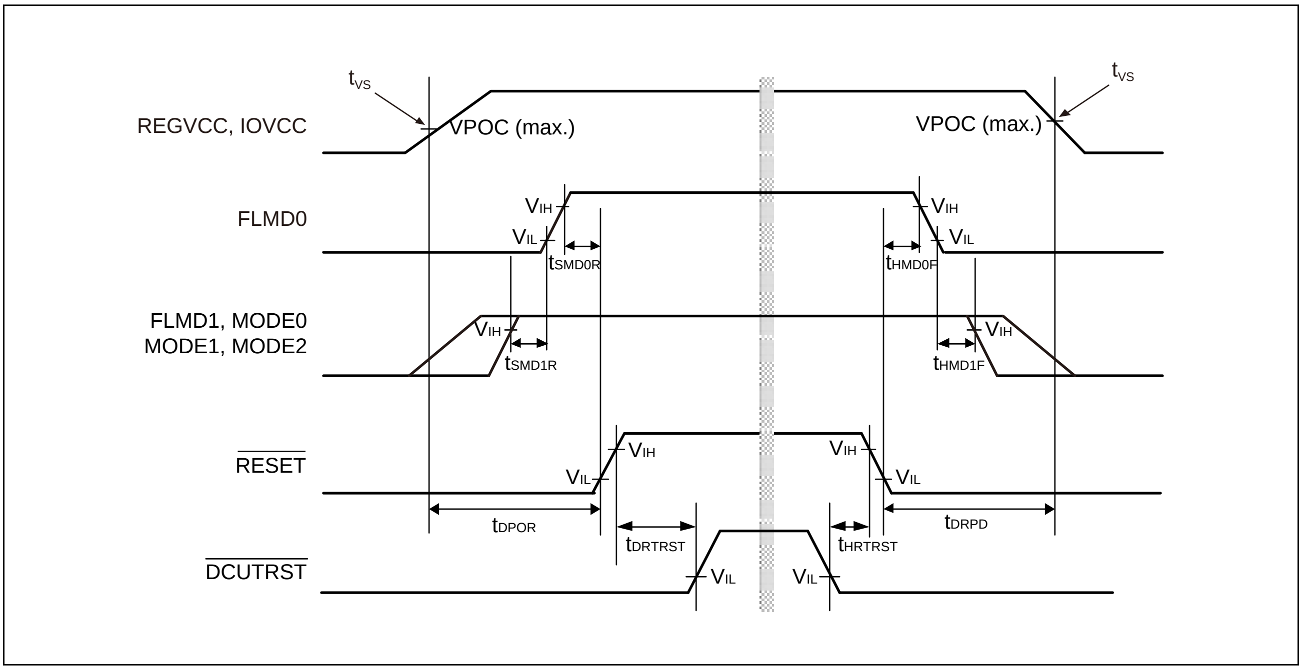


Table 47B.9 In Case the $\overline{\text{RESET}}$ Pin is Used (for User Boot Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC^{*1})	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC^{*1} ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1, MODE0, MODE1, MODE2 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			μs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			μs
FLMD1, MODE0, MODE1, MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			μs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, BVCC, A0VREF and A1VREF.

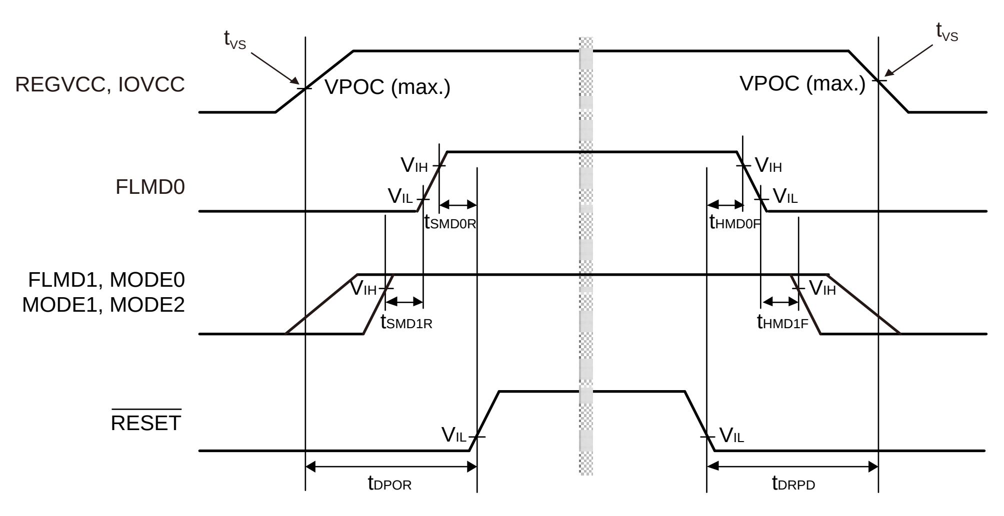
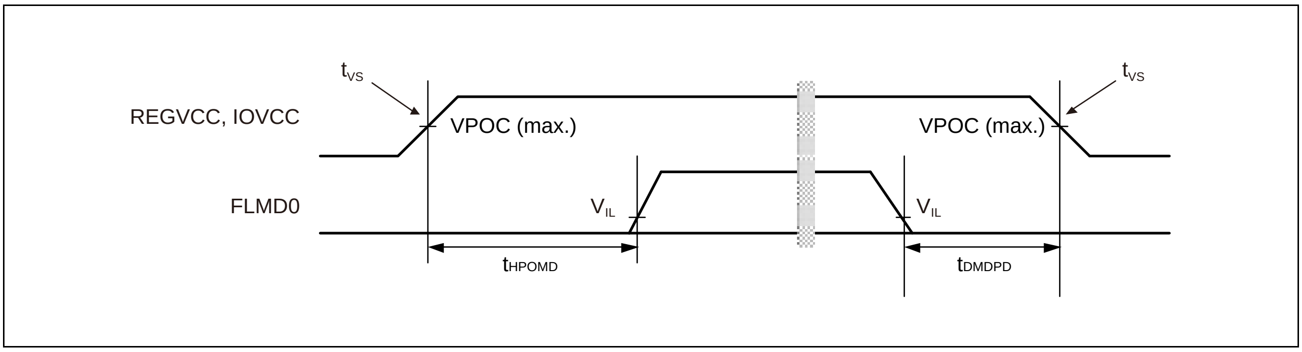


Table 47B.10 In Case the $\overline{\text{RESET}}$ Pin is Not Used and Fixed to High Level by Pull-up*[1]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*[2])	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*[2] ↑ to FLMD0 hold time	t_{HPOMD}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 ↓ to REGVCC ↓ and IOVCC*[2] ↓ delay time	t_{DMDPD}		1			μs

Note 1. This operating condition is available only in normal operation mode (include self-programming mode).
When the device is used in except normal operation mode, please use the $\overline{\text{RESET}}$ pin.

Note 2. IOVCC means EVCC, BVCC, A0VREF and A1VREF.



47B.4.5.4 CPU Reset Release Timing

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47B.11 In Case the $\overline{\text{RESET}}$ Pin is Not Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
REGVCC ↑ to CPU reset release*1	t_{DPCRR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms			2.58	ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms			8.3	ms

Note 1. This is reference value.

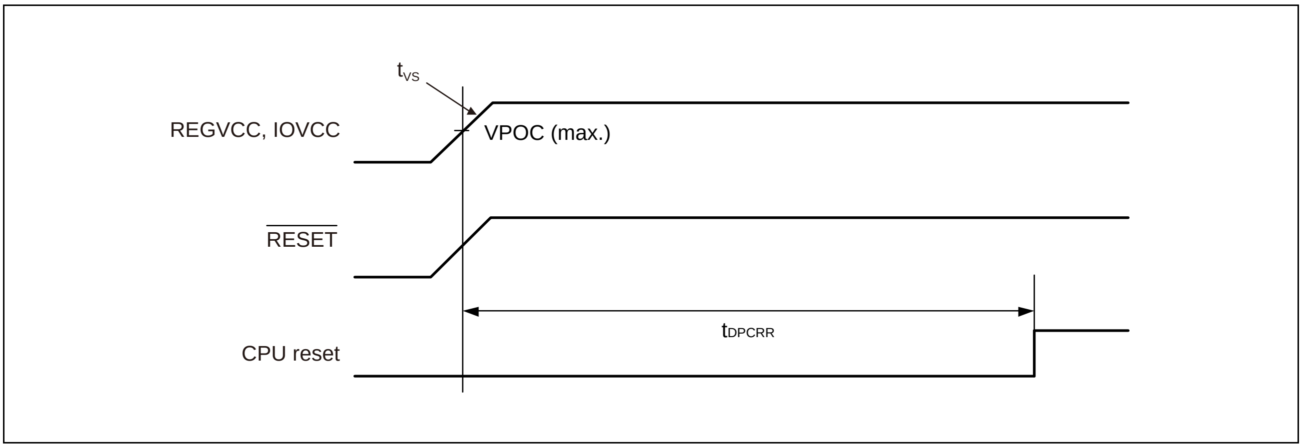
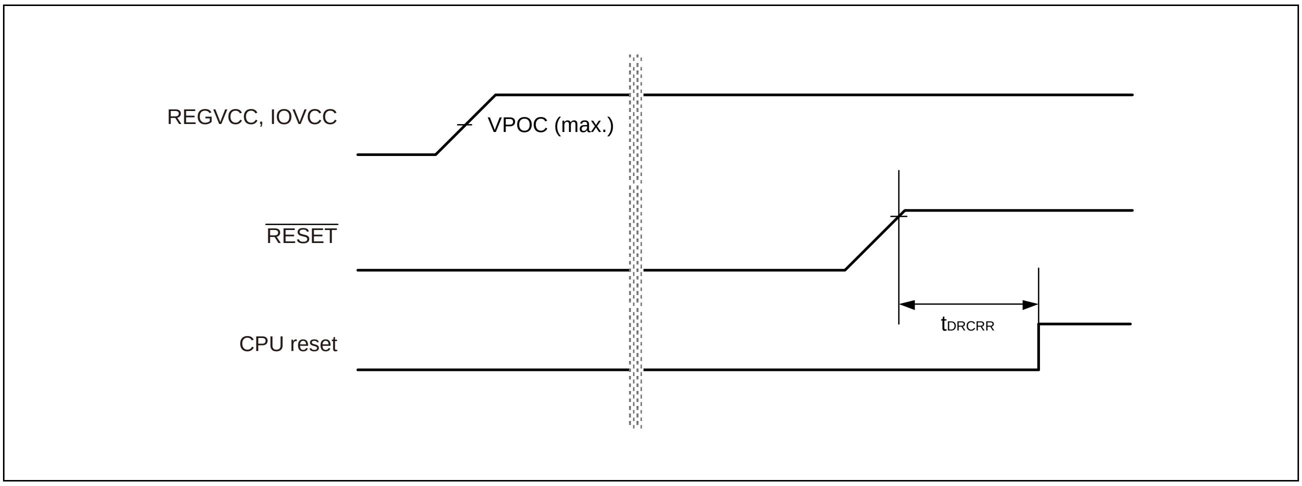


Table 47B.12 In Case the $\overline{\text{RESET}}$ Pin is Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ ↑ to CPU reset release*1	t_{DRCRR}				32*2	µs

Note 1. This is reference value.

Note 2. In case the time until releasing the $\overline{\text{RESET}}$ pin is longer than t_{DPCRR}.



47B.5 AC Characteristics

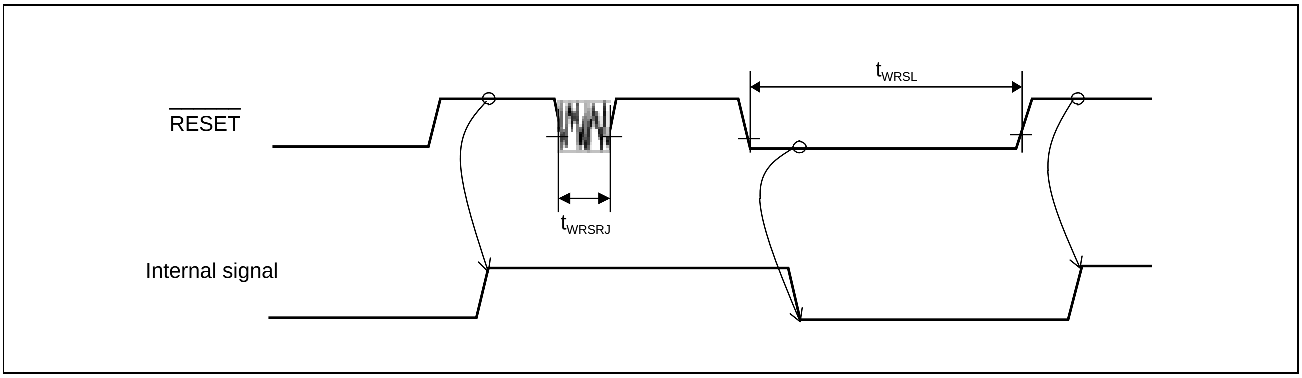
47B.5.1 $\overline{\text{RESET}}$ Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width*[1]	t_{WRSL}	Except power on	600			ns
$\overline{\text{RESET}}$ pulse rejection*[2]	t_{WRSRJ}		100			ns

Note 1. $\overline{\text{RESET}}$ input width is needed to ensure that the internal reset signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



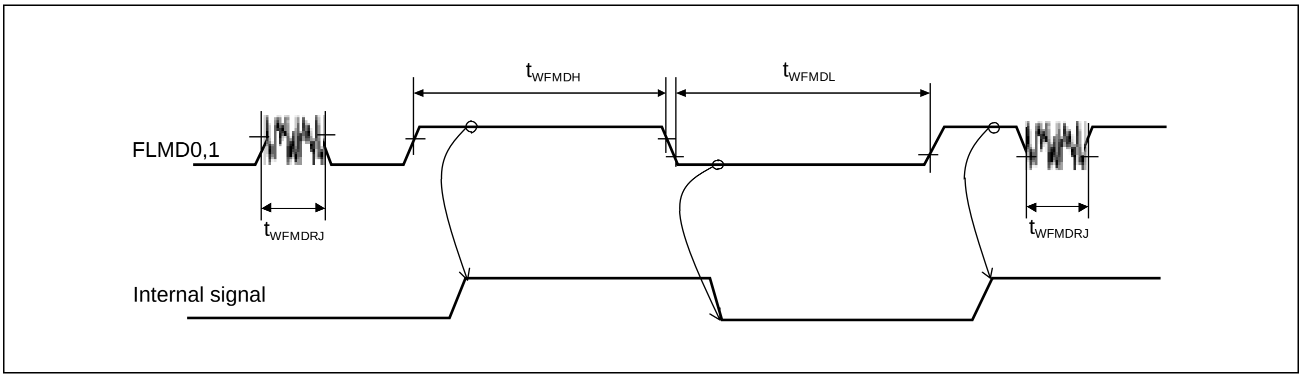
47B.5.2 Mode Timing

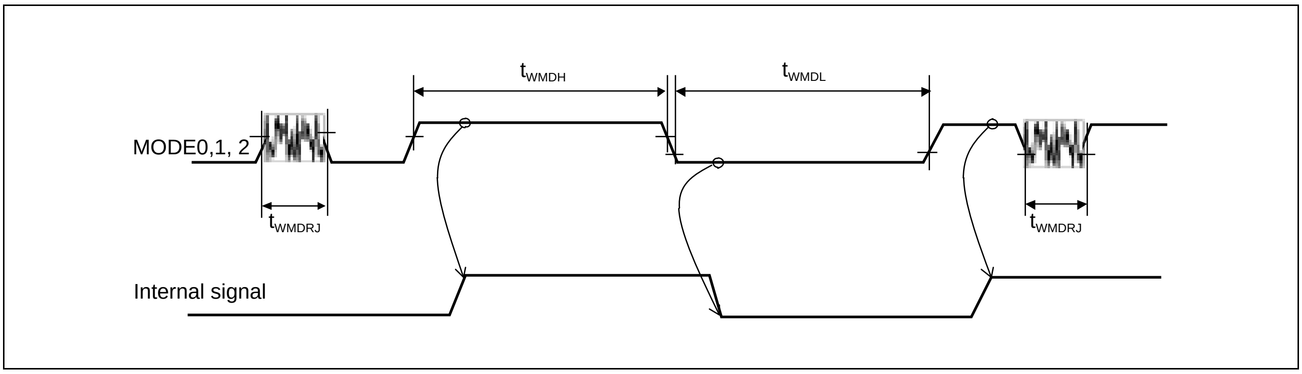
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0, 1 input high/low level width*1	t_{WFMDH}/ t_{WFMDL}		600			ns
FLMD0, 1 pulse rejection*2	t_{WFMDRJ}		100			ns
MODE0, 1, 2 input high/low level width*1	t_{WMDH}/ t_{WMDL}		600			ns
MODE0, 1, 2 pulse rejection*2	t_{WMDRJ}		100			ns

Note 1. FLMD0, 1 and MODE0, 1, 2 input width is needed to ensure that the internal mode signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.





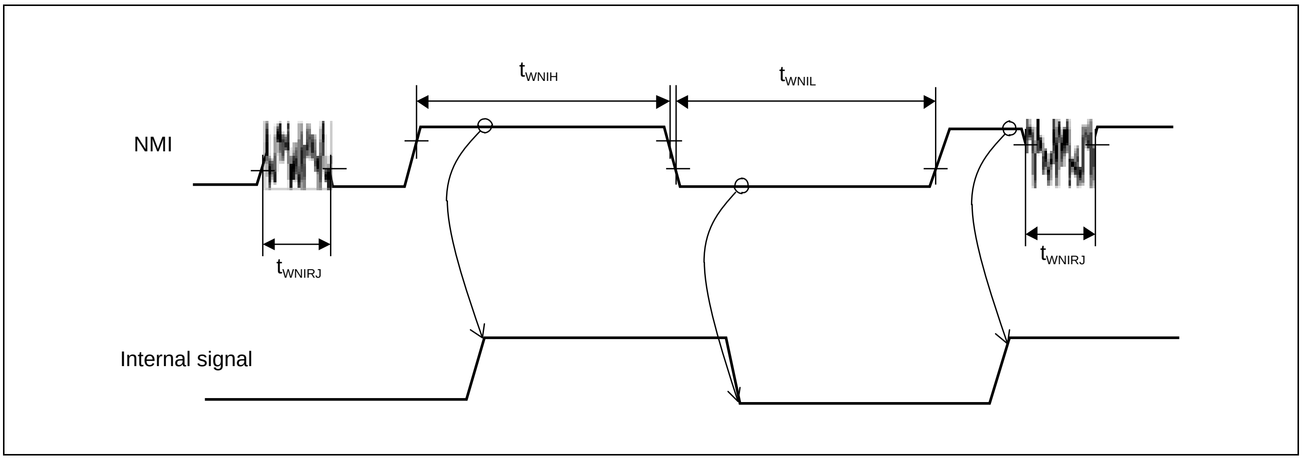
47B.5.3 Interrupt Timing

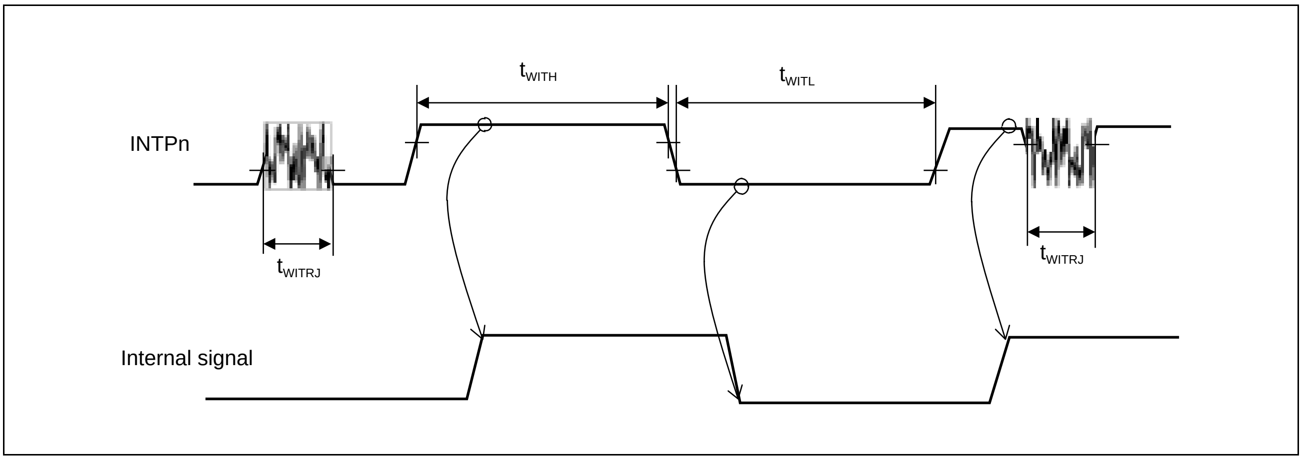
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
NMI input high/low level width*[1]	t_{WNIH}/ t_{WNIL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			µs
NMI pulse rejection*[2]	t_{WNIRJ}		100			ns
INTPn input high/low level width*[1]	t_{WITH}/ t_{WITL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			µs
INTPn pulse rejection*[2]	t_{WITRJ}		100			ns

Note 1. NMI and INTPn input width is needed to ensure that the internal interrupt signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



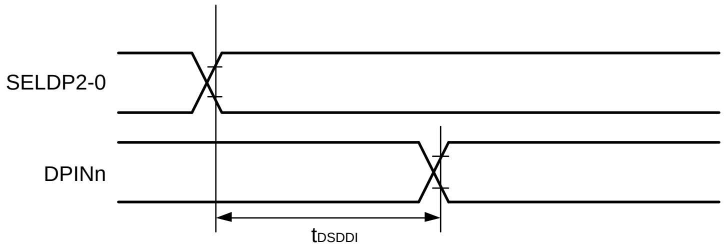


47B.5.4 Low Power Sampler (DPIN input) Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DPINn input delay time (vs SELDP2-0)	t_{DSDDI}				150	ns

Note: n = 7 to 0



47B.5.5 CSCXFOUT Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

<Output driver strength>

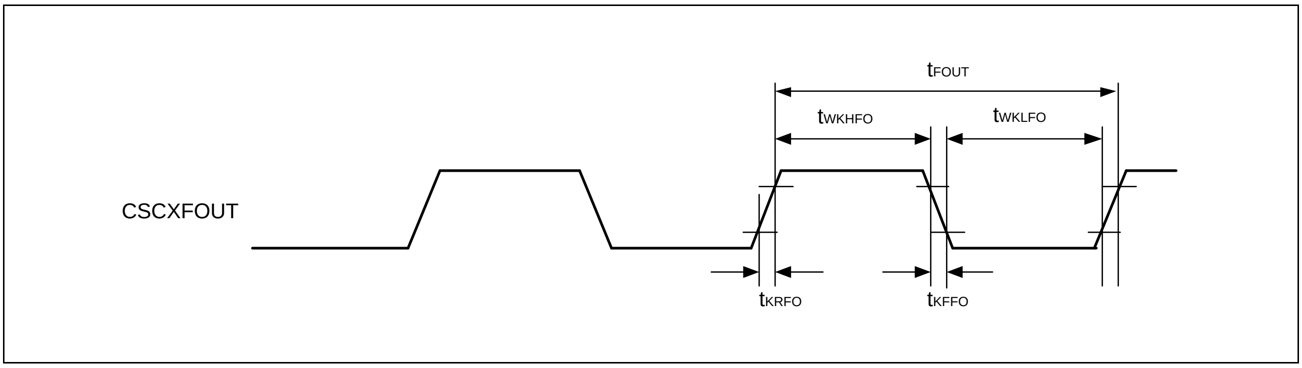
CSCXFOUT: Slow or fast mode (refer to the condition in the following table)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
CSCXFOUT output cycle	t_{FOUT}	Slow mode		100 (max. 10 MHz)			ns
		Fast mode		41.6 (max. 24 MHz)			ns
CSCXFOUT high level width	t_{WKHFO}	Slow mode	N: 1*1 or even value*2	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)*2,*3	t_{FOUT} × (N+1) / 2N – 37			ns
		Fast mode	N: 1*1 or even value*2	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)*2	t_{FOUT} × (N+1) / 2N – 10			ns
CSCXFOUT low level width	t_{WKLFO}	Slow mode	N: 1*1 or even value*2	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)*2,*3	t_{FOUT} × (N–1) / 2N – 37			ns
		Fast mode	N: 1*1 or even value*2	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)*2	t_{FOUT} × (N–1) / 2N – 10			ns
CSCXFOUT rise/ fall time	t_{KRFO}/ t_{KFFO}	Slow mode				37	ns
		Fast mode				10	ns

Note 1. When MainOSC, HS IntOSC, LS IntOSC or SubOSC is selected as source clock with the condition of N=1, the characteristics of output signal depends on the selected source clock. It is recommended to use output signal after evaluation on an actual environment.

Note 2. “N” is the value of “Clock divisor N” defined by FOUTDIV register.

Note 3. The selection of N = 3 is prohibited when slow mode is used.



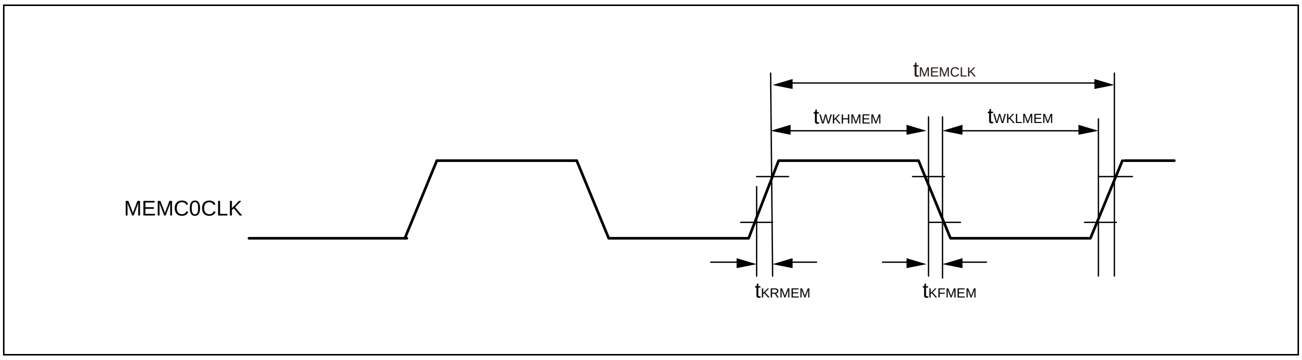
47B.5.6 MEMC0CLK Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Output driver strength>

MEMC0CLK pin: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MEMC0CLK output cycle	t_{MEMCLK}		33.4 (max.30 MHz)			ns
MEMC0CLK high / low level width	t_{WKHMEM} / t_{WKLMEM}		t_{MEMCLK} / 2 – 10			ns
MEMC0CLK rise / fall time	t_{KRMEM} / t_{KFMEM}				10	ns



47B.5.7 External Bus Timing

47B.5.7.1 MEMC0CLK Asynchronous

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

<Output driver strength>

MEMC0AD0-15, MEMC0A16-23, $\overline{\text{MEMC0CS3-0}}$, $\overline{\text{MEMC0BEN1-0}}$, $\overline{\text{MEMC0ASTB}}$, $\overline{\text{MEMC0WR}}$, and $\overline{\text{MEMC0RD}}$ pins: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Bus operational period	T		33.4 (max.30 MHz)			ns
Address[*4] setup time to $\overline{\text{MEMC0ASTB}}$ ↓	t_{SAST}	<1>	(1 + ASW) × T – 15			ns
Address (MEMC0AD15-0) hold time from $\overline{\text{MEMC0ASTB}}$ ↓	t_{HSTA}	<2>	(1 + AHW) × T – 15			ns
Address (MEMC0AD15-0) float delay time from $\overline{\text{MEMC0RD}}$ ↓	t_{FRDA}[*8]	<3>			9	ns
Address[*5] hold time from $\overline{\text{MEMC0RD}}$ ↑	t_{HRDA}	<4>	–1.5			ns
Data (MEMC0AD15-0) input delay time from $\overline{\text{MEMC0RD}}$ ↓	t_{DRDID}	<5>	9		(1 + w) × T – 35	ns
Data (MEMC0AD15-0) input hold time from $\overline{\text{MEMC0RD}}$ ↑	t_{HRDID}	<6>	0			ns
Delay time from $\overline{\text{MEMC0ASTB}}$ ↓ to $\overline{\text{MEMC0RD}}$ ↓	t_{DSTRD}	<7>	(1 + AHW) × T – 15			ns
Delay time from $\overline{\text{MEMC0ASTB}}$ ↓ to $\overline{\text{MEMC0WR}}$ ↓	t_{DSTWR}	<8>	(1 + AHW) × T – 15			ns
$\overline{\text{MEMC0RD}}$, $\overline{\text{MEMC0WR}}$ low level width	t_{WRDST}	<9>	(1 + w) × T – 10			ns
Data (MEMC0AD15-0) output delay time from $\overline{\text{MEMC0WR}}$ ↓	t_{DWROD}	<10>			11	ns
Address[*5] hold time from $\overline{\text{MEMC0WR}}$ ↑	t_{HWRA}	<11>	(1 + DHW) × T – 15			ns
Data (MEMC0AD15-0) output setup time to $\overline{\text{MEMC0WR}}$ ↑	t_{SODWR}	<12>	(1 + w) × T – 15			ns
Data (MEMC0AD15-0) output hold time from $\overline{\text{MEMC0WR}}$ ↑	t_{HWROD}	<13>	(1 + DHW) × T – 15			ns
$\overline{\text{MEMC0WAIT}}$ setting delay from $\overline{\text{MEMC0ASTB}}$ ↓	t_{SSTWT1}	<14>			(AHW + DPW) × T – 24	ns
	t_{SSTWT2}	<15> DEW ≥ 1			(AHW + DPW + DEW) × T – 24	ns
$\overline{\text{MEMC0WAIT}}$ hold time from $\overline{\text{MEMC0ASTB}}$ ↓	t_{HSTWT1}	<16>	(AHW + DPW + DEW – 1) × T – 9			ns
	t_{HSTWT2}	<17> DEW ≥ 1	(AHW + DPW + DEW) × T – 9			ns

Note 1. ASW means the number of address setup wait for multiplex bus.

Note 2. AHW means the number of address hold wait for multiplex bus.

Note 3. DPW means the number of programmable data wait for multiplex bus.
DEW means the number of external data wait for multiplex bus.
"w" means the sum of DPW and DEW.

Note 4. t_{CPUCLK}: CPU clock period.

Note 5. DHW means the number of data hold wait for multiplex bus.

Note 6. Address means MEMC0AD15-0, MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$, and $\overline{\text{MEMC0BEN1-0}}$.
272-pin product supports 24-bit address. 233/176-pin products support 23-bit address.

Note 7. Address means MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$, $\overline{\text{MEMC0BEN1-0}}$, and $\overline{\text{MEMC0ASTB}}$.
272-pin product supports 24-bit address. 233/176-pin products support 23-bit address.

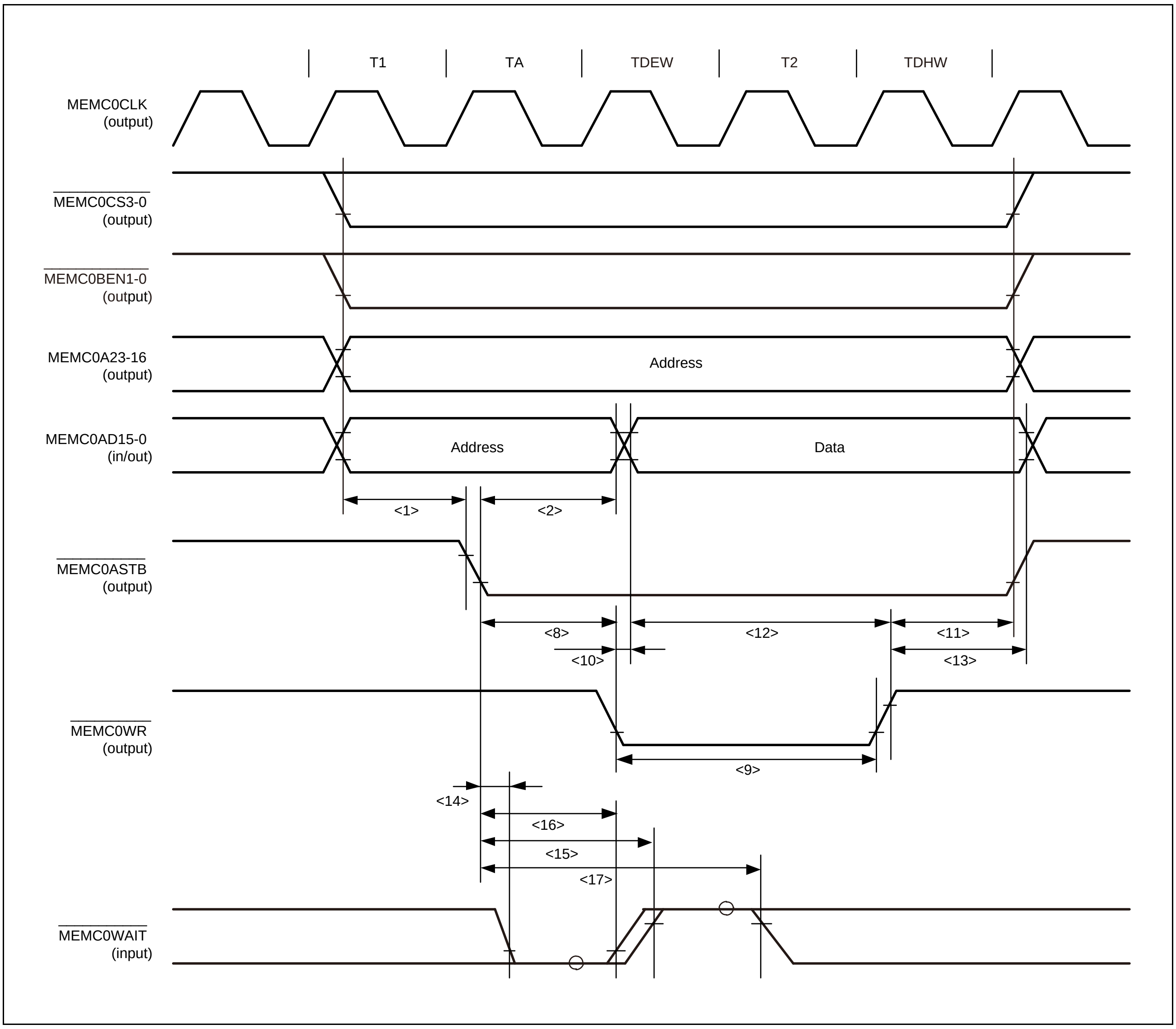
Note 8. t_{FRDA} means the period from output off to Hi-z for MEMC0AD15-0.

NOTE

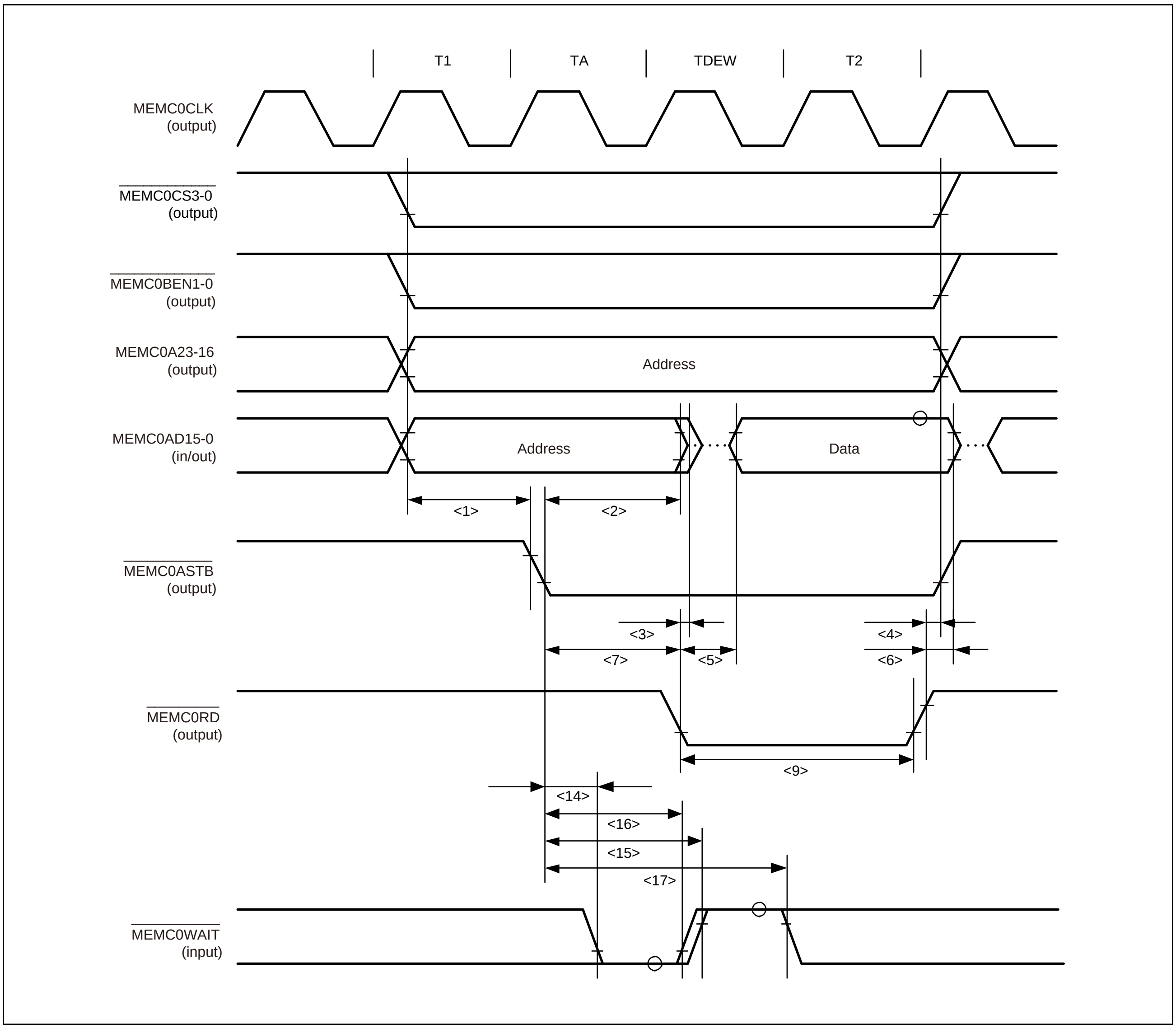
When the bus period (T) is shorter than 44 ns, t_{DRDID} spec requires at least 1data wait. (w = 1)

---

(1) Multiplex Write Cycle (Asynchronous; 1 Data Wait)



(2) Multiplex Read Cycle (Asynchronous; 1 Data Wait)



47B.5.7.2 MEMC0CLK Synchronous

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

<Output driver strength>

MEMC0AD0-15, MEMC0A16-23, $\overline{\text{MEMC0CS3-0}}$,
$\overline{\text{MEMC0BEN1-0}}$, $\overline{\text{MEMC0ASTB}}$, $\overline{\text{MEMC0WR}}$, and $\overline{\text{MEMC0RD}}$ pins: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Bus operational period	T		33.4 (max.30MHz)			ns
Delay time from MEMC0CLK ↑ to address*[1]	t_{DKA}	<18>	–0.5		15	ns
Delay time from MEMC0CLK ↑ to address (MEMC0AD15-0) float	t_{FKA}*[2]	<19>	0		12	ns
Delay time from MEMC0CLK ↑ to $\overline{\text{MEMC0ASTB}}$	t_{DKST}	<20>	0		11	ns
Delay time from MEMC0CLK ↑ to $\overline{\text{MEMC0RD}}$ and $\overline{\text{MEMC0WR}}$	t_{DKRDWR}	<21>	–2.5		6	ns
Data (MEMC0AD15-0) input setup time (from MEMC0CLK ↑)	t_{SIDK}	<22>	29			ns
Data (MEMC0AD15-0) input hold time (from MEMC0CLK ↑)	t_{HKID}	<23>	2.5			ns
Data (MEMC0AD15-0) output delay time (from MEMC0CLK ↑)	t_{DKOD}	<24>			15	ns
$\overline{\text{MEMC0WAIT}}$ setup time (to MEMC0CLK ↑)	t_{SWTK}	<25>	T + 22			ns
$\overline{\text{MEMC0WAIT}}$ hold time (from MEMC0CLK ↑)	t_{HKWT}	<26>	–T – 5			ns

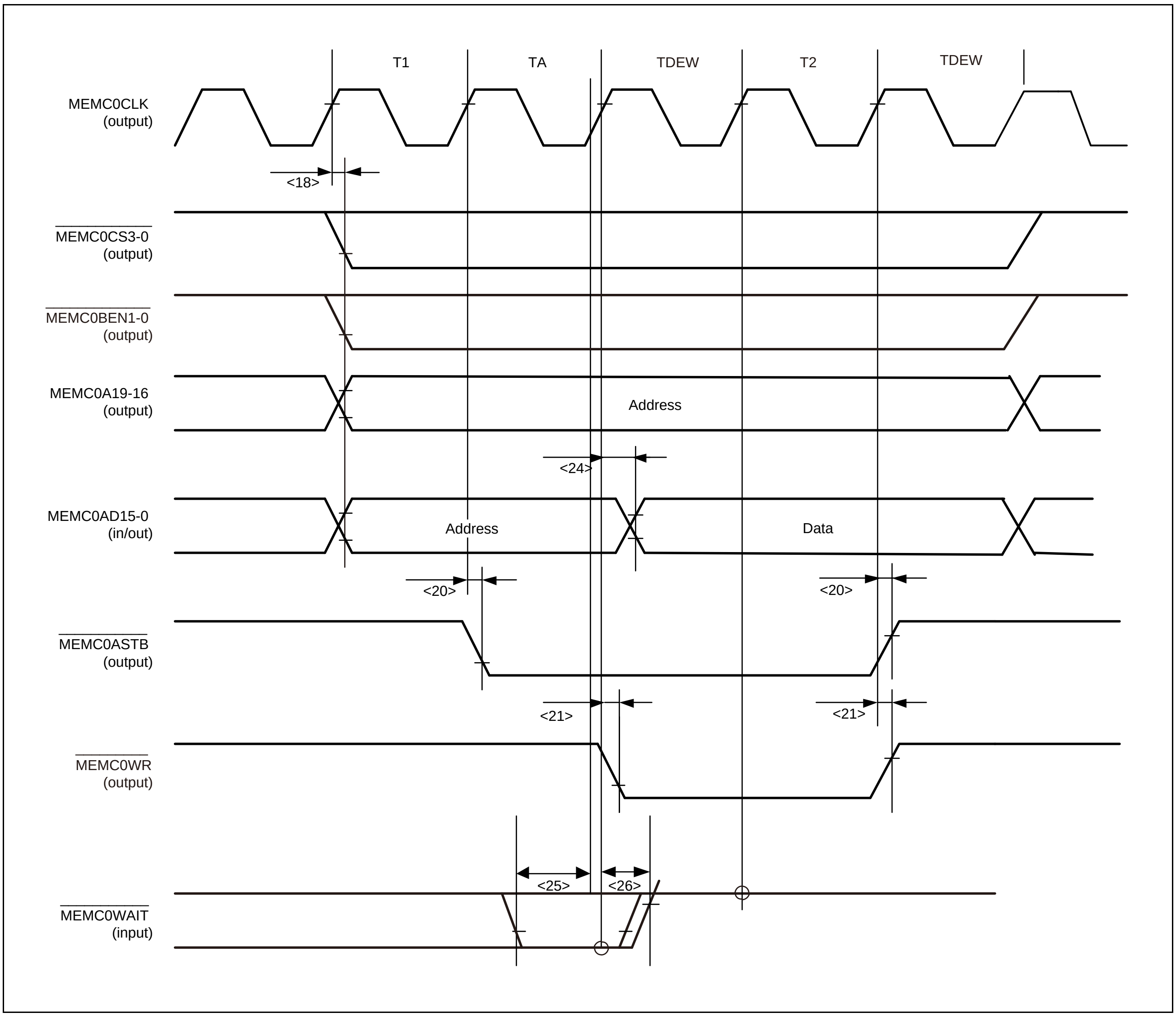
Note 1. Address means MEMC0AD15-0, MEMC0A23-16, $\overline{\text{MEMC0CS3-0}}$ and $\overline{\text{MEMC0BEN1-0}}$.
272-pin product supports 24-bit address. 233/176-pin products support 23-bit address.

Note 2. t_{FKA} means the period from output off to Hi-z for MEMC0AD15-0.

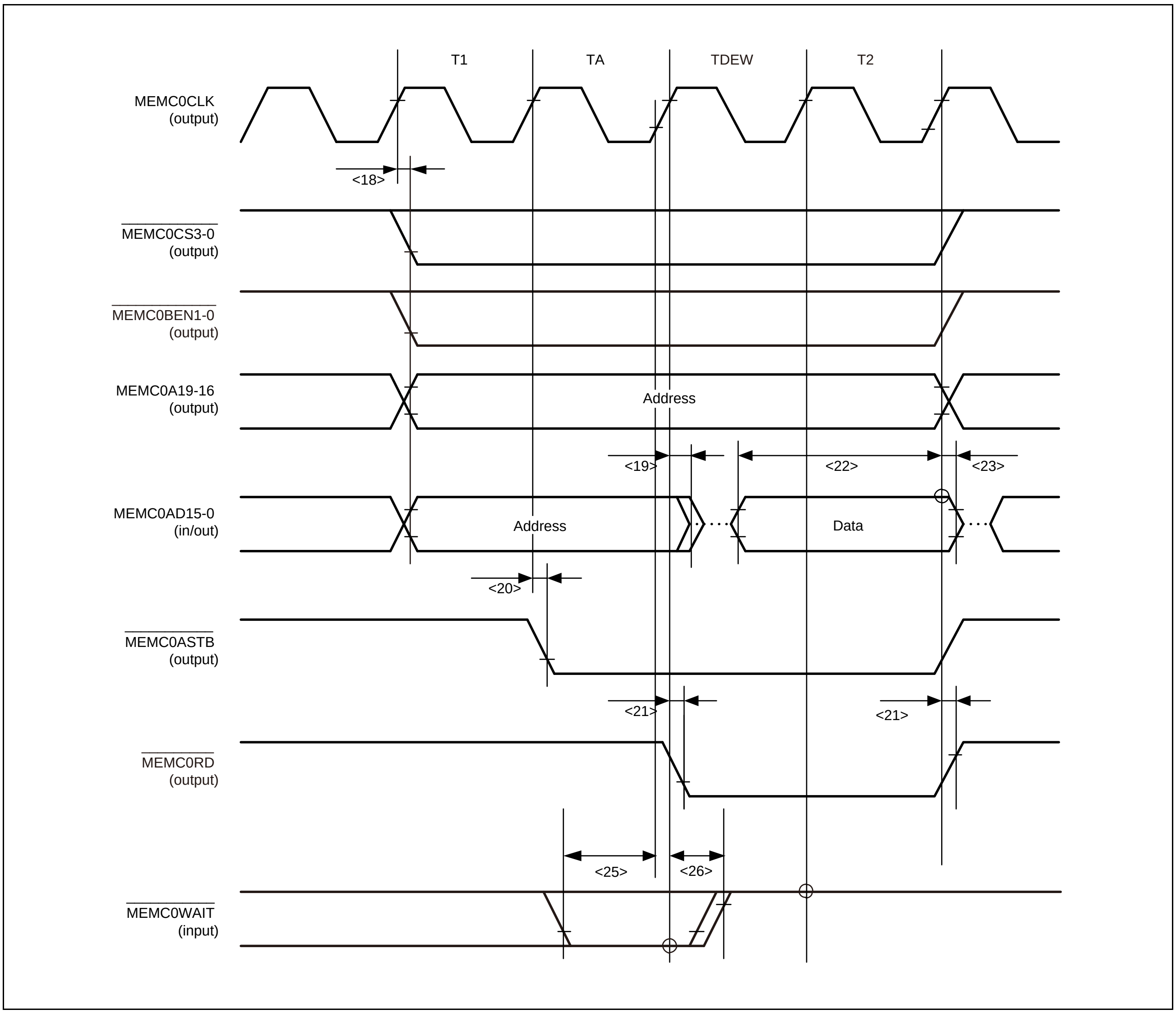
NOTE

When the bus period (T) is shorter than 44 ns, t_{DRDID} spec requires at least 1data wait. (w = 1)

(1) Multiplex Write Cycle (Synchronous; 1 Data Wait)



(2) Multiplex Read Cycle (Synchronous; 1 Data Wait)



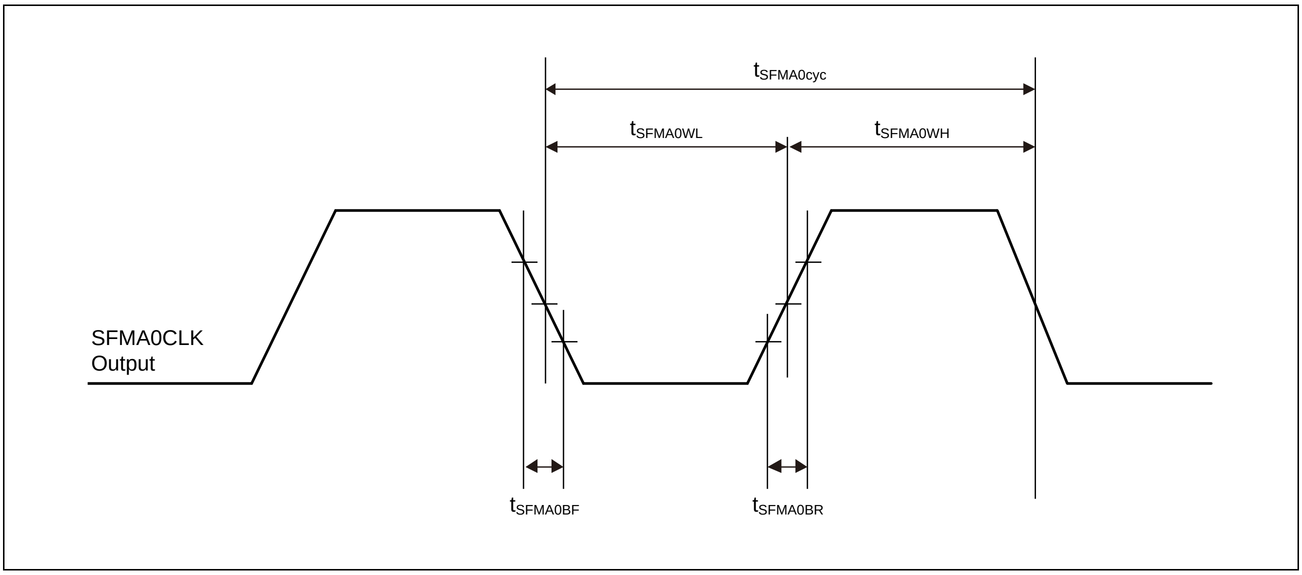
47B.5.8 SFMA Timing

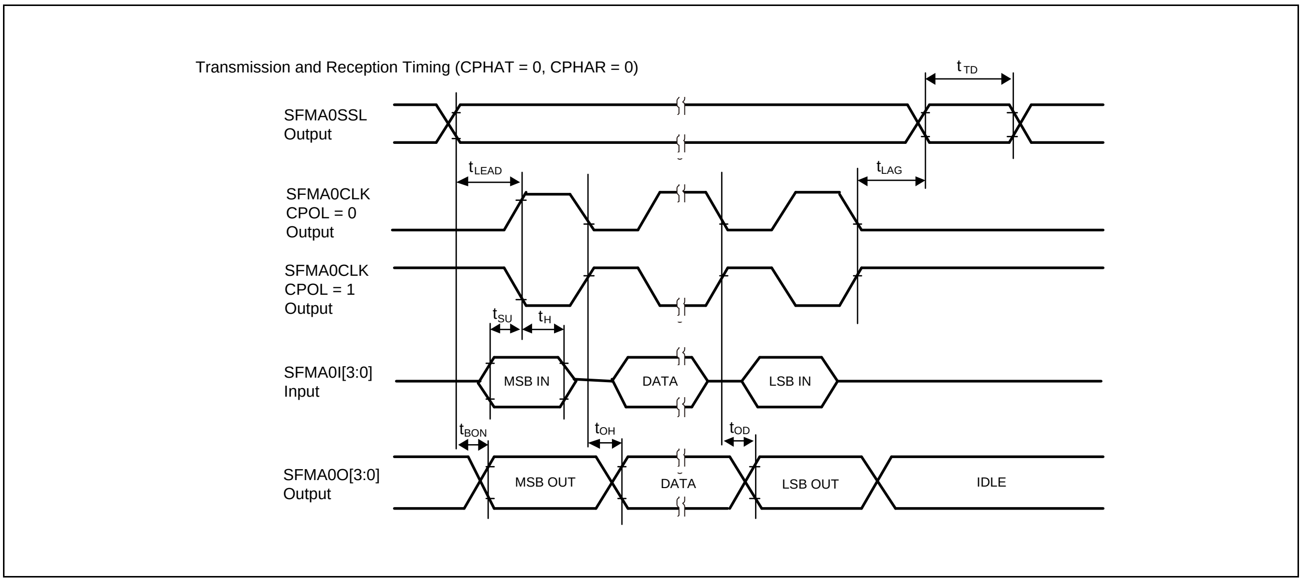
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to 3.6 V, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

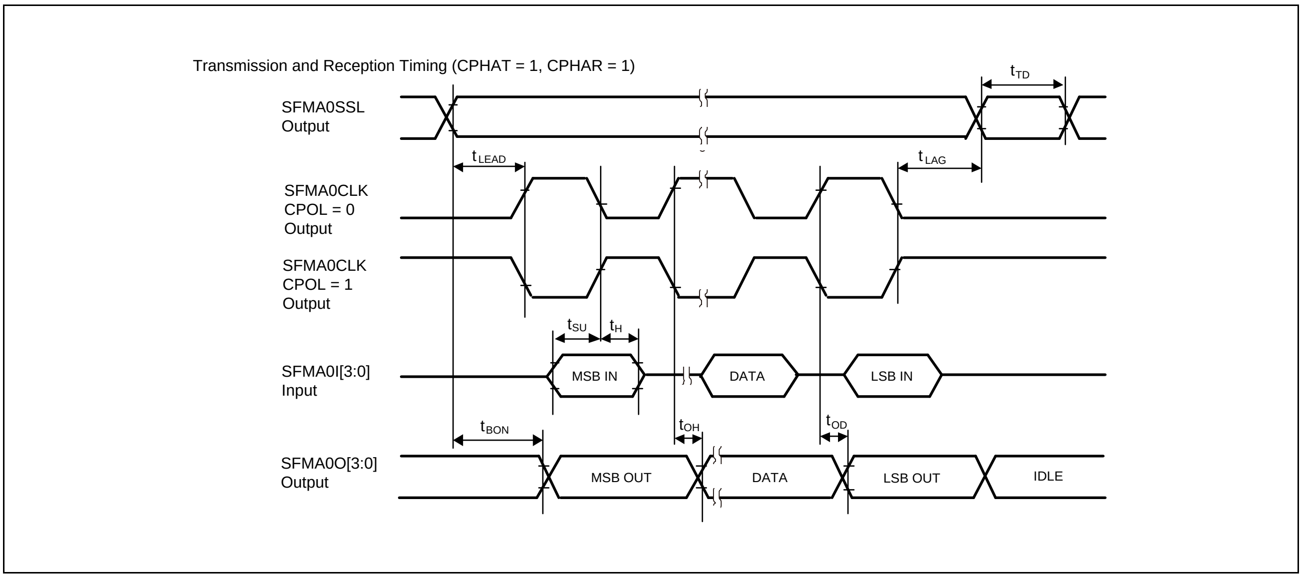
<Output driver strength>

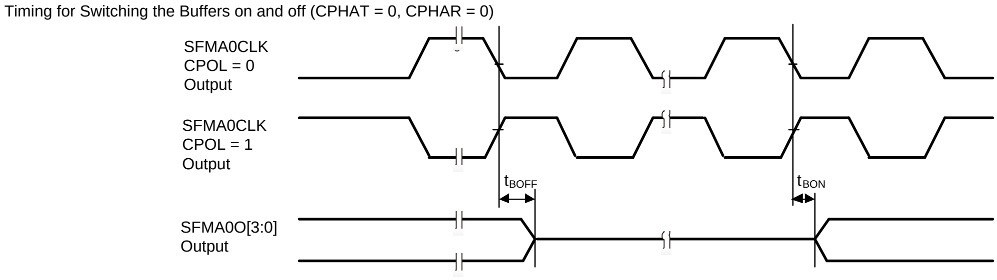
SFMA0CLK, SFMA0SSL, and SFMA0O[3:0] pins: Fast mode

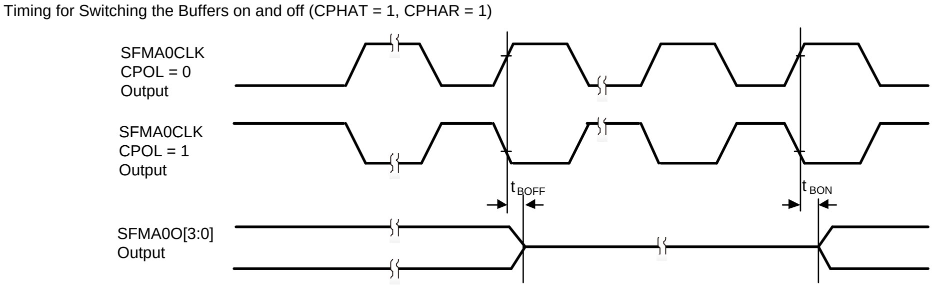
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SFMA0CLK clock cycle	$t_{SFMA0cyc}$		25			ns
SFMA0CLK high pulse width	$t_{SFMA0WH}$		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
SFMA0CLK low pulse width	$t_{SFMA0WL}$		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
SFMA0CLK rise time	t_{SFMA0R}				4.5	ns
SFMA0CLK fall time	t_{SFMA0F}				4.5	ns
Data input setup time	t_{SU}		13.0			ns
Data input hold time	t_{H}		0.0			ns
SFMA0SSL setup time	t_{LEAD}		1 × $t_{SFMA0cyc}$ – 5		8 × $t_{SFMA0cyc}$	ns
SFMA0SSL hold time	t_{LAG}		1.5 × $t_{SFMA0cyc}$		8.5 × $t_{SFMA0cyc}$ + 5	ns
Continuous transfer delay time	t_{TD}		1 × $t_{SFMA0cyc}$		8 × $t_{SFMA0cyc}$	ns
Data output delay time	t_{OD}				3.6	ns
Data output hold time	t_{OH}		–1.6			ns
Data output buffer on time	t_{BON}				3.6	ns
Data output buffer off time	t_{BOFF}		–7.0		0	ns




Transmission and Reception Timing (CPHAT = 0, CPHAR = 0)


Transmission and Reception Timing (CPHAT = 1, CPHAR = 1)





47B.5.9 Reserved

47B.5.10 CSI Timing

47B.5.10.1 CSIG Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47B.13 CSIG Timing (Master Mode)

<Output driver strength>

CSIGnSO, CSIGnSC (output): Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYMGn}		100			ns
CSIGnSC high level width	t_{KWHMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSC low level width	t_{KWLMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSIMGn}		30			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSIMGn}		0			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOMGn}				7	ns
CSIGnRYI setup time (vs. CSIGnSC)	t_{SRYIGn}	CSIGnCTL1.CSIGnSIT = x CSIGnCTL1.CSIGnHSE = 1	2 × t_{KCYGn} + 25			ns
CSIGnRYI high level width	t_{WRYIGn}	CSIGnCTL1.CSIGnHSE = 1	t_{KCYGn} + 5			ns

Note: n = 0 to 3

Table 47B.14 CSIG Timing (Slave Mode)

<Output driver strength>

CSIGnSO: Fast mode

CSIGnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYSGn}		200			ns
CSIGnSC high level width	t_{KWHSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSC low level width	t_{KWLSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSISGn}		20			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSISGn}		t_{KCYGn} + 5			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOSGn}				30	ns
CSIGnRYO output delay	t_{SRYOGn}				38	ns
$\overline{\text{CSIGnSSI}}$ setup time (vs. CSIGnSC)	$t_{SSSISGn}$		0.5 × t_{KCYSGn} – 5			ns
$\overline{\text{CSIGnSSI}}$ hold time (vs. CSIGnSC)	$t_{HSSISGn}$		t_{KCYGn} + 5			ns

Note: n = 0 to 3

47B.5.10.2 CSIH Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47B.15 CSIH Timing (Master Mode: 10 Mbps)

<Output driver strength>

CSIHnSO, CSIHnSC (output): Fast mode (CL = 100pF@n=0 / 50pF@n=1-3)

CSIHnCSSx: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYMHn}		100			ns
CSIHnSC high level width	t_{KWHMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSC low level width	t_{KWLMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	19			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	14			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	0			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	$t_{KCYHn}/2$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOMHn}				7	ns
CSIHnRYI setup time (vs. CSIHnSC)	t_{SRYIHn}	CSIHnCTL1.CSIHnSIT = x CSIHnCTL1.CSIHnHSE = 1	2 × t_{KCYHn} + 25			ns
CSIHnRYI high level width	t_{WRYIHn}	CSIHnCTL1.CSIHnHSE = 1	t_{KCYHn} + 5			ns
CSIHnCSS0-7 inactive width	$t_{WSCSBHn}$		CSIDLE × t_{KCYMHn} – 15			ns
CSIHnCSS0-7 setup time (vs. CSIHnSC)	$t_{SSCSBHn0}$	CSIHnCFGx.CSIHnDAP = 0	CSSETUP × t_{KCYMHn} – 23			ns
	$t_{SSCSBHn1}$	CSIHnCFGx.CSIHnDAP = 1	(CSSETUP + 0.5) × t_{KCYMHn} – 23			ns
CSIHnCSS0-7 hold time (vs. CSIHnSC)	$t_{HSCSBHn0}$	CSIHnCTL1.CSIHnSIT = 0	CSSHOLD × t_{KCYMHn} – 5			ns
	$t_{HSCSBHn1}$	CSIHnCTL1.CSIHnSIT = 1	(CSSHOLD + 0.5) × t_{KCYMHn} – 5			ns

Note: n = 0 to 3

NOTE

CSIDLE: Setting value of CSIHnCFGx.CSIHnIDx[2:0]

CSSETUP: Setting value of CSIHnCFGx.CSIHnSPx[3:0]

CSSHOLD: Setting value of CSIHnCFGx.CSIHnHDx[3:0]

x: Depends on number of the chip select signals.

CAUTION

When the serial clock level is changed during the communication (CSIHnCFGx.CSIHnCKPx) and the IDLE has a setting of 0.5 transmission clock cycles, an inactive width time $t_{WSCSBHn}$ of "0.5 × t_{KCYMHn}" is added.

Table 47B.16 CSIH Timing (Slave Mode: 5 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		200			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		20			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				30	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			38	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$38 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 0 to 3

Table 47B.17 CSIH Timing (Slave Mode: 8 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

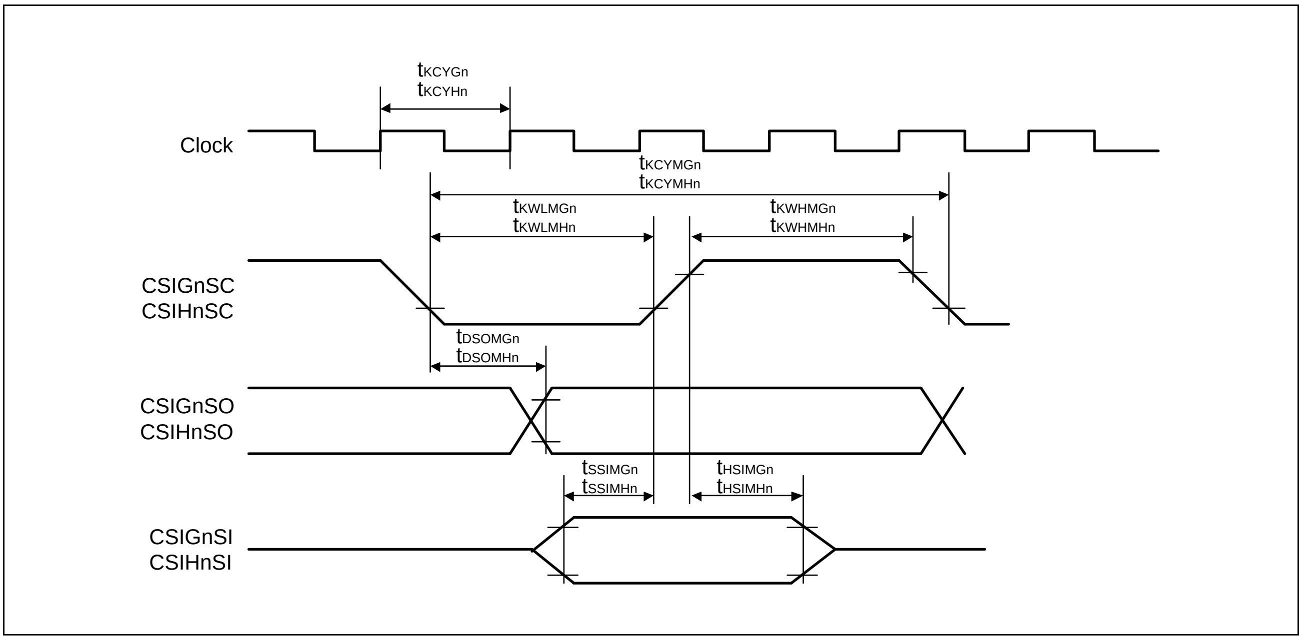
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		125			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		12.5			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				25	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			27	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$27 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 2 (Only for CSIH2)

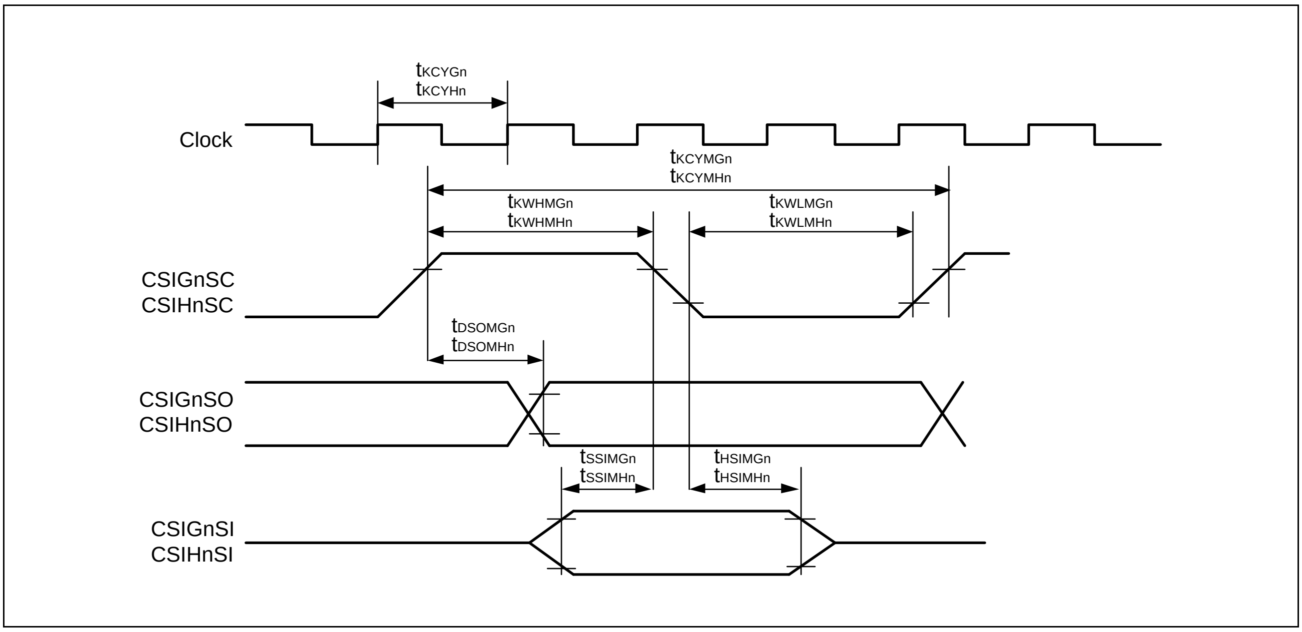
(1) SC/SI/SO

Master mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

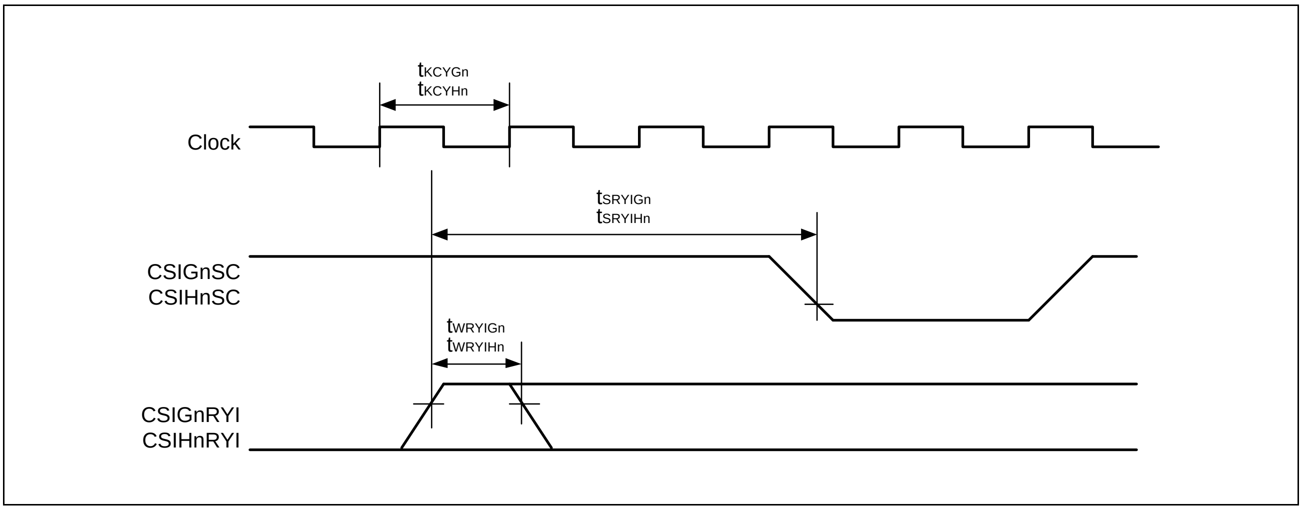


(2) RYI

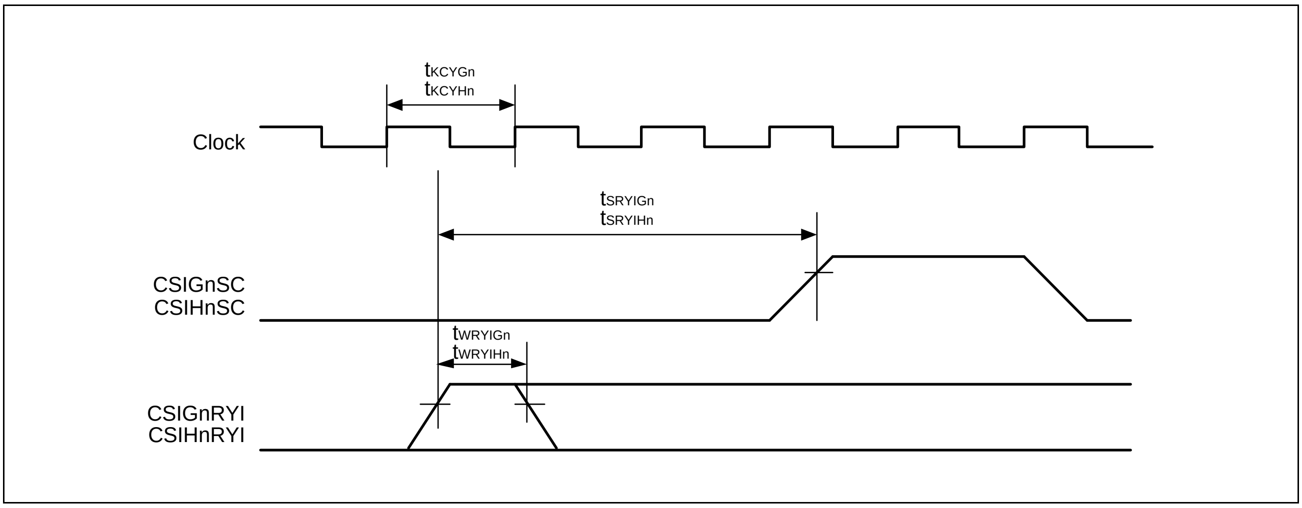
Master mode:

- CSIG: Only master mode (CSIGnCTL1: CSIGnHSE = 1, CSIGnCTL1: CSIGnSIT = 0)
- CSIH: Only master mode (CSIHnCTL1: CSIHnHSE = 1, CSIHnCTL1: CSIHnSIT = 0)

  - CSIG (CSIGnCTL1: CSIGnCKR = 0)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 0)



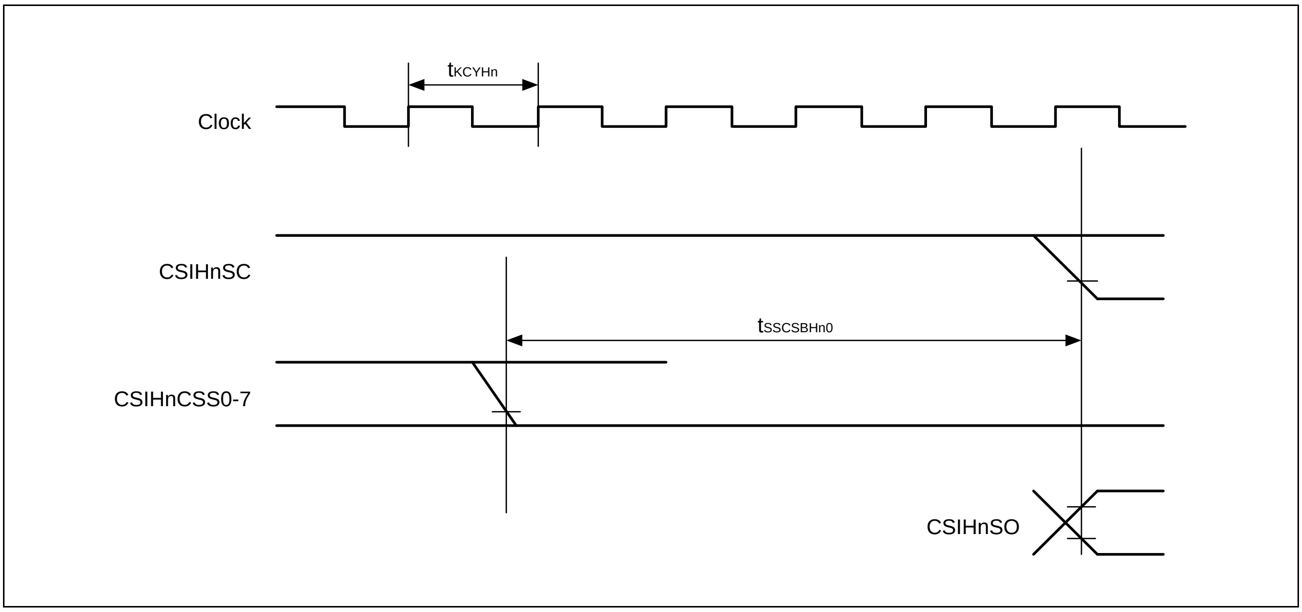
  - CSIG (CSIGnCTL1: CSIGnCKR = 1)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 1)



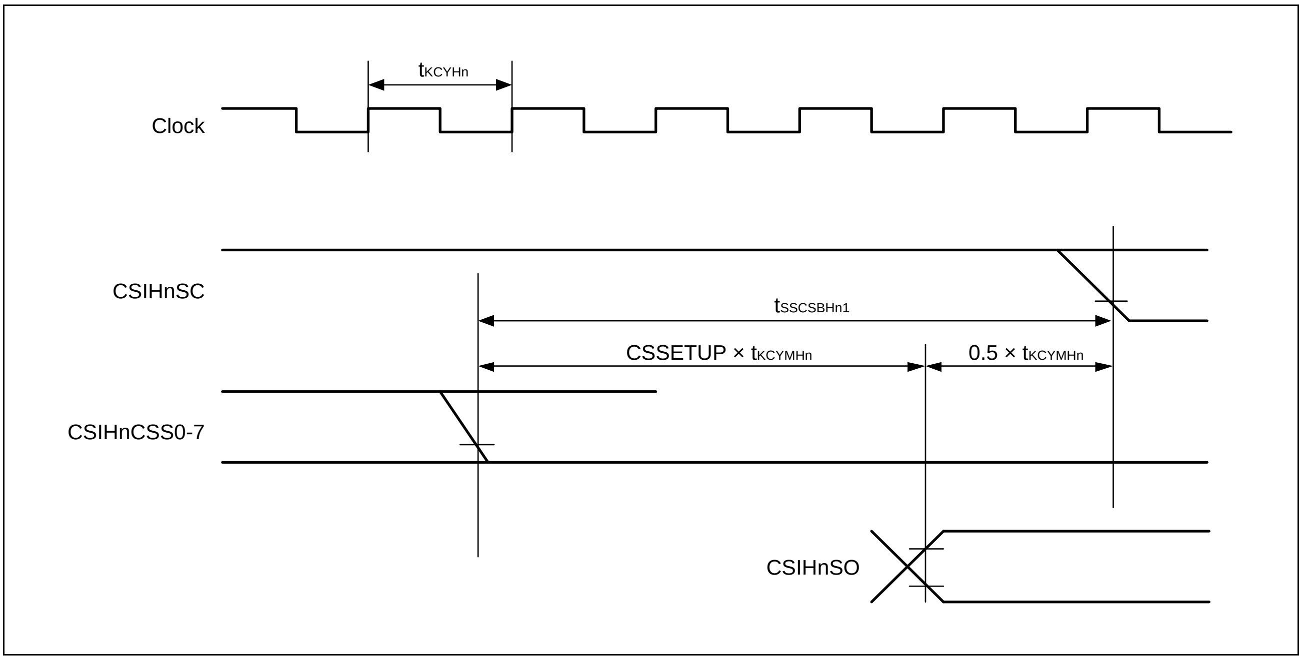
(3) CSSx

Only master mode (setup time):

- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0

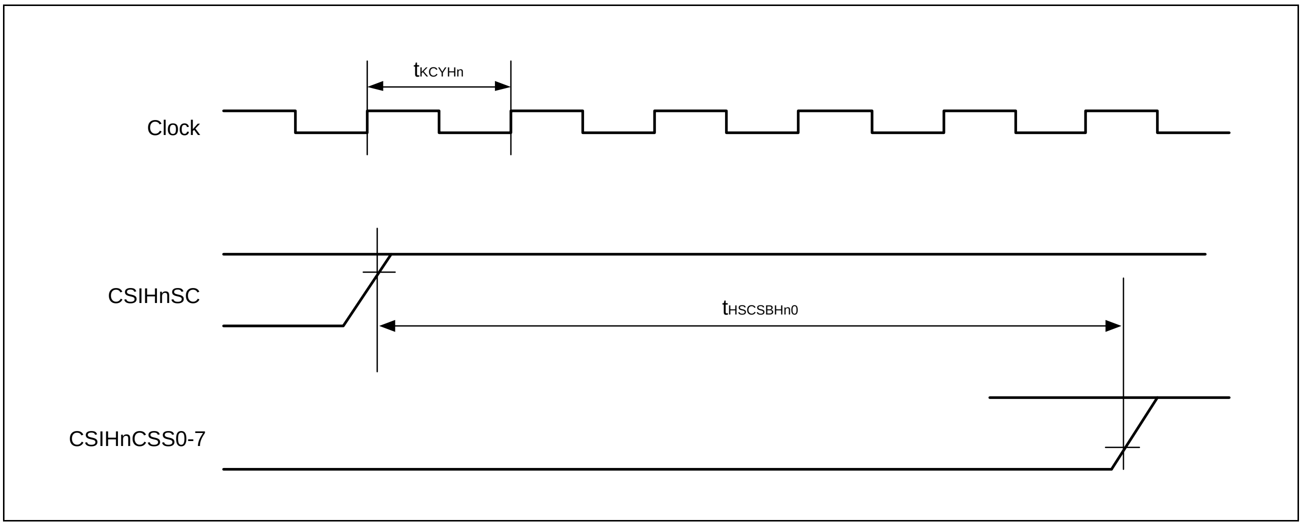


- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 1

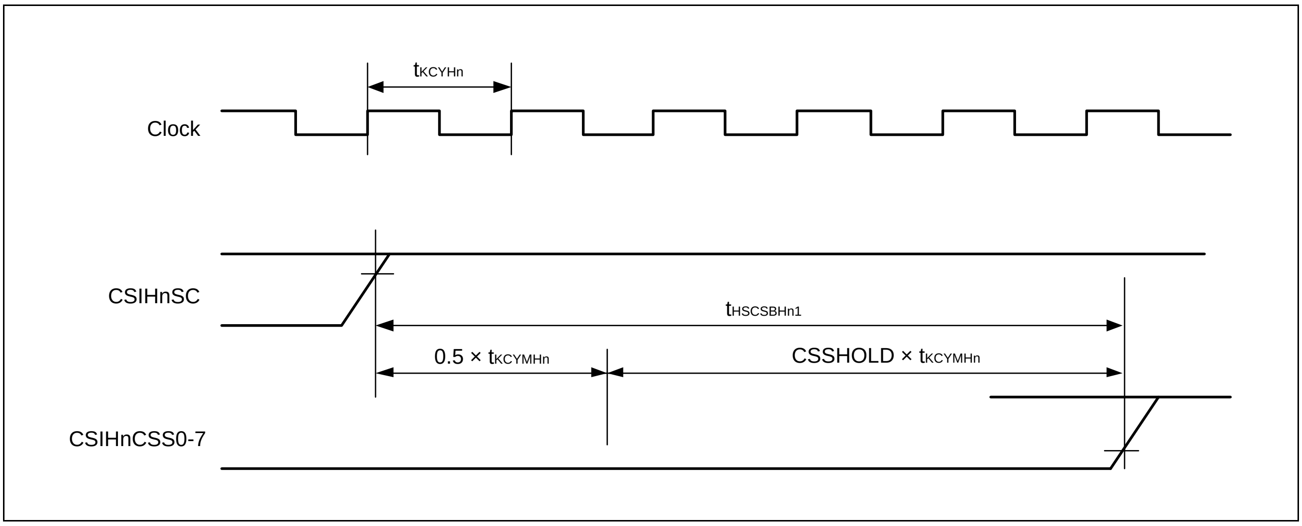


Only master mode (hold time):

- CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



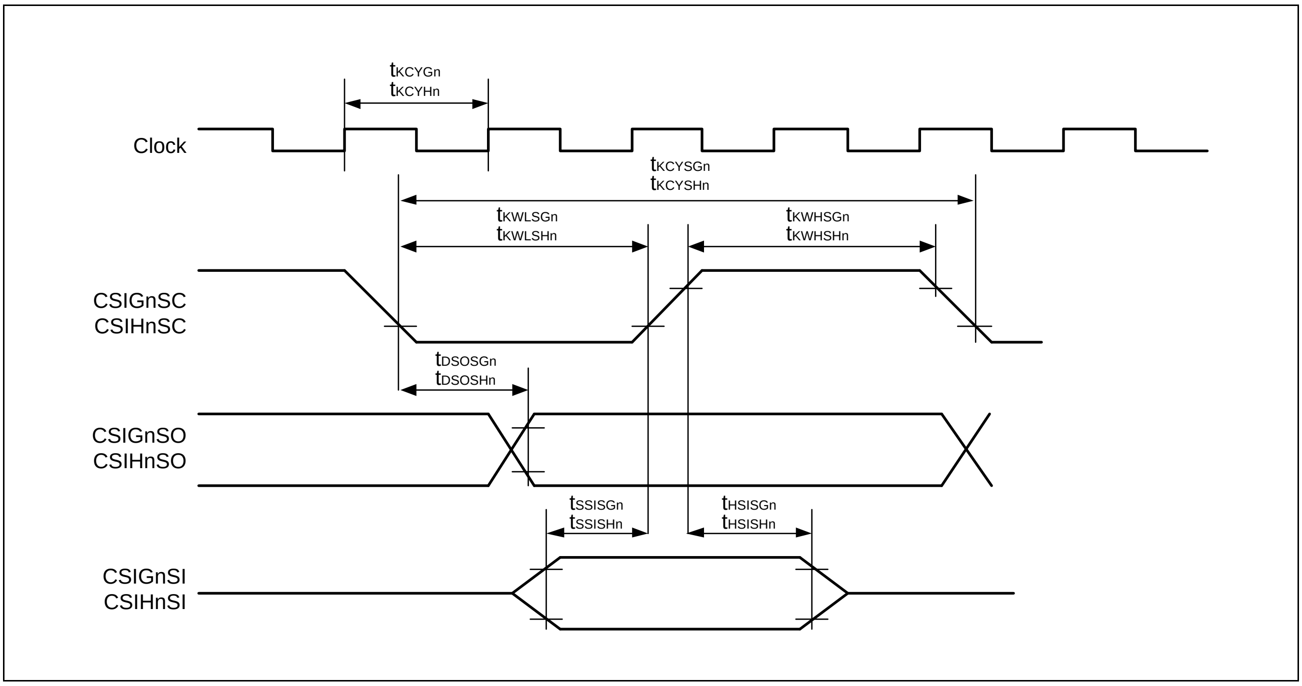
- CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



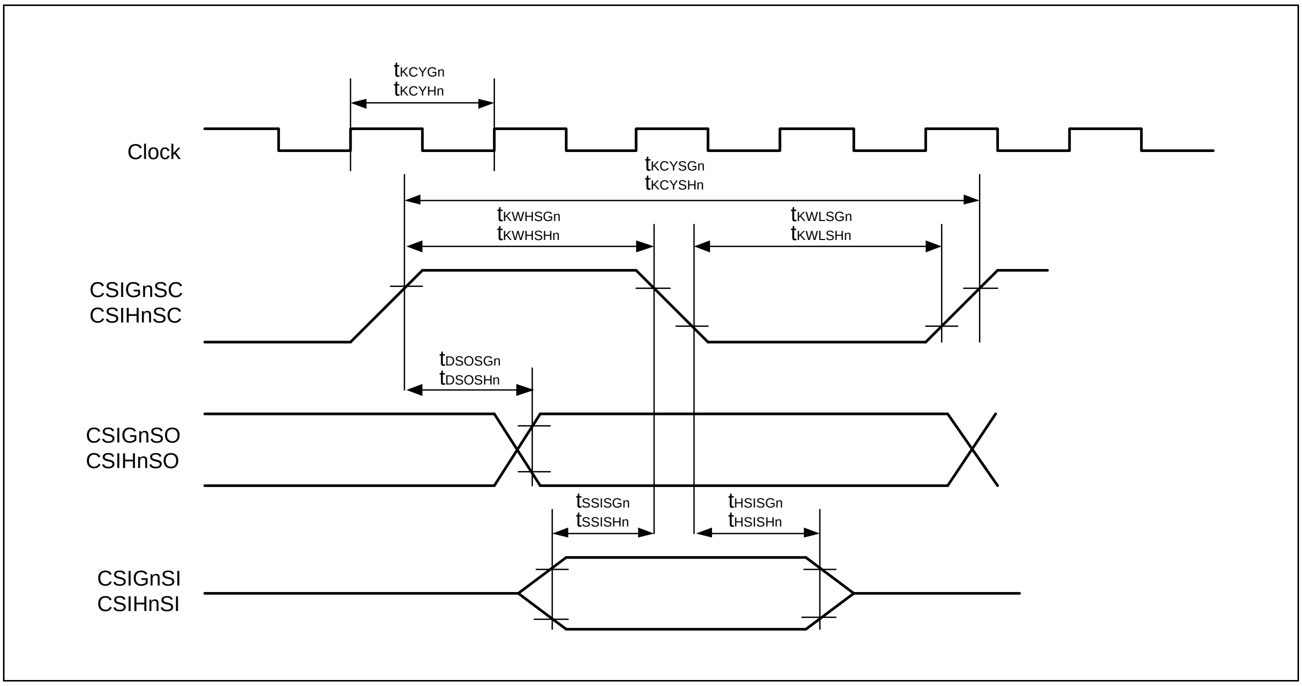
(4) SC/SI/SO

Slave mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)

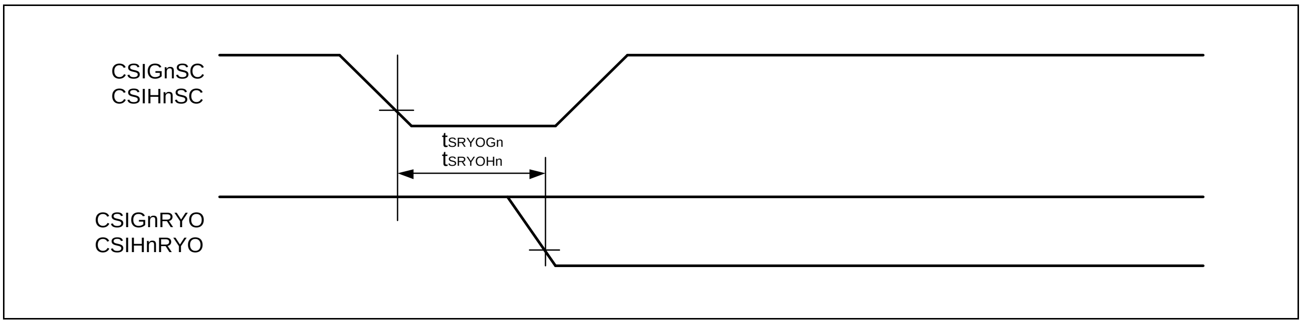


- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

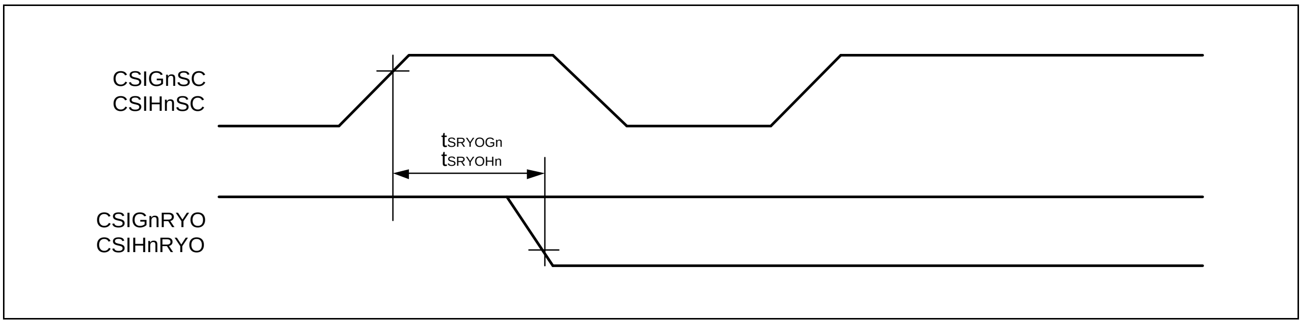


(5) RYO

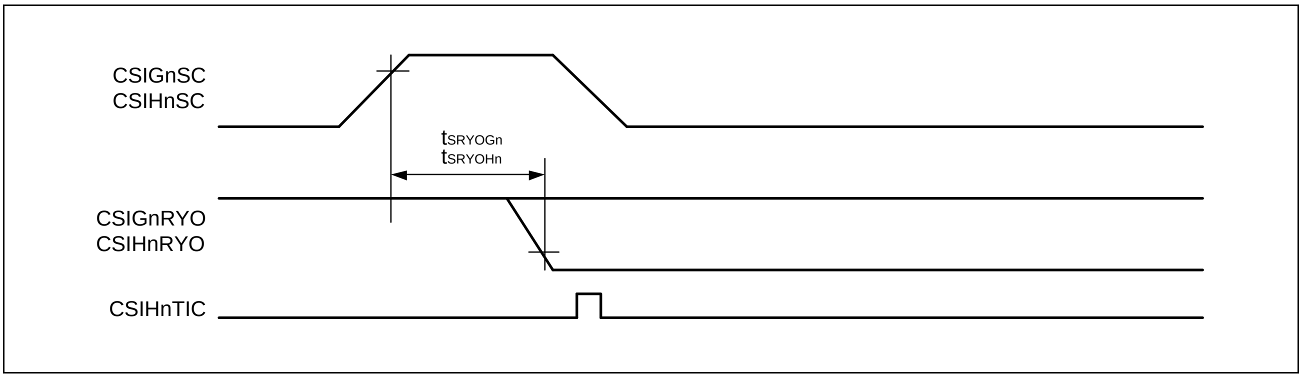
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0)



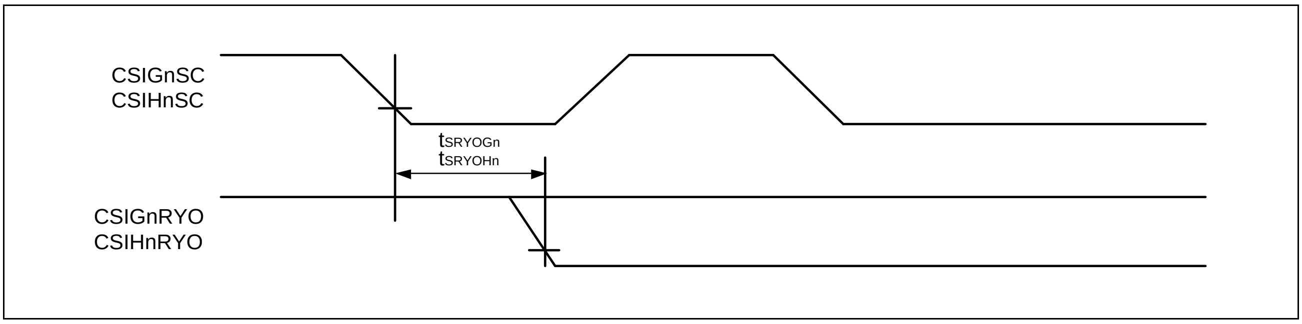
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0)



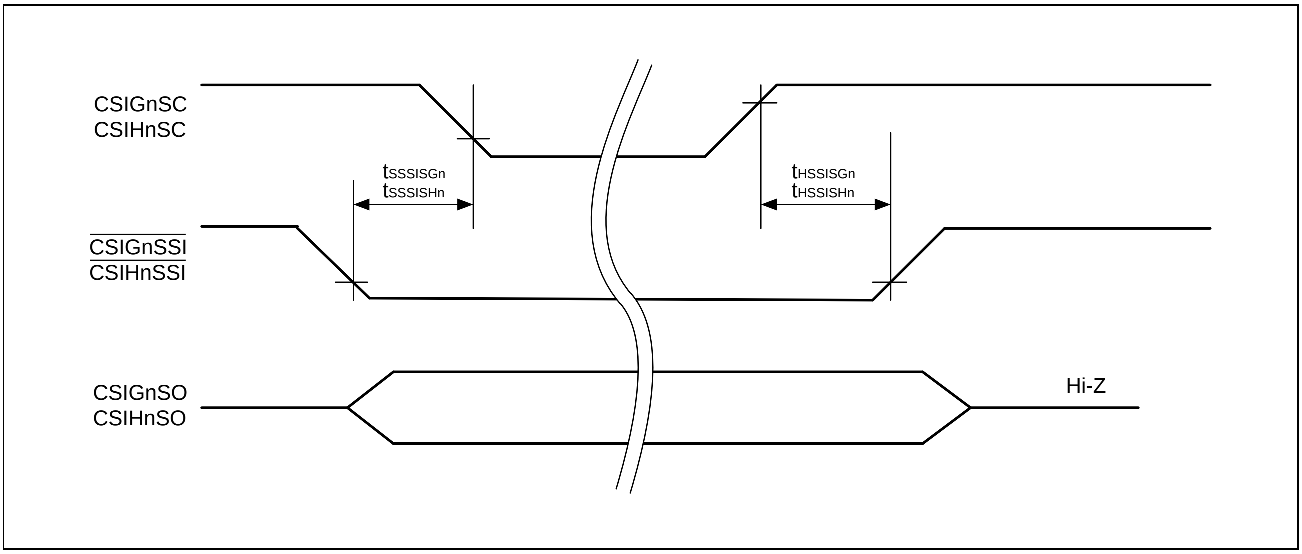
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/1)



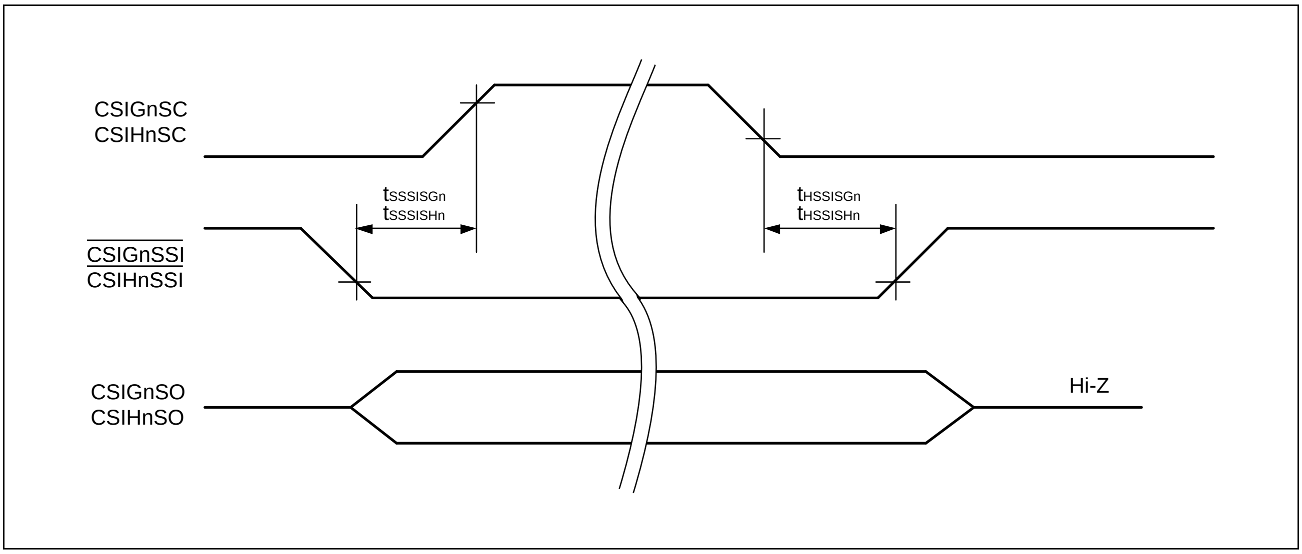
(6) SSI

Slave mode:

- CSIG (CSIGnCTL1: CSIGnSSE=1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnSSE=1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)



47B.5.11 RLIN2/RLIN3 Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RLIN3 transfer rate		LIN specification	1		20	kbps
		LIN extended baud rate	1		115.2*[1]	kbps
		UART function			1.5	Mbps
RLIN2 transfer rate		LIN specification	1		20	kbps

Note 1. The LIN extended baud rate is not part of the LIN standard specification.

47B.5.12 RIIC Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C

Table 47B.18 RIIC Timing (Normal Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				100	kHz
Bus free time (between stop/start condition)	t_{BUF}		4.7			µs
Hold time*[1]	t_{HD}: STA		4.0			µs
RIICnSCL clock low-level width	t_{LOW}		4.7			µs
RIICnSCL clock high-level time	t_{HIGH}		4.0			µs
Setup time for start/restart condition	t_{SU}: STA		4.7			µs
Data hold time	t_{HD}: DAT	CBUS compatible master	5.0			µs
		I^2C mode	0*[2]			µs
Data setup time	t_{SU}: DAT		250			ns
Stop condition setup time	t_{SU}: STO		4.0			µs
Capacitance load of each bus line	Cb				400	pF

Remark: n = 0, 1

Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Table 47B.19 RIIC Timing (Fast Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				400	kHz
Bus free time (between stop/start condition)	t_{BUF}		1.3			μs
Hold time*[1]	t_{HD}: STA		0.6			μs
RIICnSCL clock low-level width	t_{LOW}		1.3			μs
RIICnSCL clock high-level time	t_{HIGH}		0.6			μs
Setup time for start/restart condition	t_{SU}: STA		0.6			μs
Data hold time	t_{HD}: DAT	I^2C mode	0*[2]			μs
Data setup time	t_{SU}: DAT		100*[3]			ns
Stop condition setup time	t_{SU}: STO		0.6			μs
Pulse width with spike suppressed by input filter	t_{SP}		0		50	ns
Capacitance load of each bus line	Cb				400	pF

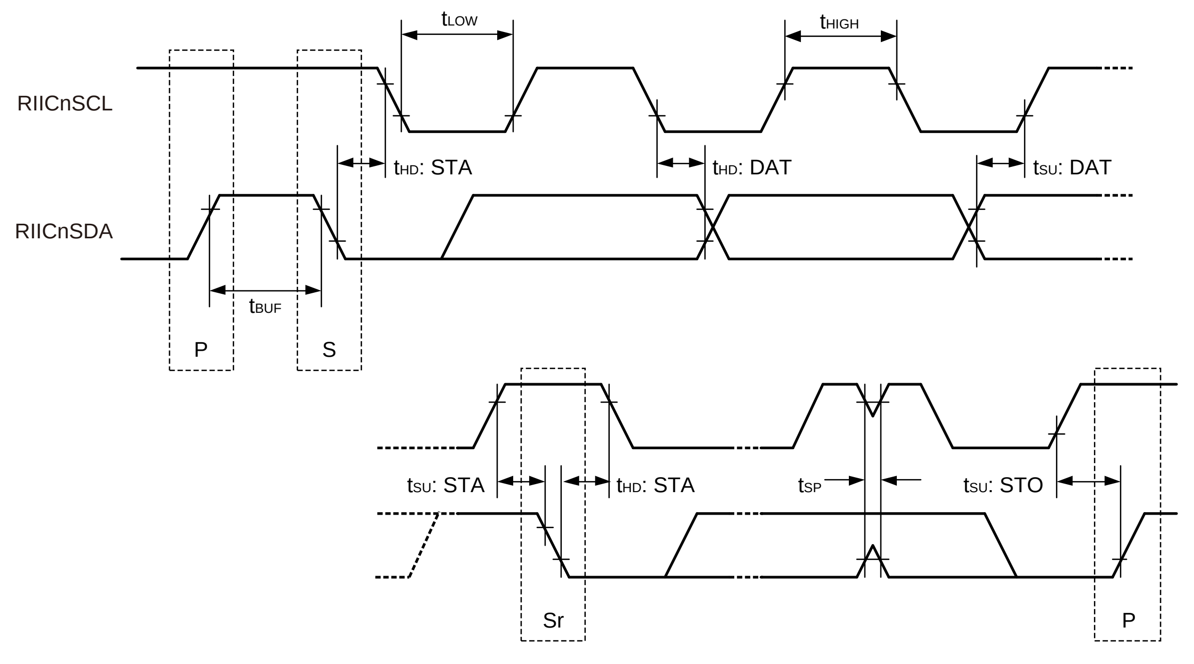
Remark: n = 0, 1

Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Note 3. The fast mode I^2C bus can be used in normal mode I^2C bus system. In this case, set the fast mode I^2C bus so that it meets the following conditions.
- If the system does not extend the RIICnSCL signal's low state hold time: t_{SU}: DAT ≥ 250 ns
- If the system extends the RIICnSCL signal's low state hold time:
Transmit the following data bit to the RIICnSDA line prior to releasing the RIICnSCL line
(1250 ns: Normal mode I^2C bus specification).



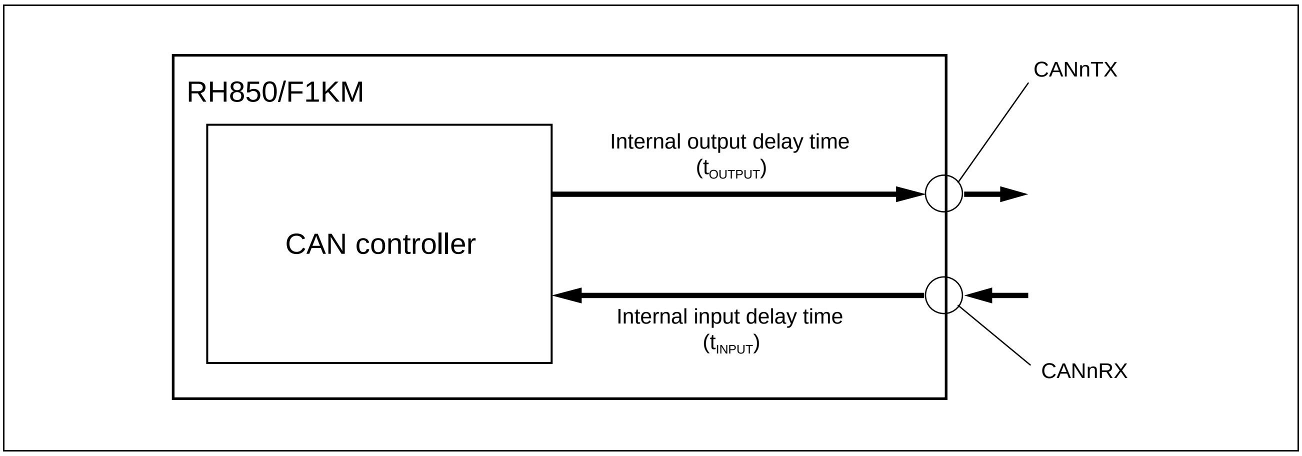
Note: P: Stop condition S: Start condition Sr: Restart condition

47B.5.13 RS-CANFD Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate		Classical CAN mode			1	Mbps
Data bit rate (CAN FD mode)		Nominal bit rate ≤ 500 kbps			5	Mbps
		Nominal bit rate > 500 kbps			2	Mbps
Internal delay time*[1]	t_{NODE}				50	ns

Note 1. t_{NODE} = Internal input delay time (t_{INPUT}) + Internal output delay time (t_{OUTPUT})



47B.5.14 FlexRay Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Products of CPU frequency 240 MHz max.

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					10	Mbps

Products of CPU frequency 160 MHz max.

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					5	Mbps

47B.5.15 Ethernet Timing

47B.5.15.1 MII Interface

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to 3.6 V, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 15 pF

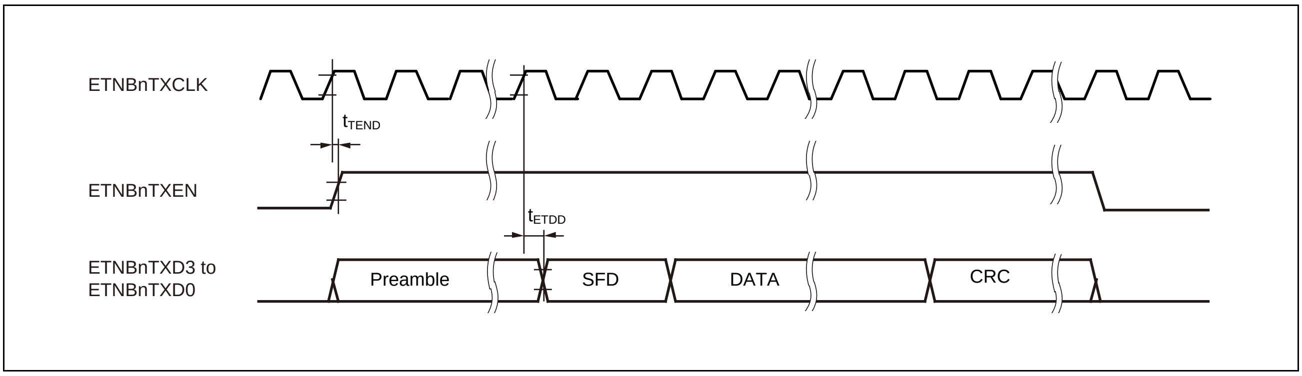
<Output driver strength>

ETNB0TXD3-0 and ETNB0TXEN pins: Fast mode

ETNB0TXCLK pin: TTL type

Table 47B.20 MII Interface (Transmission Interface)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNB0TXCLK clock period	t_{Tcyc}	100 Mbps	40 – 100 ppm	40	40 + 100 ppm	ns
		10 Mbps	400 – 100 ppm	400	400 + 100 ppm	ns
ETNB0TXEN delay vs ETNB0TXCLK ↑	t_{TEND}	CL = 15 pF			18	ns
ETNB0TXD[3:0] delay vs ETNB0TXCLK ↑	t_{ETDD}	CL = 15 pF			18	ns

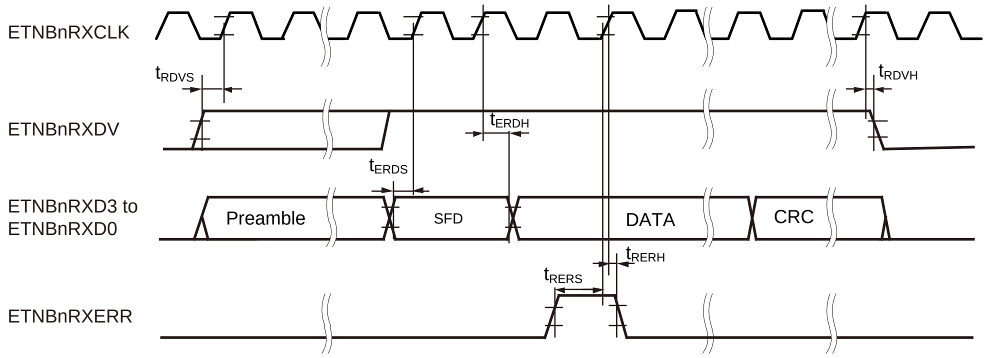


<Input buffer>

ETNB0RXCLK, ETNB0RXDV, ETNB0RXD[3:0], and ETNB0RXER pins: TTL type

Table 47B.21 MII Interface (Reception Interface)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNB0RXCLK clock period	t_{Rcyc}	100 Mbps	40 – 100 ppm	40	40 + 100 ppm	ns
		10 Mbps	400 – 100 ppm	400	400 + 100 ppm	ns
ETNB0RXDV hold time vs ETNB0RXCLK ↑	t_{RDVH}		10			ns
ETNB0RXDV setup time vs ETNB0RXCLK ↑	t_{RDVS}		10			ns
ETNB0RXD[3:0] hold time vs ETNB0RXCLK ↑	t_{ERDH}		10			ns
ETNB0RXD[3:0] setup time vs ETNB0RXCLK ↑	t_{ERDS}		10			ns
ETNB0RXERR hold time vs ETNB0RXCLK ↑	t_{RERH}		10			ns
ETNB0RXERR setup time vs ETNB0RXCLK ↑	t_{RERS}		10			ns



47B.5.15.2 Management Interface

Timing of management interface (ETNB0MDC and ETNB0MDIO) depends on software. It is necessary to adjust wait time according to AC specification of PHY.

47B.5.16 RSENT Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Tick Time			1		90	µs

47B.5.17 Timer Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

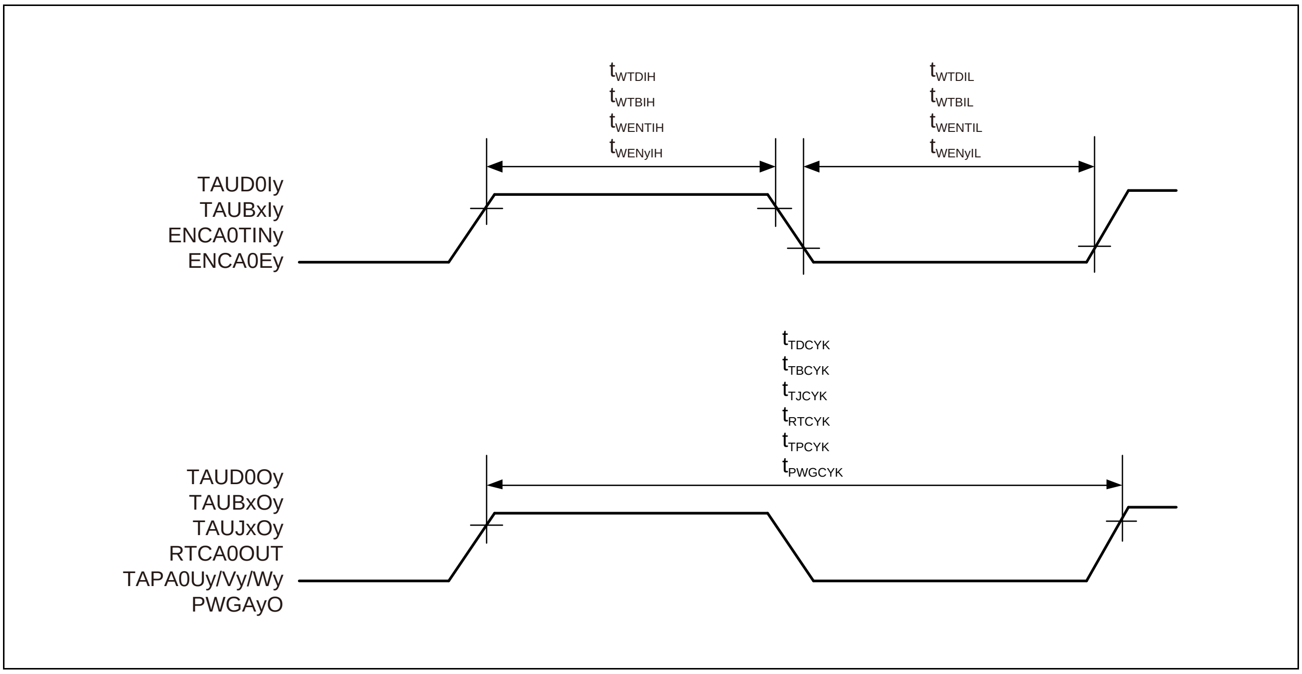
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TAUD0Iy input high/low level width (y = 0 to 15)	t_{WTDIH}/ t_{WTDIL}		n × Tsamp + 20[*1,*2]			ns
TAUD0Oy output cycle (y = 0 to 15)	t_{TDCYK}	Slow mode			10	MHz
TAUBxIy input high/low level width (x = 0, 1, y = 0 to 15)	t_{WTBIH}/ t_{WTBIL}		n × Tsamp + 20[*1,*2]			ns
TAUBxOy output cycle (x = 0, 1, y = 0 to 15)	t_{TBCYK}	Slow mode			10	MHz
TAUJxIy input high/low level width[*3] (x = 0 to 3, y = 0 to 3)	t_{WTJIH}/ t_{WTJIL}		600			ns
TAUJxIy pulse rejection[*4]	t_{WTIJRJ}		100			ns
TAUJxOy output cycle (x = 0 to 3, y = 0 to 3)	t_{TJCYK}	Slow mode			10	MHz
RTCA0OUT output cycle	t_{RTCYK}			1		Hz
TAPA0ESO input high/low level width[*3]	t_{WESIH}/ t_{WESIL}		600			ns
TAPA0ESO pulse rejection[*4]	t_{WESIRJ}		100			ns
TAPA0Uy/Vy/Wy output cycle (y = P, N)	t_{TPCYK}	Slow mode			10	MHz
ENCA0TINy input high/low level width (y = 0, 1)	t_{WENTIH}/ t_{WENTIL}		n × Tsamp + 20[*1]			ns
ENCA0Ey input high/low level width (y = 0,1,C)	t_{WENyIH}/ t_{WENyIL}		n × Tsamp + 20[*1]			ns
PWGAyO output cycle (y = 0 to 95)	t_{PWGCYK}	Slow mode			10	MHz

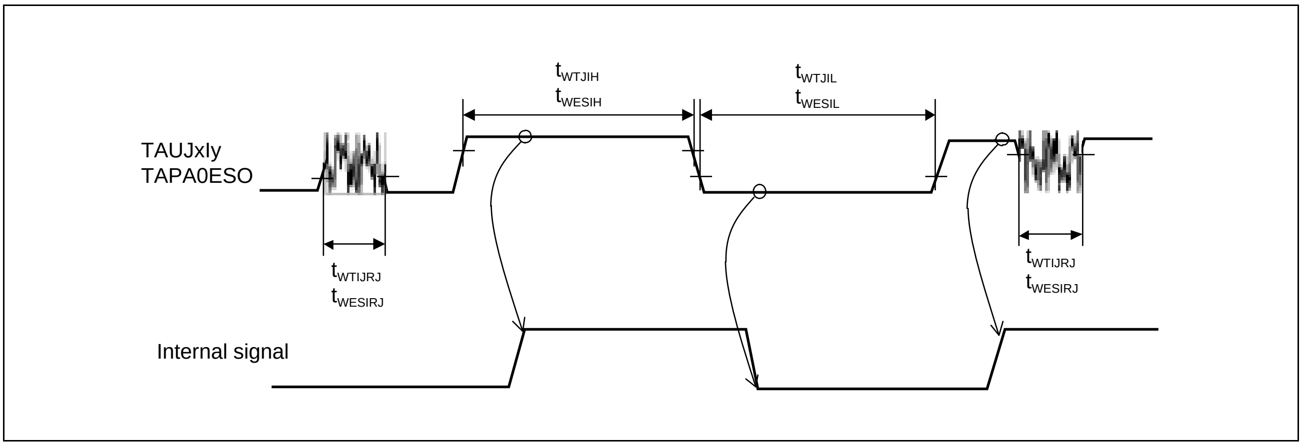
Note 1. n: Sampling number of the digital noise filter for each input.
Tsamp: Sampling time of the digital noise filter for each input.

Note 2. Input more than 1 count clock width of each timer counter channel.

Note 3. TAUJxIy and TAPA0ESO input width is needed to ensure that the internal timer input signal is activated.

Note 4. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.


t_{WTDIH}
t_{WTBIH}
t_{WENTIH}
t_{WENyIH}
t_{WTDIL}
t_{WTBIL}
t_{WENTIL}
t_{WENyIL}
TAUD0Iy
TAUBxIy
ENCA0TINy
ENCA0Ey
t_{TDCYK}
t_{TBCYK}
t_{TJCYK}
t_{RTCYK}
t_{TPCYK}
t_{PWGCYK}
TAUD0Oy
TAUBxOy
TAUJxOy
RTCA0OUT
TAPA0Uy/Vy/Wy
PWGAyO

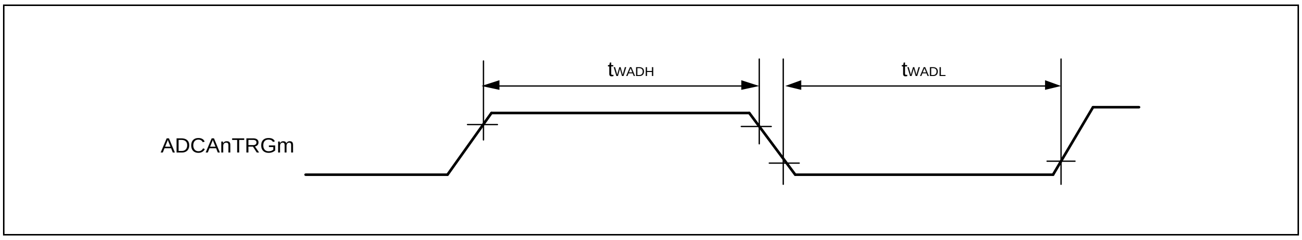

t_{WTJIH}
t_{WESIH}
t_{WTJIL}
t_{WESIL}
TAUJxIy
TAPA0ESO
t_{WTIJRJ}
t_{WESIRJ}
t_{WTIJRJ}
t_{WESIRJ}
Internal signal

47B.5.18 ADTRG Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ADCAnTRGm input high/ low level width	t_{WADH}/t_{WADL}		k × Tsamp + 20*[1]			ns

Note 1. k: Sampling number of the digital noise filter for each input.
Tsamp: Sampling time of the digital noise filter for each input.



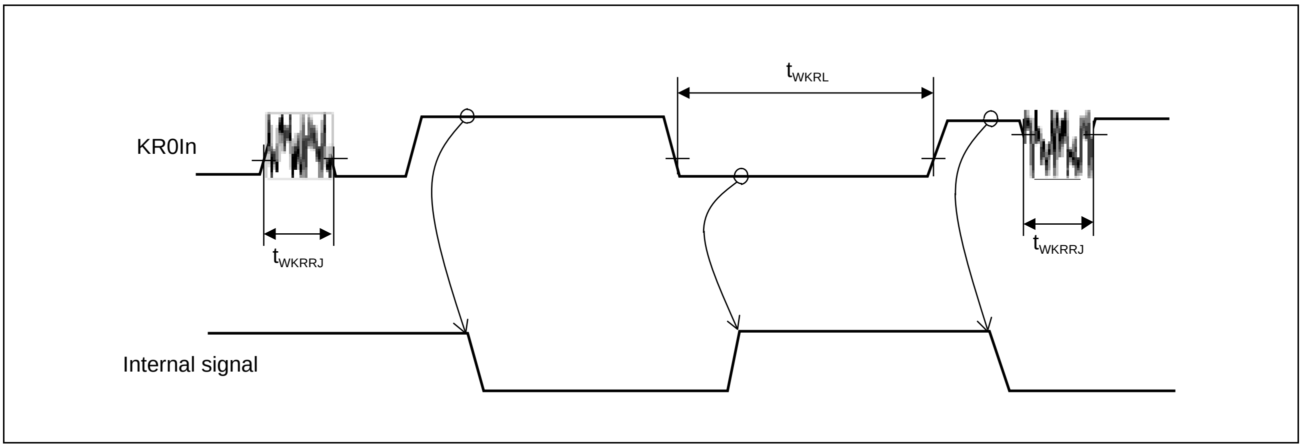
47B.5.19 Key Return Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
KR0In input low level width*[1]	t_{WKRL}		600			ns
KR0In pulse rejection*[2]	t_{WKRRJ}		100			ns

Note 1. KR0In input width is needed to ensure that the internal key input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



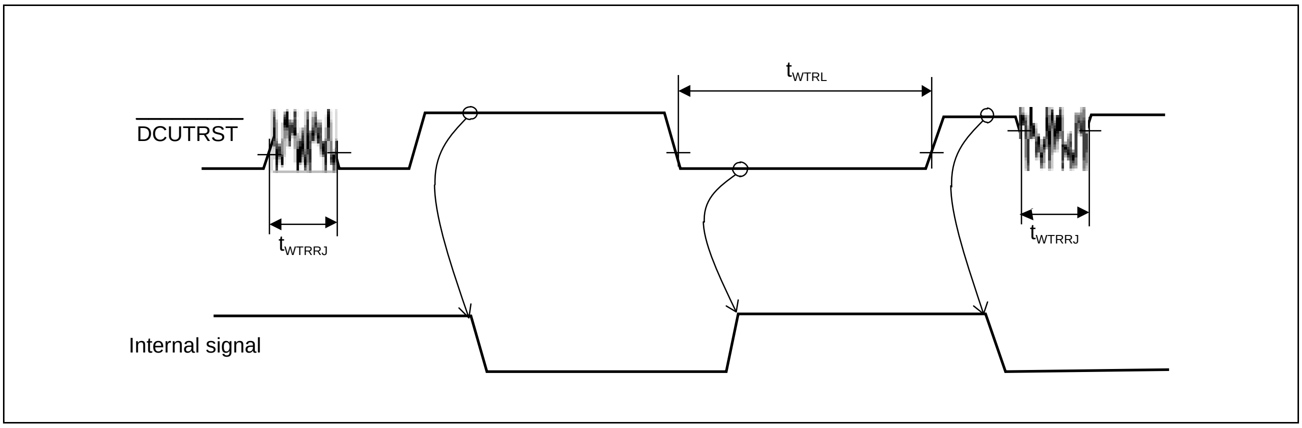
47B.5.20 $\overline{\text{DCUTRST}}$ Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{DCUTRST}}$ input low level width*[1]	t_{WTRL}		600			ns
$\overline{\text{DCUTRST}}$ pulse rejection*[2]	t_{WTRRJ}		100			ns

Note 1. $\overline{\text{DCUTRST}}$ input width is needed to ensure that the internal DCU reset input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



47B.5.21 Debug Interface Characteristics

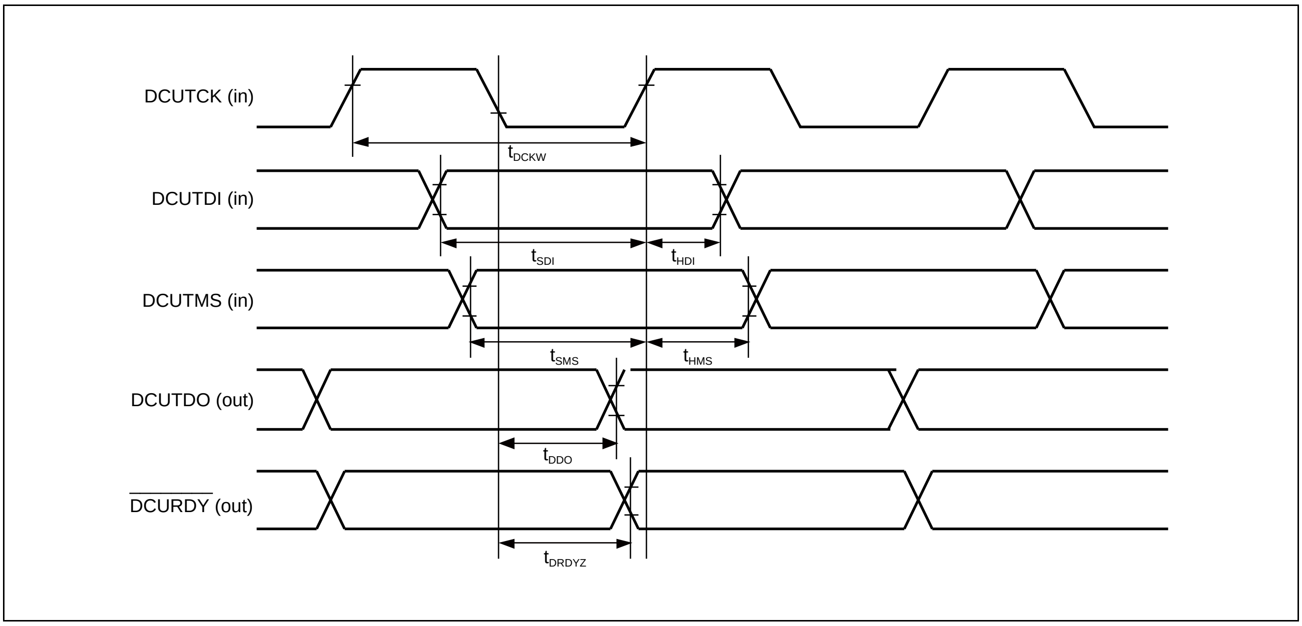
47B.5.21.1 Nexus Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

<Input buffer>
DCUTDI, DCUTCK, DCUTMS, $\overline{\text{DCUTRST}}$: TTL

<Output driver strength>
DCUTDO, $\overline{\text{DCURDY}}$: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DCUTCK cycle width	t_{DCKW}		50			ns
DCUTDI setup time (vs DCUTCK ↑)	t_{SDI}		12			ns
DCUTDI hold time (vs DCUTCK ↑)	t_{HDI}		3			ns
DCUTMS setup time (vs DCUTCK ↑)	t_{SMS}		12			ns
DCUTMS hold time (vs DCUTCK ↑)	t_{HMS}		3			ns
DCUTDO delay time (↓ DCUTCK)	t_{DDO}		0		20	ns
$\overline{\text{DCURDY}}$ delay time (↓ DCUTCK)	t_{RDYZ}		0		20	ns



47B.5.21.2 LPD (4 Pins) Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C, CL = 100 pF

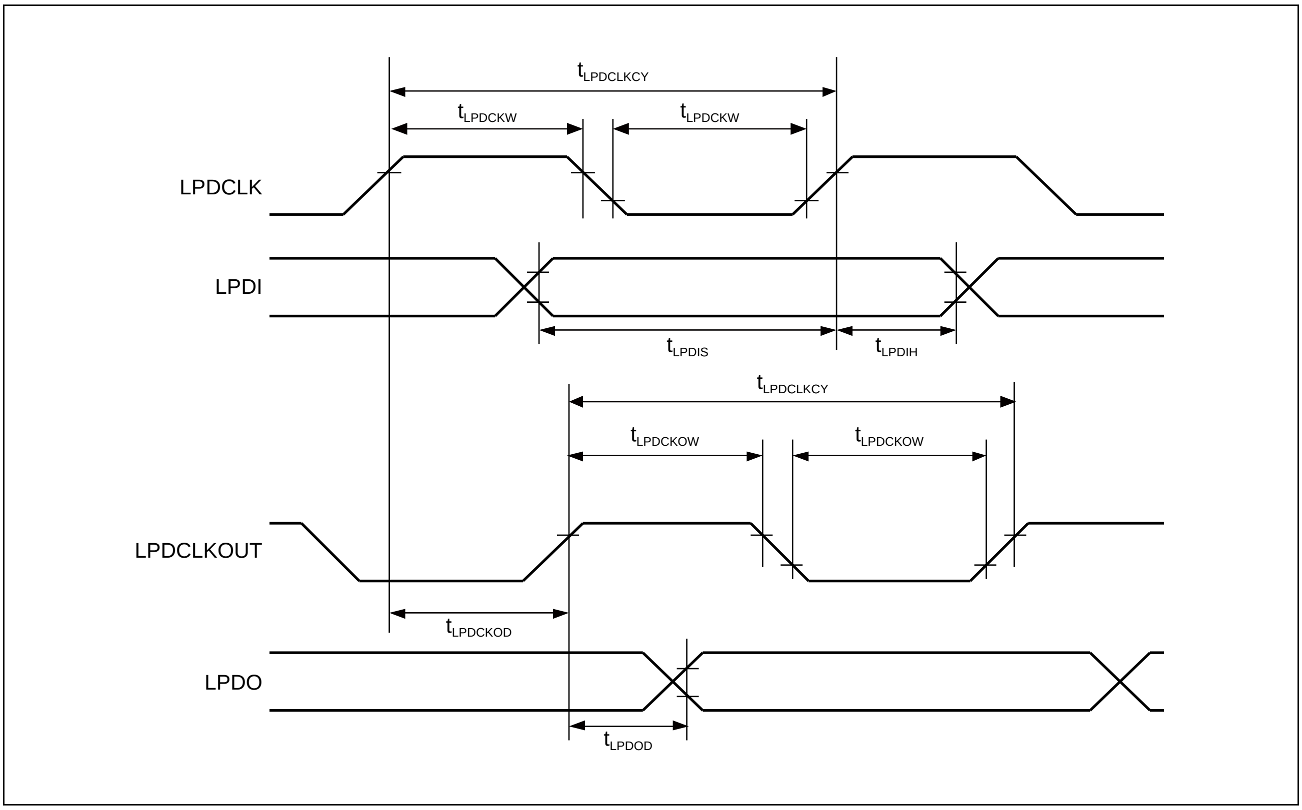
<Input buffer>

LPDCLK, LPDI: TTL

<Output driver strength>

LPDCLKOUT, LPDO: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPDCLK cycle time/ LPDCLKOUT cycle time	$t_{LPDCLKCY}$		83.3 (max.12 MHz)			ns
LPDCLK High-level width/ LPDCLK Low-level width	t_{LPDCKW}		0.5 × $t_{LPDCLKCY}$ – 10			ns
LPDCLKOUT High-level width/ LPDCLKOUT low-level width	$t_{LPDCKOW}$		t_{LPDCKW} – 10			ns
LPDI setup time (LPDCLK ↑)	t_{LPDIS}		41			ns
LPDI hold time (LPDCLK ↑)	t_{LPDIH}		3			ns
LPDCLK to LPDCLKOUT delay time	$t_{LPDCKOD}$				44	ns
LPDO delay time (LPDCLKOUT ↑)	t_{LPDOD}		0		15	ns



47B.5.21.3 LPD (1 Pin) Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 50 pF

<Input buffer>
LPDIO: TTL

<Output driver strength>
LPDIO: Fast mode

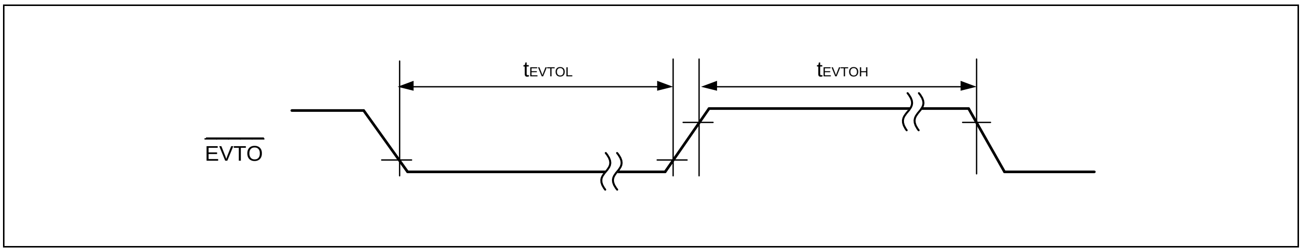
<External pull-up resistor>
LPDIO: 4.7 kΩ to 10 kΩ

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPD (1 pin) baud rate					2.0	Mbps

47B.5.21.4 Debug Event Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 50 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{EVTO}}$ high/low level width	t_{EVTOH}/t_{EVTOL}		50			ns



47B.6 A/D Converter Characteristics

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

(1/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Conversion clock	ADCLKn				8*3		40	MHz
Resolution	RESn	12-bit mode			12	12	12	bit
		10-bit mode			10	10	10	bit
Conversion time	t_{CONn}	ADCAnSMPCR.SMPT[7:0] = 12 H(40 cycle) (8M Hz*3 ≤ ADCLKn ≤ 32 MHz), External MPX is not used			1.25		5	μs
		ADCAnSMPCR.SMPT[7:0] = 18 H (46 cycle) (8 MHz*3 ≤ ADCLKn ≤ 40 MHz), External MPX is not used			1.15		5.75	μs
		ADCAnSMPCR.SMPT[7:0] = 12 H (80 cycle) (8 MHz*3 ≤ ADCLKn ≤ 32 MHz), External MPX is used			2.5*4		10	μs
		ADCAnSMPCR.SMPT[7:0] = 18 H (92 cycle) (8 MHz*3 ≤ ADCLKn ≤ 40 MHz), External MPX is used			2.3*4		11.5	μs
Sampling time	t_{SMP}	ADCAnSMPCR.SMPT[7:0] = 12 H (18 cycle) (8 MHz*3 ≤ ADCLKn ≤ 32 MHz)			0.56		2.25	μs
		ADCAnSMPCR.SMPT[7:0] = 18 H (24cycle) (8 MHz*3 ≤ ADCLKn ≤ 40 MHz)			0.6		3	μs
Overall error*1	TOEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±4.0	LSB
				ADCA0I0-5 (T&H used)			±6.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±6.0	LSB
				ADCA0I0-5 (T&H used)			±8.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Analog input voltage	VAIN0SN	ADCAnIm	T&H not used		AnVSS		AnVREF	V
		ADCA0I0-5	T&H used		0.2		A0VREF – 0.2	V
		ADCA0ImS	A0VREF ≥ EVCC		A0VSS		EVCC	V
			A0VREF < EVCC		A0VSS		A0VREF	V
		ADCA1ImS	A1VREF ≥ BVCC		A1VSS		BVCC	V
			A1VREF < BVCC		A1VSS		A1VREF	V
Operation current	IA0VREF IA1VREF	T&H not used				1.1	3.0	mA
		T&H used (max. 6 pins)					*2	mA
STOP, DeepSTOP, Cyclic STOP current (@LPS is stopped)	IA0VREFS IA1VREFS					1	10	μA
T&H current	ITH					0.5	1.3	mA/ch
T&H sampling time	t_{THSMP}				450			ns
T&H hold time	t_{THHOLD}						10	μs
Set up time of self diagnosis voltage circuit	t_{BOOT}				500			ns
Set up time of self diagnosis voltage level	t_{OUT}				500			ns
Pull-down resistor for diagnosis of open pins		ADCnIm pins	VI = AnVREF		350	500	650	kΩ
		ADCnImS pins	A0VREF ≥ EVCC: VI = EVCC A0VREF < EVCC: VI = A0VREF		100	215	800	kΩ
			A1VREF ≥ BVCC: VI = BVCC A1VREF < BVCC: VI = A1VREF					

(2/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Accuracy of self-diagnosis function (except diagnosis of open pins)	TESH0SN	12-bit mode	Self-diagnosis voltage level = AnVREF		4015-|TOEn|		4095	—
			Self-diagnosis voltage level = 2/3AnVREF		2651-|TOEn|	2731	2811+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		1968-|TOEn|	2048	2128+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		1285-|TOEn|	1365	1445+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		80+|TOEn|	—
		10-bit mode	Self-diagnosis voltage level = AnVREF		1003-|TOEn|		1023	—
			Self-diagnosis voltage level = 2/3AnVREF		663-|TOEn|	683	703+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		492-|TOEn|	512	532+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		321-|TOEn|	341	361+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		20+|TOEn|	—
Integral nonlinearity error*1	ILEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±2.0	LSB
				ADCA0I0-5 (T&H used)			±3.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Differential nonlinearity error*1	DLEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±1.0	LSB
				ADCA0I0-5 (T&H used)			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0V to 4.5V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Zero scale error (offset error)*1	ZSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Full scale error*1	FSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB

Note: Conversion accuracy when ADCA0ImS terminal is converted in 12-bit mode: Conversion accuracy can be applied if lower 2-bit is ignored from conversion result.

Note 1. This does not include quantization error.

Note 2. 3.0 + 1.3 × (the number of used T&H)

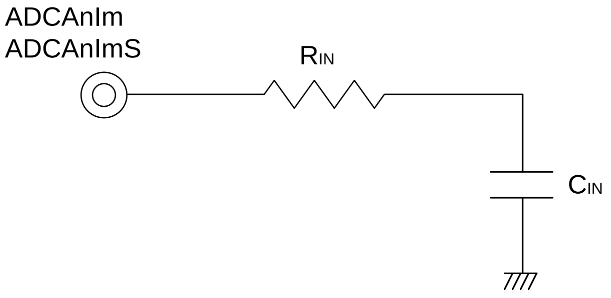
Note 3. Include the oscillation accuracy of HS IntOSC.

Note 4. When the external multiplexer is used, the detailed time of A/D conversion is MPX setup time, sampling time and successive approximation time. MPX setup time is same as “sampling time + successive approximation time”.

CAUTION

When an external digital pulse is applied to AP0, AP1, P8, P9, P18, and P19 pins during an A/D conversion this may lead to an A/D conversion result with a larger conversion error as expected due to the coupling noise of the external digital pulse.
The same behavior may apply when the digital buffer is used as an output pin. For the output port the potential degradation increases with the driven total output current of the port. In addition the conversion resolution may drop if the output current fluctuates at adjacent pins due to the coupling effect of the external circuit connected to these port pins.

47B.6.1 Equivalent Circuit of the Analog Input Block



Terminals	Condition	RIN (kΩ)	CIN (pF)
ADCA0I0 to 5	When T&H is used	14.1	2.2
	When T&H is not used	3.8	2.1
ADCA0I6 to 15	—	3.8	2.1
ADCA0I0S to 3S, 5S to 11S, 14S to 16S	—	5.3	9.3
ADCA0I4S, 17S to 19S	—	7.2	9.3
ADCA1I0 to 15	—	3.8	2
ADCA1I0S to 19S	—	5.2	7.4

CAUTION

This specification is not tested during outgoing inspection. Therefore RIN and CIN are reference values only and not guaranteed. In addition these values are specified as maximum values.

47B.7 Flash Programming Characteristics

47B.7.1 Code Flash

The code flash memory is shipped in the erased state. If the code flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47B.22 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*[3]		5*[4]		30	MHz
Number of rewrites*[1]	CWRT	Data retention of 20 years*[2]	1000			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 1000), the device can be erased “n” times for each block. For example, when a block of 32 KB is erased after 256 bytes of writing have been performed for different addresses 128 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the code flash memory.

Note 3. f_{PCLK} = 1/8 f_{CPUCLK_H}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47B.23 Programming Characteristic

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		f_{PCLK} ≥ 20 MHz CWRT < 100 times	256 B		0.4*[1]	6*[1]	ms
			8 KB		20	90	ms
			32 KB		80	360	ms
			256 KB		0.6	2.7	s
			384 KB		0.9	4.1	s
			512 KB		1.2	5.4	s
			768 KB		1.7	8.1	s
			1 MB		2.3	10.8	s
			1.5 MB		3.4	16.2	s
			2 MB		4.5	21.5	s
			3 MB		6.8	32.3	s
			4 MB		9	43	s
		f_{PCLK} ≥ 20 MHz CWRT ≥ 100 times	256 B		0.5*[1]	7.2*[1]	ms
			8 KB		24	108	ms
			32 KB		96	432	ms
			256 KB		0.7	3.3	s
			384 KB		1.1	4.9	s
			512 KB		1.4	6.5	s
			768 KB		2.1	9.8	s
			1 MB		2.7	13	s
			1.5 MB		4.1	19.5	s
			2 MB		5.4	26	s
			3 MB		8.1	39	s
			4 MB		10.8	52	s

Table 47B.23 Programming Characteristic

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Erase time		$f_{PCLK} \geq 20$ MHz CWRT < 100 times	8 KB		39	120	ms
			32 KB		141	480	ms
			256 KB		1.2	3.5	s
			384 KB		1.7	5.3	s
			512 KB		2.3	7	s
			768 KB		3.4	10.5	s
			1 MB		4.5	14	s
			1.5 MB		6.8	21	s
			2 MB		9	28	s
			3 MB		13.5	42	s
			4 MB		18	56	s
		$f_{PCLK} \geq 20$ MHz CWRT ≥ 100 times	8 KB		47	144	ms
			32 KB		169	576	ms
			256 KB		1.4	4.2	s
			384 KB		2.1	6.3	s
			512 KB		2.7	8.4	s
			768 KB		4.1	12.6	s
			1 MB		5.4	16.8	s
			1.5 MB		8.1	25.2	s
			2 MB		10.8	33.6	s
			3 MB		16.2	50.4	s
			4 MB		21.6	67.2	s

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

47B.7.2 Data Flash

The data flash memory is shipped in the erased state. If the data flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REGVCC = EVCC = VPOC to 5.5 V, BVCC = VPOC to REGVCC, A0VREF = 3.0 V to 5.5 V,
A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C,
CL = 30 pF

Table 47B.24 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*[3]		5*[4]		30	MHz
Number of rewrites*[1]	CWRT	Data retention 20 years*[2]	125 k			times
		Data retention 3 years*[2]	250 k			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 125000), the device can be erased “n” times for each block. For example, when a block of 64 bytes is erased after 4 bytes of writing have been performed for different addresses 168 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the data flash memory.

Note 3. f_{PCLK} = 1/8 f_{CPUCLK_H}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47B.25 Programming Characteristics

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		$f_{PCLK} \geq 20$ MHz	4 B		0.16*[1]	1.7*[1]	ms
			32 KB		1.4	6.8	s
			64 KB		2.79	13.44	s
			128 KB		5.58	26.88	s
Erase time		$f_{PCLK} \geq 20$ MHz	64 B		1.7*[1]	10*[1]	ms
			32 KB		0.9	5.2	s
			64 KB		1.74	10.24	s
			128 KB		3.48	20.48	s
Blank check time		$f_{PCLK} \geq 20$ MHz	4 B			30*[1]	µs
			64 B			100*[1]	µs
			32 KB			35.2	ms
			64 KB			70.4	ms
			128 KB			140.8	ms

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

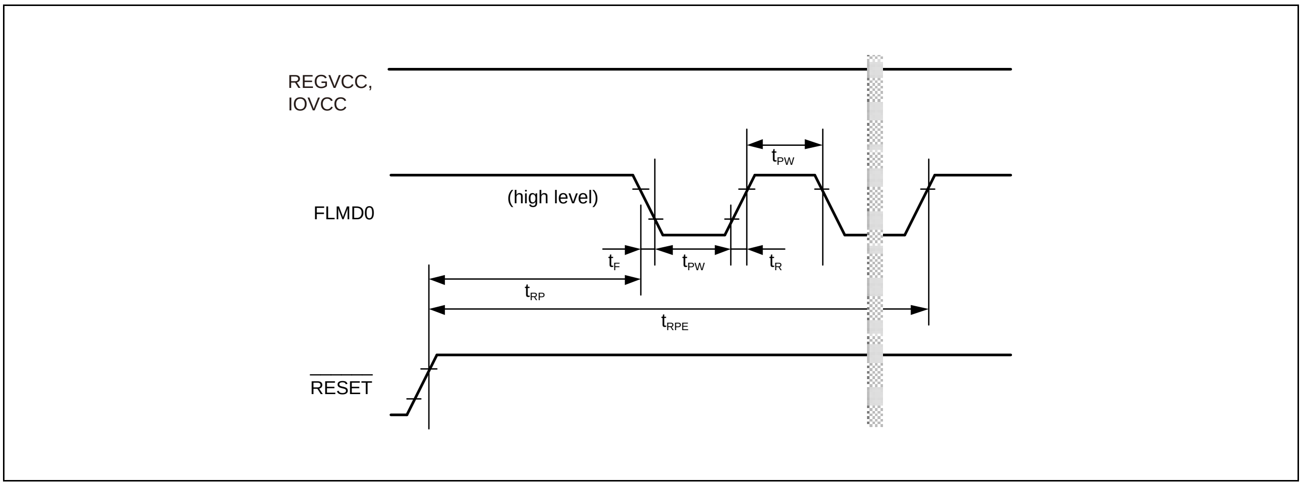
47B.7.3 Serial Programming Interface

47B.7.3.1 Serial Programmer Setup Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0 pulse input start time	t_{RP}		1.5			ms
FLMD0 pulse input end time	t_{RPE}				101.5	ms
FLMD0 low/high level width	t_{PW}		3.2			µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns

Note: IOVCC: EVCC = BVCC = A0VREF = A1VREF



47B.7.3.2 Flash Programming Interface

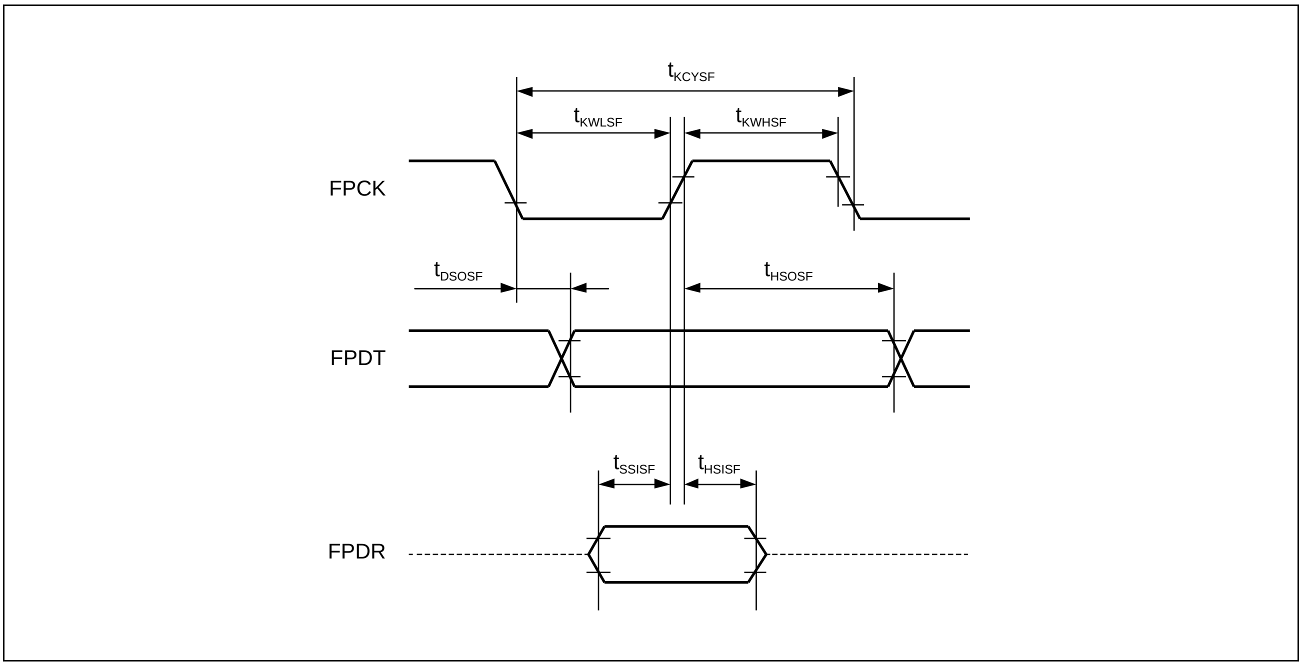
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, BVCC = 3.0 V to REGVCC, A0VREF = 3.0 V to 5.5 V, A1VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = BVSS = A0VSS = A1VSS = 0 V, CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%, Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Flash Programming transfer rate		1-wired UART mode			1	Mbps
		2-wired UART mode (Products of CPU frequency 240 MHz max.)			1.5	Mbps
		2-wired UART mode (Products of CPU frequency 160 MHz max.)			1	Mbps
FPCK cycle time	t_{KCYSF}	3-wired clock sync mode	200*[1]			ns
FPCK high level width	t_{KWHSF}	3-wired clock sync mode	t_{KCYSF} / 2 – 15			ns
FPCK low level width	t_{KWLSF}	3-wired clock sync mode	t_{KCYSF} / 2 – 15			ns
FPDR setup time (vs. FPCK)	t_{SSISF}	3-wired clock sync mode	t_{Pcyc} × 2			ns
FPDR hold time (vs. FPCK)	t_{HSISF}	3-wired clock sync mode	t_{Pcyc} × 2			ns
FPDT output delay (vs. FPCK)	t_{DSOSF}	3-wired clock sync mode Not continuous transfer (data: 1st bit)			0	ns
		3-wired clock sync mode Not continuous transfer (data: except 1st bit)			$-t_{KWHSF} + 3 \times t_{Pcyc} + 36$	ns
FPDT hold time (vs. FPCK)	t_{HSOSF}	3-wired clock sync mode	t_{Pcyc} × 2			ns

Note 1. Input an external clock that is more than 6 clocks of PCLK.

NOTE

t_{Pcyc} is period of PCLK.



47B.8 Thermal Characteristics

47B.8.1 Parameters

Package	Item	Symbol	Estimate	Unit	Note
272-pin FPBGA	Thermal Resistance	Θja	21.1	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	11.8		
233-pin FPBGA	Thermal Resistance	Θja	21.2	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	11.8		
176-pin LQFP	Thermal Resistance	Θja	35.5	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	27.6		
144-pin LQFP	Thermal Resistance	Θja	35.5	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	26.9		
100-pin LQFP	Thermal Resistance	Θja	38.3	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	28.1		

NOTE

The thermal resistance and characterization parameters depend on the usage environment.

47B.8.2 Board

Conforming to JESD51-7 (4 layers)

	Board Size (mm)		Area (mm^2)
	X	Y	
Board	76.2	114.3	8709.66
Remaining copper rates	Thickness of conductors		
50-95-95-50%	70-35-35-70 µm		

Section 47C Electrical Characteristics of RH850/F1KM-S1

47C.1 Overview

The electrical spec of this device is guaranteed by the following operational condition. But, this condition is different depends on each characteristics, so refer to each chapter for more detail.

47C.1.1 Pin Groups

47C.1.1.1 100-Pin Version

Symbol	Pin Group Supplied by	Related Pins / Ports
PgR	REGVCC, AWOVSS	X1, X2
PgE	EVCC, EVSS	Related ports: JP0, P0, P8, P9, P10, P11
		Related pins: $\overline{\text{RESET}}$, FLMD0
PgA0	A0VREF, A0VSS	Related port: AP0

47C.1.1.2 80-Pin Version

Symbol	Pin Group Supplied by	Related Pins / Ports
PgR	REGVCC, AWOVSS	X1, X2
PgE	EVCC, EVSS	Related ports: JP0, P0, P8, P9, P10, P11
		Related pins: $\overline{\text{RESET}}$, FLMD0
PgA0	A0VREF, A0VSS	Related port: AP0

47C.1.1.3 64-Pin Version

Symbol	Pin Group Supplied by	Related Pins / Ports
PgR	REGVCC, AWOVSS	X1, X2
PgE	EVCC, EVSS	Related ports: JP0, P0, P8, P9, P10
		Related pins: $\overline{\text{RESET}}$, FLMD0
PgA0	A0VREF, A0VSS	Related port: AP0

47C.1.1.4 48-Pin Version

Symbol	Pin Group Supplied by	Related Pins / Ports
PgR	REGVCC, AWOVSS	X1, X2
PgE	EVCC, EVSS	Related ports: JP0, P0, P8, P9, P10
		Related pins: $\overline{\text{RESET}}$, FLMD0
PgA0	A0VREF, A0VSS	Related port: AP0

47C.1.2 General Measurement Conditions

47C.1.2.1 Common Conditions

- Power supply
  - REGVCC = EVCC = VPOC*[1] to 5.5 V
  - A0VREF = 3.0 V to 5.5 V
  - AWOVSS = ISOVSS = EVSS = A0VSS = 0 V
- Capacitance of the internal regulator
  - CAWOVCL: 0.1 µF ±30%
  - CISOVCL: 0.1 µF ±30%
- Operating temperature
  - Tj = –40 to +130°C @ R7F7016xx3AFP*[2]
  - Tj = –40 to +150°C @ R7F7016xx4AFP*[2]

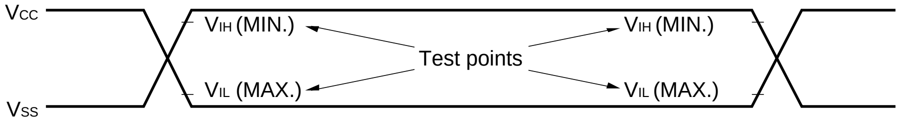
  xx = 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95
- Load conditions
  - CL = 30 pF

Note 1. “VPOC” means POC (power-on clear) detection voltage. For more detail, see **Section 47C.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

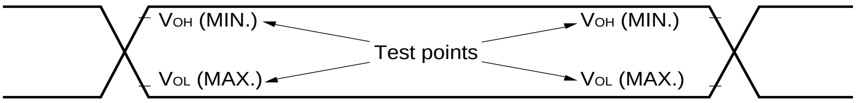
Note 2. Regarding operation temperature of each product, see **Section 1C.3, RH850/F1KM Product Lineup**.

47C.1.2.2 AC Characteristic Measurement Condition

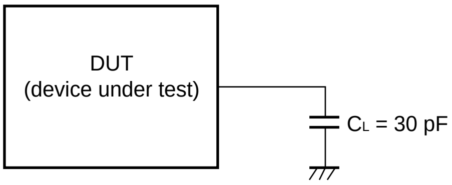
(1) AC Test Input Measurement Points



(2) AC Test Output Measurement Points



(3) Load Conditions



CAUTION

If the load capacitance exceeds 30 pF due to the circuit configuration, it is recommended to insert a buffer in order to reduce capacitance to less than 30 pF.

47C.2 Absolute Maximum Ratings

CAUTIONS

1. Do not directly connect outputs (or input/outputs) to each other, power supply and ground.
2. Even momentarily exceeding the absolute maximum rating for just one item creates a threat of failure in the reliability of the products. That is, the absolute maximum ratings are the levels that raise a threat of physical damage to the products. Be sure to use the products only under conditions that do not exceed the ratings. The quality and normal operation of the product are guaranteed under the standards and conditions given as DC and AC characteristics.
3. When designing an external circuit ensure that the connections don't conflict with the port state of this device.

47C.2.1 Supply Voltages

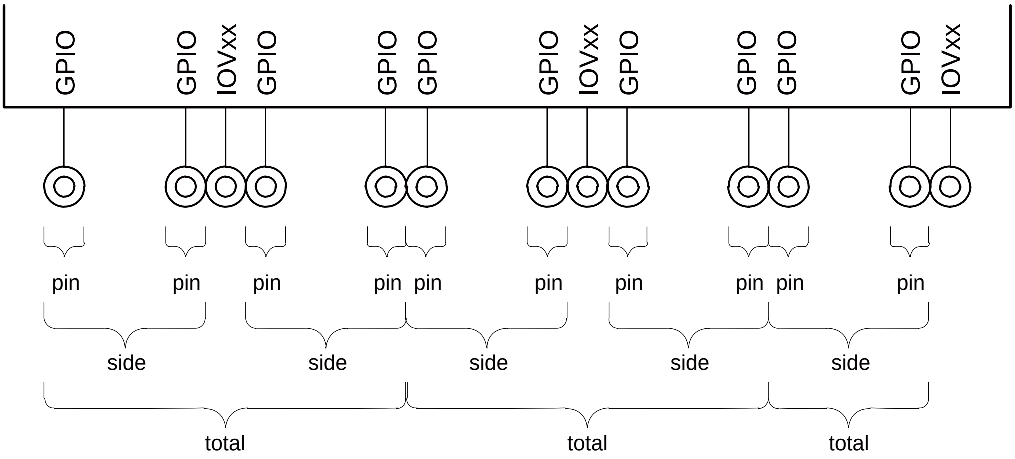
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
System supply voltage	REGVCC		–0.5		6.5	V
	AWOVSS		–0.5		0.5	V
	ISOVSS		–0.5		0.5	V
Port supply voltage	EVCC		–0.5		6.5	V
	EVSS		–0.5		0.5	V
A/D-converter supply voltage	A0VREF		–0.5		6.5	V
	A0VSS		–0.5		0.5	V

47C.2.2 Port Voltages

Item	Pin Group*[1]	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	PgR	VI		–0.5		REGVCC + 0.5 (Do not exceed 6.5 V)	V
	PgE			–0.5		EVCC + 0.5 (Do not exceed 6.5 V)	V
	PgA0			–0.5		A0VREF + 0.5 (Do not exceed 6.5 V)	V

Note 1. The characteristics of the alternative-function pins are the same as those of the port pins unless otherwise specified.

47C.2.3 Port Current



Definition of the condition:

- Per pin: Output current of one GPIO
- Per side: Total output current of all GPIO pins on one side of one IOVxx
- Total: Total output current of both sides of one IOVxx

Note:

- GPIO: General-purpose I/O pin (JP0, P0, P8, P9, P10, P11, AP0)
- IOVxx: Power supply pin for I/O pins (EVCC/EVSS, A0VREF/A0VSS)

47C.2.3.1 100-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (Total of P9_0 to P9_6)			–48	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5, P10_15, P11_0)			–48	mA
			Per side (Total of JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P8_2, P8_10 to P8_12)			–48	mA
			Per side (Total of JP0_0 to JP0_2)			–30	mA
			Per side (Total of P0_7 to P0_10, P8_0, P8_1, P8_3 to P8_9)			–48	mA
			Per side (Total of P10_6 to P10_9)			–40	mA
			Per side(Total of P10_10 to P10_14, P11_1 to P11_7)			–48	mA
			Per side (Total of P10_0 to P10_2)			–30	mA
			Total (EVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (Total of P9_0 to P9_6)			48	mA
			Per side (Total of P0_0 to P0_6, P0_11 to P0_14, P10_3 to P10_5, P10_15, P11_0)			48	mA
			Per side (Total of JP0_0 to JP0_5, P8_2, P8_10 to P8_12)			48	mA
			Per side (Total of P0_7 to P0_10)			40	mA
			Per side (Total of P8_0, P8_1, P8_3 to P8_9)			48	mA
			Per side(Total of P10_6 to P10_14, P11_1, P11_2)			48	mA
			Per side (Total of P11_3 to P11_7)			48	mA
			Per side (Total of P10_0 to P10_2)			30	mA
			Total (EVCC)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA

47C.2.3.2 80-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (Total of P9_0 to P9_6, P10_6 to P10_14, P11_1 to P11_4)			–48	mA
			Per side (Total of P10_0 to P10_2)			–30	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5, P10_15, P11_0)			–48	mA
			Per side (Total of JP0_0 to JP0_5, P0_4 to P0_12, P8_0 to P8_6)			–48	mA
			Total (EVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (Total of P9_0 to P9_6, P10_6 to P10_14, P11_1 to P11_4)			48	mA
			Per side (Total of P10_0 to P10_2)			30	mA
			Per side (Total of P0_0 to P0_6, P0_11, P0_12, P10_3 to P10_5, P10_15, P11_0)			48	mA
			Per side (Total of JP0_0 to JP0_5, P0_7 to P0_10, P8_0 to P8_6)			48	mA
			Total (EVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA

47C.2.3.3 64-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (Total of P9_0 to P9_3, P10_6 to P10_14)			–48	mA
			Per side (Total of P10_0 to P10_2)			–30	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5)			–48	mA
			Per side (Total of JP0_0 to JP0_5, P0_4 to P0_6, P8_0 to P8_6)			–48	mA
			Total (EVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (Total of P9_0 to P9_3, P10_6 to P10_14)			48	mA
			Per side (Total of P10_0 to P10_2)			30	mA
			Per side (Total of P0_0 to P0_6, P10_3 to P10_5)			48	mA
			Per side (Total of JP0_0 to JP0_5, P8_0 to P8_6)			48	mA
			Total (EVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA

47C.2.3.4 48-Pin Version

Item	Symbol	Pin Group	Condition	MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per pin			–10	mA
			Per side (Total of P9_0, P9_1, P10_6 to P10_10)			–48	mA
			Per side (Total of P10_0 to P10_2)			–30	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5)			–48	mA
			Per side (Total of JP0_0 to JP0_5, P8_0, P8_1)			–48	mA
			Total (EVCC)			–60	mA
		PgA0	Per pin			–10	mA
			Total (A0VREF)			–48	mA
Low-level output current	IOL	PgE	Per pin			10	mA
			Per side (Total of P9_0, P9_1, P10_6 to P10_10)			48	mA
			Per side (Total of P10_0 to P10_2)			30	mA
			Per side (Total of P0_0 to P0_3, P10_3 to P10_5)			48	mA
			Per side (Total of JP0_0 to JP0_5, P8_0, P8_1)			48	mA
			Total (EVSS)			60	mA
		PgA0	Per pin			10	mA
			Total (A0VSS)			48	mA

47C.2.4 Temperature Condition

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Storage temperature	Tstg		–55		150	°C
Junction temperature	Tj	R7F7016xx3AFP	–40		130	°C
		R7F7016xx4AFP	–40		150	°C

Note: xx = 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95

Regarding operation temperature of each product, see **Section 1C.3, RH850/F1KM Product Lineup**.

47C.3 Operational Condition

47C.3.1 Recommended Operating Conditions

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CPU clock frequency	$f_{\mathrm{CPUCLK\_M}}$				120	MHz
	$f_{\mathrm{CPUCLK\_L}}$	for OSTMn			60	MHz
Peripheral clock (clock domain) frequency[*1]	$f_{\mathrm{CKSCLK\_AWDTA}}$	for WDTA0			240[*2]	kHz
	$f_{\mathrm{CKSCLK\_ATAUJ}}$	for TAUJ0			40	MHz
		for TAUJ2				
	$f_{\mathrm{CKSCLK\_ARTCA}}$	for RTCA0			4	MHz
	$f_{\mathrm{CKSCLK\_AADCA}}$	for ADCA0			40	MHz
	$f_{\mathrm{CKSCLK\_AFOUT}}$	for FOUT			24	MHz
	$f_{\mathrm{CKSCLK\_IPERI1}}$	for TAUD0			80	MHz
		for TAUJ1				
		for TAUJ3				
		for ENCA0				
		for TAPA0				
		for PIC0				
	$f_{\mathrm{CKSCLK\_IPERI2}}$	for TAUB0			40	MHz
		for RCFDCn (clkc)				
		for RSENTn				
		for PWBAn				
		for PWGAn				
		for PWSAn				
	$f_{\mathrm{CKSCLK\_ILIN}}$	for RLIN24n			40	MHz
		for RLIN3n				
	$f_{\mathrm{CKSCLK\_ICAN}}$	for RCFDCn (pclk)			80	MHz
	$f_{\mathrm{CKSCLK\_ICANOSC}}$	for RCFDCn (clk_xincan)			24	MHz
	$f_{\mathrm{CKSCLK\_ICSI}}$	for CSIGn			80	MHz
		for CSIHn				
	$f_{\mathrm{LS\ IntOSC}}$	for WDTA1			240[*2]	kHz
	$f_{\mathrm{CKSCLK\_IIIC}}$	for RIICn			40	MHz
	f_{EMCLK}	for LPSn			8	MHz
Power supply	REGVCC	REGVCC = EVCC	VPOC[*3]		5.5	V
	EVCC					
	A0VREF		3.0		5.5	V
Normal operation voltage	AWOVCL		1.1	1.25	1.35	V
	ISOVCL					
Limited operation voltage[*4]	AWOVCL		1.35		1.43	V
	ISOVCL					

Note 1. For clock specification of peripherals, see **Section 12C, Clock Controller of RH850/F1KM-S1**.

Note 2. This frequency depends on the internal oscillator (LS IntOSC).

Note 3. "VPOC" means POC (power-on clear) detection voltage (TYP. 2.85 V). For detail, see **Section 47C.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.
In addition, the guaranteed operation in DC characteristic.
And AC characteristic is guaranteed when more than 3.0 V.
When the power supply voltage is VPOC to 3.0 V, the device does not malfunction.

Note 4. Reliability restrictions from 1.35 V to 1.43 V.

47C.3.2 Oscillator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C

(1) MainOSC (In Case of Using a Crystal/Ceramic)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MainOSC frequency*3	f_{MOSC}		8		24	MHz
MainOSC current consumption	I_{MOSC}	After stabilization		1.9*2	2.3*2	mA
MainOSC oscillation start point	V_{MOSCSP}		VPOC			V
MainOSC oscillation operating point	V_{MOSCOP}			0.5 × REGVCC*2		V
MainOSC oscillation amplitude	$V_{MOSCAMP}$		0.4 × REGVCC − 0.2*2			V
MainOSC oscillation stabilization time	t_{MSTB}			2*1, *2		ms
MainOSC transconductance	g_{m_MOSC}	MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 00		11.1*1		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 01		10.6*1		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 10		9.3*1		mA/V
		MOSCS.MOSCCLKACT = 0, MOSCC.MOSCAMPSEL[1:0] = 11		7.8*1		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 00		8.6*1		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 01		7.8*1		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 10		6.1*1		mA/V
		MOSCS.MOSCCLKACT = 1, MOSCC.MOSCAMPSEL[1:0] = 11		4.0*1		mA/V

Note 1. Oscillator stabilization time is time until being set (“1”) in MOSCS.MOSCCLKACT bit after MOSCE.MOSCENTRG bit is written “1”, and depends on the setting value of MOSCST register. Please decide appropriate oscillation stabilization time by matching test with resonator and oscillation circuit.

Note 2. This is reference value.

Note 3. The following four crystal/ceramic resonator frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

(2) MainOSC (In Case of External Clock Input to X1)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
X1 clock Input frequency*[1]	f_{EX}		8		24	MHz
X1 clock Input cycle time	t_{EXCYC}		41.7		125	ns
X1 High level Input voltage	V_{IH}		0.7 × REGVCC		REGVCC + 0.5	V
		@Flash Programing Interface*[2]	0.8 × REGVCC		REGVCC + 0.5	V
X1 Low level Input voltage	V_{IL}		–0.5		0.3 × REGVCC	V
		@Flash Programing Interface*[2]	–0.5		0.2 × REGVCC	V
X1 Input leakage current	I_{LIH}	VI = REGVCC			0.5	μA
	I_{LIL}	VI = 0 V			–0.5	μA
X1 clock Input low-level pulse width	t_{EXL}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input high-level pulse width	t_{EXH}	f_{EX} = 8 MHz	58			ns
		f_{EX} = 16 MHz	26			ns
		f_{EX} = 20 MHz	20			ns
		f_{EX} = 24 MHz	16			ns
X1 clock Input period jitter			–0.3		0.3	ns

Note 1. The following four external clock input frequencies are supported: 8 MHz, 16 MHz, 20 MHz and 24 MHz.

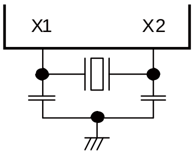
Note 2. X2 should be open and its parasitic capacitance should be less than 5 pF.

CAUTION

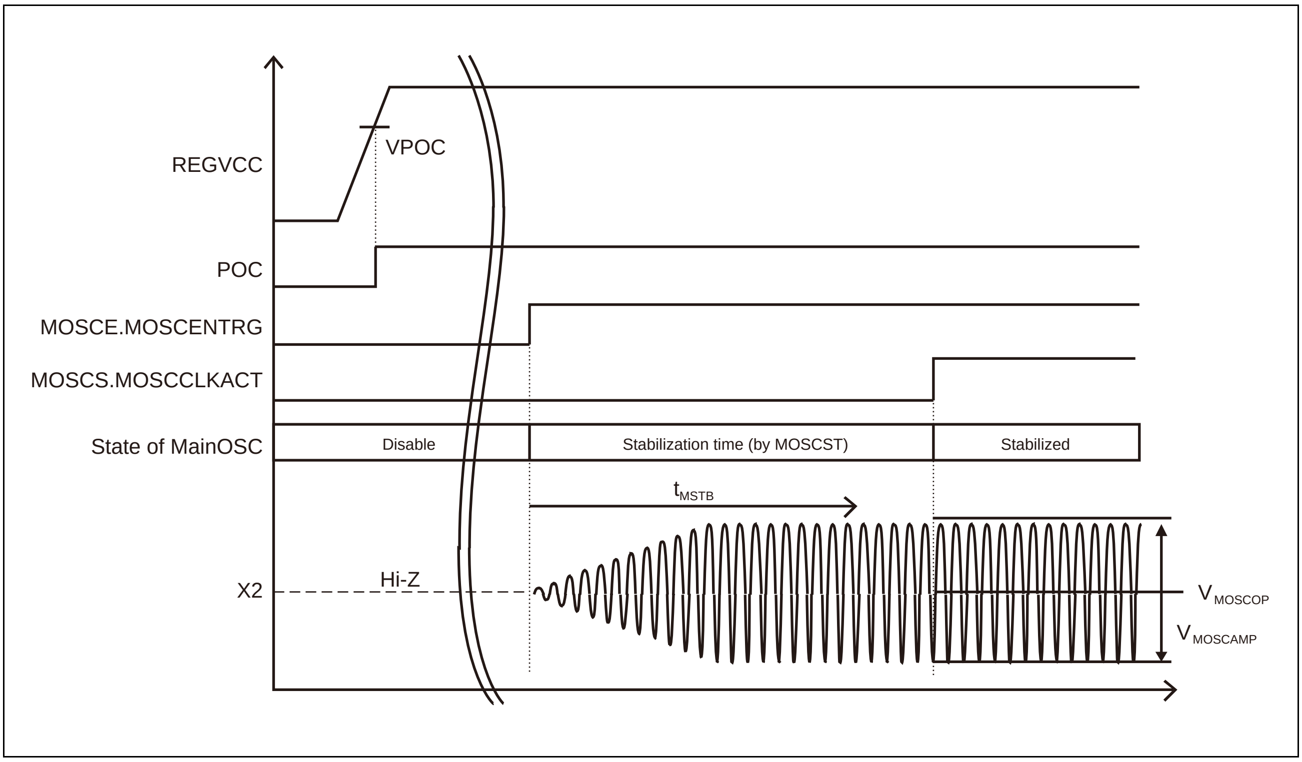
The oscillation stabilization time differs according the matching with the external resonator circuit. It is recommended to determine the oscillation stabilization time by an oscillator matching test.

NOTE

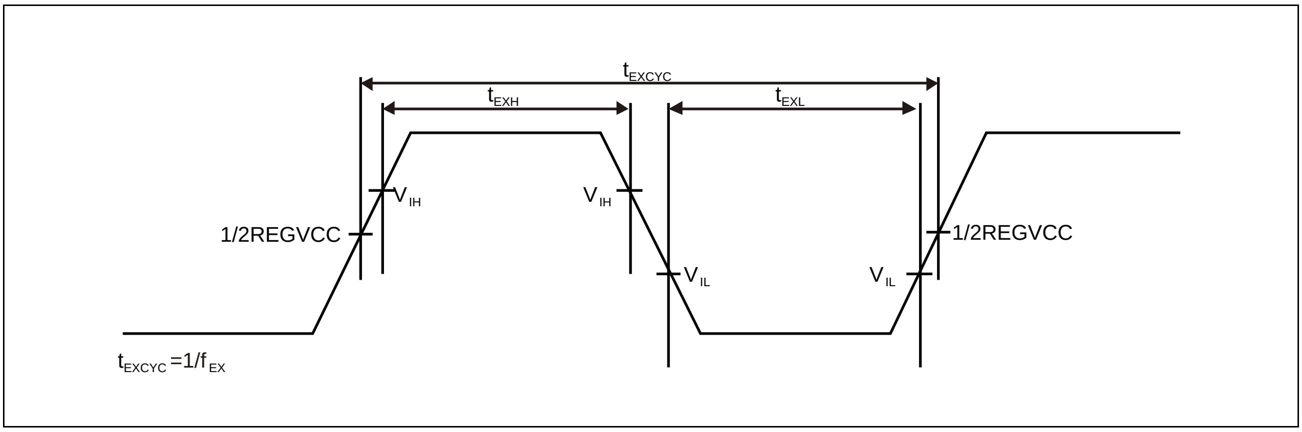
Recommended oscillator circuit is shown below.



MainOSC



External clock



47C.3.3 Internal Oscillator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LS IntOSC frequency	f_{RL}		220.8	240	259.2	kHz
HS IntOSC frequency*2, *3	f_{RH}		7.6	8	8.4	MHz
		After user trimming @ trimming temp	7.92	8	8.08	MHz
HS IntOSC current consumption	I_{RH}	After stabilization			170*1	µA
HS IntOSC oscillation stabilization time	t_{RHSTB}				54.4	µs

Note 1. This is reference value.

Note 2. The HS IntOSC frequency may not meet the specification range (8.00 MHz ±0.4 MHz, 8.00 MHz ±0.08 MHz after user trimming @ trimming temp) in the while writing/erasing the code/data flash.

Note 3. The HS IntOSC frequency may not meet the specification range in the Cyclic STOP/Cyclic RUN mode.

47C.3.4 PLL Characteristics

47C.3.4.1 PLL1 (for CPU/Peripheral) Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input frequency	$f_{PLL1CLKIN}$	MainOSC		8		24	MHz
		HS IntOSC	After user trimming @ trimming temp*3, *4	7.92	8.0	8.08	MHz
Output frequency	$f_{CPLL1OUT}$	MainOSC		80		120	MHz
		HS IntOSC*3	After user trimming @ trimming temp*3	79.2	80	80.8	MHz
	$f_{PPLLOUT}$			79.2	80	80.8	MHz
Output period jitter*1	t_{CPJ1}			–100		100	ps
Long term jitter*1	t_{LTJ}	term = 1 µs		–500		500	ps
		term = 10 µs		–1		1	ns
		term = 20 µs		–2		2	ns
Lock time*2	t_{LCK1}			104	112.3	122.1	µs

Note 1. This is reference value.

Note 2. Lock time is time until being set (“1”) in PLL1S.PLL1CLKACT bit after PLL1E.PLL1ENTRG bit is written “1”.

Note 3. The HS IntOSC has a frequency deviation. When the HSIntOSC is used the frequency deviation should be considered for the customer application as it affects peripheral functions (e.g. TAUx, ADCAn, etc.).

Note 4. Do not select the PLL1 as clock source during the code/data flash write and/or erase.

47C.4 DC Characteristics

47C.4.1 Capacitance

Condition: REGVCC = EVCC = A0VREF = AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
Ta = 25°C

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input capacitance	CI*[1]	f = 1 MHz 0 V for non measurement pins			10	pF
Input/output capacitance	CIO*[2]				10	pF

Note 1. CI: Capacitance between the input pin and ground

Note 2. CIO: Capacitance between the input/output pin and ground

47C.4.2 Pin Characteristics

Condition: Some of the conditions mentioned in this chapter can be selected by software and described in the hardware user's manual.

(1/2)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
$\overline{\text{RESET}}$	—	—	✓	—	—	—	—	—	—
FLMD0	—	✓	—	—	—	—	—	✓	✓
AP0_0	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_1	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_2	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_3	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_4	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_5	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_6	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_7	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_8	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_9	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_10	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_11	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_12	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_13	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_14	✓	—	—	—	—	✓	Slow	—	✓*1
AP0_15	✓	—	—	—	—	✓	Slow	—	✓*1
JP0_0	—	✓	—	✓	✓	—	Slow	✓	✓
JP0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
JP0_2	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_3	—	✓	—	✓	✓	—	Slow/Fast	✓	✓
JP0_4	—	—	—	✓	—*5	—	Slow	✓	✓
JP0_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_2	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_3	—	✓	—	✓	—	—	Slow/Fast*2	✓	✓
P0_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_5	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P0_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P0_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_1	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓

(2/2)

Port Input Buffer Function							Port Output Drive Strength Mode	Other Port Function	
Pin Name	CMOS	SHMT1	SHMT2	SHMT4	TTL	Analog		Pull-up	Pull-down
P10_2	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P10_3	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_6	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_7	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_8	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_9	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_10	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_11	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_12	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_13	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_14	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P10_15	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_0	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_1	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_2	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_3	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_4	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_5	—	✓	—	✓	—	—	Slow/Fast	✓	✓
P11_6	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P11_7	—	✓	—	✓	—	—	Slow/Fast*3	✓	✓
P8_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_4	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_5	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_6	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_7	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_8	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_9	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_10	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_11	—	✓	—	✓	—	✓	Slow	✓	✓*4
P8_12	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_0	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_1	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_2	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_3	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_4	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_5	—	✓	—	✓	—	✓	Slow	✓	✓*4
P9_6	—	✓	—	✓	—	✓	Slow	✓	✓*4

Note 1. Pull-down resistor for ADC diagnostic purpose. Control via ADC self-diagnostic register.

Note 2. Supports Cload: 100 pF

Note 3. Supports Cload: 50 pF

Note 4. Pull-down resistors for ADC diagnostic and internal pull-down purposes. For ADC diagnostic, control via ADC self-diagnostic register. For internal pull-down, control via PD register.

Note 5. TTL is selected for Boundary scan mode or Nexus in normal operating mode.

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

(1/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High level input voltage	VIH	CMOS		0.65 ×IOVCC		IOVCC + 0.3	V
		SHMT1*3		0.65 × IOVCC		IOVCC + 0.3	V
		SHMT2		0.75 × IOVCC		IOVCC + 0.3	V
		SHMT4		0.8 × IOVCC		IOVCC + 0.3	V
		TTL	EVCC = VPOC to 3.6 V	2.0		IOVCC + 0.3	V
			EVCC = 3.6 V to 5.5 V	2.2		IOVCC + 0.3	V
Low level input voltage	VIL	CMOS		–0.3		0.35 × IOVCC	V
		SHMT1		–0.3		0.35 × IOVCC	V
		SHMT2		–0.3		0.25 × IOVCC	V
		SHMT4		–0.3		0.5 × IOVCC	V
		TTL		–0.3		0.8	V
Input hysteresis for Schmitt	VH	SHMT1		0.3			V
		SHMT2		0.2 × IOVCC			V
		SHMT4		0.1			V
Input leakage current	ILIH	$\overline{\text{RESET}}$, FLMD0, JP0, P0, P8, P9, P10, P11 pin, VI = EVCC*2				0.5	µA
		AP0 pin, VI = A0VREF*2, Tj ≤ 130°C				0.3	µA
		AP0 pin, VI = A0VREF*2				0.5	µA
	ILIL	$\overline{\text{RESET}}$, FLMD0, JP0, P0, P8, P9, P10, P11 pin, VI = 0 V*2				–0.5	µA
		AP0 pin, VI = 0 V*2, Tj ≤ 130°C				–0.3	µA
		AP0 pin, VI = 0 V*2				–0.5	µA
Internal pull-up resistance	RU	except FLMD0 pin, VI = 0 V		20 (275 µA)	40	100	kΩ
		FLMD0 pin, VI = 0 V*3		4 (1375 µA)		36	kΩ
Internal pull-down resistance	RD	except FLMD0 pin, VI = EVCC		20 (275 µA)	40	100	kΩ
		FLMD0 pin, VI = EVCC		4 (1375 µA)		36	kΩ
High level output voltage	VOH	Fast mode	IOH = –5 mA (6 pins) *4	IOVCC – 1.0			V
			IOH = –3 mA (10 pins) *4	IOVCC – 1.0			V
			IOH = –1 mA (16 pins) *4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins) *4	IOVCC – 0.5			V
		Slow mode	IOH = –1 mA (16 pins) *4	IOVCC – 0.5			V
			IOH = –0.1 mA (16 pins) *4	IOVCC – 0.5			V
Low level output voltage	VOL	Fast mode	IOL = 5 mA (6 pins)*4			0.4	V
			IOL = 3 mA (10 pins)*4			0.4	V
			IOL = 1 mA (16 pins)*4			0.4	V
		Slow mode	IOL = 1 mA (16 pins)*4			0.4	V

(2/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Rise/Fall time	t_{KRP}/t_{KFP}	Fast mode (except below pins)*5	CL = 30 pF			7	ns
			CL = 50 pF			12	ns
			CL = 100 pF			24	ns
		Fast mode (P0_5, P0_6, P10_1, P10_2, P11_2, P11_3, P11_6, P11_7)*6	CL = 50 pF			6	ns
		Fast mode (P0_2, P0_3)*6	CL = 100 pF			6.15	ns
		Slow mode*5	CL = 30 pF			37	ns
			CL = 50 pF			62	ns
			CL = 100 pF			124	ns
Output frequency	f_O	Fast mode	CL = 30 pF			40	MHz
		Slow mode	CL = 30 pF			10	MHz
			CL = 50 pF			6	MHz
			CL = 100 pF			3	MHz

Note 1. “IOVCC” means the pins are assigned to the power supply (EVCC, and A0VREF).

Note 2. Not select the analog input function of ADCn.

Note 3. When the internal pull-up resistor of FLMD0 pin is applied by FLMDCNT register, please connect 86 kΩ or more as external pull-down resistor.

Note 4. The number of pin indicates simultaneous ON.

Note 5. Measurement point: 0.1 × IOVCC to 0.9 × IOVCC

Note 6. Measurement point: 0.2 × IOVCC to 0.8 × IOVCC

47C.4.2.1 Output Current

(1) 100-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_6			–7	mA
				P0_0 to P0_3, P10_3 to P10_5, P10_15, P11_0			–30	mA
				JP0_3 to JP0_5, P0_4 to P0_6, P0_11 to P0_14, P8_2, P8_10 to P8_12			–30	mA
				JP0_0 to JP0_2			–11	mA
				P0_7 to P0_10, P8_0, P8_1, P8_3 to P8_9			–29	mA
				P10_6 to P10_9			–20	mA
				P10_10 to P10_14, P11_1 to P11_7			–30	mA
				P10_0 to P10_2			–15	mA
			Total (EVCC)				–60	mA
		PgA0	Total (A0VREF)				–16	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_6			7	mA
				P0_0 to P0_6, P0_11 to P0_14, P10_3 to P10_5, P10_15, P11_0			30	mA
				JP0_0 to JP0_5, P8_2, P8_10 to P8_12			26	mA
				P0_7 to P0_10			20	mA
				P8_0, P8_1, P8_3 to P8_9			9	mA
				P10_6 to P10_14, P11_1, P11_2			30	mA
				P11_3 to P11_7			25	mA
				P10_0 to P10_2			15	mA
			Total (EVSS)				60	mA
		PgA0	Total (A0VSS)				16	mA

Note: For detail of the definition of “side” and “total”, see **Section 47C.2.3, Port Current**.

(2) 80-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_6, P10_6 to P10_14, P11_1 to P11_4			–30	mA
				P10_0 to P10_2			–15	mA
				P0_0 to P0_3, P10_3 to P10_5, P10_15, P11_0			–30	mA
				JP0_0 to JP0_5, P0_4 to P0_12, P8_0 to P8_6			–30	mA
			Total (EVCC)				–60	mA
		PgA0	Total (A0VREF)				–11	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_6, P10_6 to P10_14, P11_1 to P11_4			30	mA
				P10_0 to P10_2			15	mA
				P0_0 to P0_6, P0_11, P0_12, P10_3 to P10_5, P10_15, P11_0			30	mA
				JP0_0 to JP0_5, P0_7 to P0_10, P8_0 to P8_6			21	mA
			Total (EVSS)				60	mA
		PgA0	Total (A0VSS)				11	mA

Note: For detail of the definition of “side” and “total”, see **Section 47C.2.3, Port Current**.

(3) 64-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0 to P9_3, P10_6 to P10_14			–30	mA
				P10_0 to P10_2			–15	mA
				P0_0 to P0_3, P10_3 to P10_5			–27	mA
				JP0_0 to JP0_5, P0_4 to P0_6, P8_0 to P8_6			–24	mA
			Total (EVCC)				–60	mA
		PgA0	Total (A0VREF)				–10	mA
Low-level output current	IOL	PgE	Per side	P9_0 to P9_3, P10_6 to P10_14			30	mA
				P10_0 to P10_2			15	mA
				P0_0 to P0_6, P10_3 to P10_5			30	mA
				JP0_0 to JP0_5, P8_0 to P8_6			13	mA
			Total (EVSS)				60	mA
		PgA0	Total (A0VSS)				10	mA

Note: For detail of the definition of “side” and “total”, see **Section 47C.2.3, Port Current**.

(4) 48-Pin Version

Item	Symbol	Pin Group	Condition		MIN.	TYP.	MAX.	Unit
High-level output current	IOH	PgE	Per side	P9_0, P9_1, P10_6 to P10_10			–27	mA
				P10_0 to P10_2			–15	mA
				P0_0 to P0_3, P10_3 to P10_5			–27	mA
				JP0_0 to JP0_5, P8_0, P8_1			–8	mA
			Total (EVCC)				–60	mA
		PgA0	Total (A0VREF)				–8	mA
Low-level output current	IOL	PgE	Per side	P9_0, P9_1, P10_6 to P10_10			27	mA
				P10_0 to P10_2			15	mA
				P0_0 to P0_3, P10_3 to P10_5			27	mA
				JP0_0 to JP0_5, P8_0, P8_1			8	mA
			Total (EVSS)				60	mA
		PgA0	Total (A0VSS)				8	mA

Note: For detail of the definition of “side” and “total”, see **Section 47C.2.3, Port Current**.

47C.4.3 Power Supply Currents

Condition: REGVCC, EVCC, A0VREF total current. But the I/O buffer is stopped.

Item	Symbol	Condition				MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2				
RUN mode current	IDDR	Run (120 MHz)	Run	–40 to 150°C	Run(#1)		32	67	mA
				25°C	Stop(#1)		26		mA
RUN mode current (During data/code flash programming)	IDDR3	Run (120 MHz)	Run	–40 to 150°C	Run(#2)		43	82	mA
RUN mode current (HALT state)	IDDH	Run (120 MHz)	Run	–40 to 150°C	Run(#3)		29	63	mA

Item	Symbol	Condition				MIN.	TYP.*1	MAX.	Unit
		CPU	PLL	Tj	Peripheral*2				
STOP mode current	IDDS	Stop	Stop	–40 to 90°C	Stop(#2)		0.7	12	mA
				110°C	Stop(#2)			17	mA
				135°C	Stop(#2)			31	mA
DeepSTOP mode current	IDDDS	Power off	Power off	–40 to 85°C	Stop(#3)		50	470	μA
				105°C	Stop(#3)			830	μA
				125°C	Stop(#3)			1370	μA
Cyclic RUN mode current	IDDCR	Run (HS IntOSC)	Stop	–40 to 90°C	Run(#4)		3.6	21	mA
				115°C	Run(#4)			28	mA
				135°C	Run(#4)			40	mA
Cyclic STOP mode current	IDDCS	Stop	Stop	–40 to 90°C	Run(#5)		1.1	13	mA
				110°C	Run(#5)			18	mA
				135°C	Run(#5)			32	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- REGVCC = EVCC = A0VREF = 5.0 V
- AWOVSS = EVSS = A0VSS = 0 V

Note 2. Operating condition of each peripheral function is shown in the table of next page.

Caution: It must be ensured that the junction temperature in the Ta range remains below Tj ≤ 150°C and does not exceed its limit under application conditions (thermal resistance, power supply current, peripheral current (if not included in power supply current), port output current and injection current).

Function		Run					Stop		
		(#1)	(#2)	(#3)	(#4)	(#5)	(#1)	(#2)	(#3)
AWO	MainOSC	Run	Run	Run	Stop	Stop	Run	Stop	Stop
	HS IntOSC	Run	Run	Run	Run	Stop	Run	Stop	Stop
	FOUT	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	LPS	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	RRAM	Read/Write	Read/Write	No access	Fetch	No access	Read/Write	No access	No access
	WDTA0	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop
	TAUJ0, TAUJ2	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Stop	Stop	Stop
	RTCA0	Run	Run	Run	Run (LS IntOSC)	Run (LS IntOSC)	Stop	Stop	Stop
	CLMA0	Run	Run	Run	Run	Stop	Stop	Stop	Stop
	CLMA1	Run	Run	Run	Stop	Stop	Stop	Stop	Stop
	ADCA0	Run*[1]	Run*[1]	Run*[1]	Stop	Stop	Stop	Stop	Stop
ISO	CPU	Run (PLL1)	Run (PLL1)	HALT (PLL1)	Run (HS IntOSC)	Stop	Run (PLL1)	Stop	Power off
	DMA	Run	Run	Run	Stop	Stop	Stop	Stop	
	PLL1	Run	Run	Run	Stop	Stop	Run	Stop	
	Code flash	Fetch	Fetch	No access	No access	No access	Fetch	No access	
	Data flash	Read	Write/Erase	No access	No access	No access	Read	No access	
	LRAM	Read/Write	Read/Write	No access	No access	No access	Read/Write	No access	
	OSTM0	Run	Run	Run	Stop	Stop	Stop	Stop	
	WDTA1	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	TAUD0	Run	Run	Run	Stop	Stop	Stop	Stop	
	TAUBn	Run	Run	Run	Stop	Stop	Stop	Stop	
	TAUJ1, TAUJ3	Run	Run	Run	Stop	Stop	Stop	Stop	
	TAPA, PIC	Stop	Stop	Stop	Stop	Stop	Stop	Stop	
	ENCA0	Run	Run	Run	Stop	Stop	Stop	Stop	
	PWM-diag	Run	Run	Run	Stop	Stop	Stop	Stop	
	RLIN3n	Run	Run	Run	Stop	Stop	Stop	Stop	
	RLIN24n	Wait	Wait	Wait	Stop	Stop	Stop	Stop	
	RCFDCn	Wait	Wait	Wait	Stop	Stop	Stop	Stop	
	CSIGn	Run	Run	Run	Stop	Stop	Stop	Stop	
	CSIHn	Run	Run	Run	Stop	Stop	Stop	Stop	
	RIICn	Wait	Wait	Wait	Stop	Stop	Stop	Stop	
	KR	Wait	Wait	Wait	Stop	Stop	Stop	Stop	
	RSENTn	Run	Run	Run	Stop	Stop	Stop	Stop	
	CLMA3	Run	Run	Run	Stop	Stop	Stop	Stop	

Note 1. T&H used.

47C.4.4 Injection Currents

Table 47C.1 Definition of Pin Group (100-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgE	EVCC, EVSS	JP0, P0, P10, P11
PgE’	EVCC, EVSS	P8, P9
PgA0	A0VREF, A0VSS	AP0

Table 47C.2 Definition of Pin Group (80-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgE	EVCC, EVSS	JP0, P0, P10, P11
PgE’	EVCC, EVSS	P8, P9
PgA0	A0VREF, A0VSS	AP0

Table 47C.3 Definition of Pin Group (64-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgE	EVCC, EVSS	JP0, P0, P10
PgE’	EVCC, EVSS	P8, P9
PgA0	A0VREF, A0VSS	AP0

Table 47C.4 Definition of Pin Group (48-Pin Version)

Symbol	Power Supply for Pin Group	Pin
PgE	EVCC, EVSS	JP0, P0, P10
PgE’	EVCC, EVSS	P8, P9
PgA0	A0VREF, A0VSS	AP0

47C.4.4.1 Absolute Maximum Ratings

(1) 100-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

(2) 80-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

(3) 64-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

(4) 48-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJPM}	PgE	Per pin			10	mA
			Total			60	mA
		PgE’	Per pin			10	mA
			Total			60	mA
		PgA0	Per pin			10	mA
			Total			60	mA
Negative overload current VIN < VSS	I_{INJNM}	PgE	Per pin			–10	mA
			Total			–60	mA
		PgE’	Per pin			–10	mA
			Total			–60	mA
		PgA0	Per pin			–10	mA
			Total			–60	mA

CAUTIONS

1. The DC injection current (Total) must satisfy the specifications of the injection current per pin.
2. In case of an injected current condition for PgA0, TESH0SN is kept when the injected current is applied to an adjacent pin where the ADC self-diagnosis is executed. When an injected current is applied to the same pin where the ADC self-diagnosis is executed the TESH0SN deviating value will increase sharply with increasing absolute value of injection current.

47C.4.4.2 DC Characteristics for Overload Current

(1) 100-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgE’	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgE’	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

(2) 80-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgE’	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgE’	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

(3) 64-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgE'	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgE'	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

(4) 48-Pin Version

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Positive overload current VIN > VCC	I_{INJP}	PgE	Per pin			2	mA
			Total			50	mA
		PgE'	Per pin			3	mA
			Total			20	mA
		PgA0	Per pin			3	mA
			Total			20	mA
Negative overload current VIN < VSS	I_{INJN}	PgE	Per pin			–2	mA
			Total			–50	mA
		PgE'	Per pin			–3	mA
			Total			–20	mA
		PgA0	Per pin			–3	mA
			Total			–20	mA

NOTE

These specifications are not tested on sorting and are specified based on the device characterization.

47C.4.5 Power Management Characteristics

47C.4.5.1 Regulator Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input voltage	REGVCC		VPOC*[1]		5.5	V
Output voltage	AWOVCL	AWOVCL pin	1.15	1.25	1.35	V
	ISOVCL	ISOVCL pin	1.15	1.25	1.35	V
Capacitance	CAWOVCL	AWOVCL pin	0.07	0.10	0.13	μF
	CISOVCL	ISOVCL pin	0.07	0.10	0.13	μF
Equivalent series resistance for load capacitance	RVRAWO	for CAWOVCL			40*[2]	mΩ
	RVRISO	for CISOVCL			40*[2]	mΩ
Inrush current during power-on					120	mA

Note 1. “VPOC” means POC (power-on clear) detection voltage (typ. 2.85 V). For detail, see **Section 47C.4.5.2, Voltage Detector (POC, LVI, VLVI, CVM) Characteristics**.

Note 2. This is reference value.

47C.4.5.2 Voltage Detector (POC, LVI, VLVI, CVM) Characteristics

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Detection voltage (REGVCC)	VPOC	POC			2.7	2.85	3.0	V
	VLVI0	LVI	Rise		3.87	4.0	4.13	V
			Fall		3.9	4.0	4.1	V
	VLVI1		Rise		3.57	3.7	3.83	V
			Fall		3.6	3.7	3.8	V
	VLVI2		Rise		3.37	3.5	3.63	V
			Fall		3.4	3.5	3.6	V
	VVLVI	VLVI			1.8	1.9	2.0	V
Detection voltage (ISOVCL)	VCVMH	CVM	High voltage[Caution]		1.35	1.39	1.43	V
	VCVML*8		Low voltage[Caution]		1.10	1.15	1.20	V
Response time	t_{D_POC1}*6	POC	At power-on (Rise)	*1			2	ms
				*2			6.3	ms
			After power-on (Rise)	*3			2	ms
				*4			5	ms
	t_{D_POC2}*7		After power-on (Fall)	*5			5	μs
	t_{D_LVI}	LVI					2	ms
	t_{D_VLVI}	VLVI		*3			2	ms
				*4			5	ms
	t_{D_CVM}	CVM			0.2		10	μs
Setup time	t_{S_LVI}	LVI	LVICNT0,1 bits are set to 1 (except 00_B), then LVI is ready to operate				80	μs
REGVCC minimum width	t_{W_POC}	POC			0.2			ms
	t_{W_LVI}	LVI			0.2			ms
	t_{W_VLVI}	VLVI			0.2			ms

Note 1. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms

Note 2. Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms

Note 3. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 20 V/ms

Note 4. Voltage slope (t_{VS}): 20 V/ms < t_{VS} ≤ 500 V/ms

Note 5. Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 500 V/ms

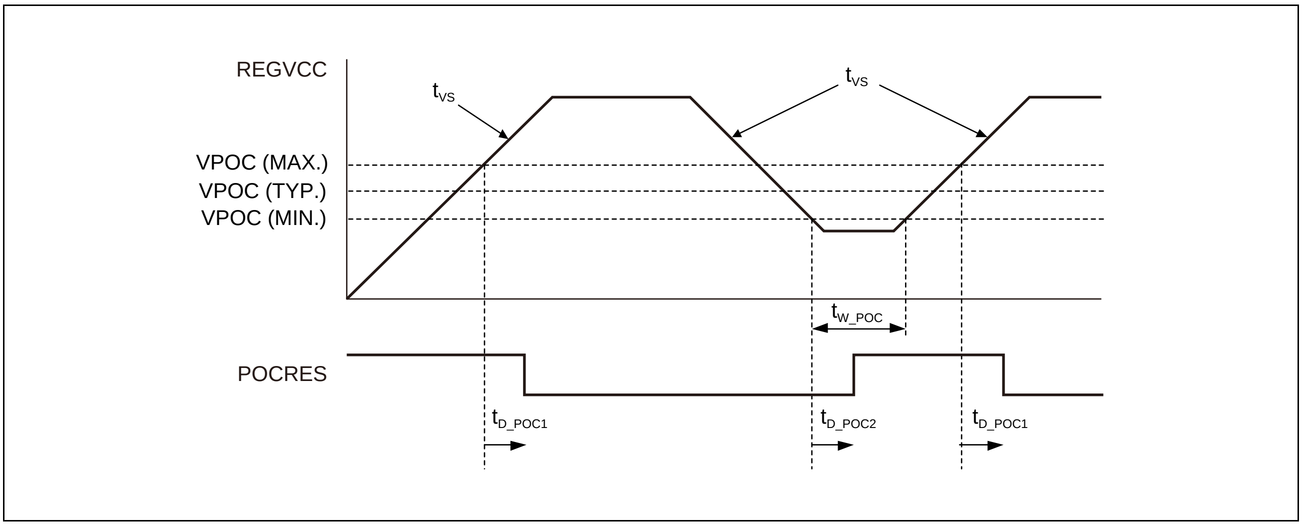
Note 6. t_{D_POC1} is the time from detection voltage to release of reset signal.

Note 7. t_{D_POC2} is the time from detection voltage to occurrence of reset signal.

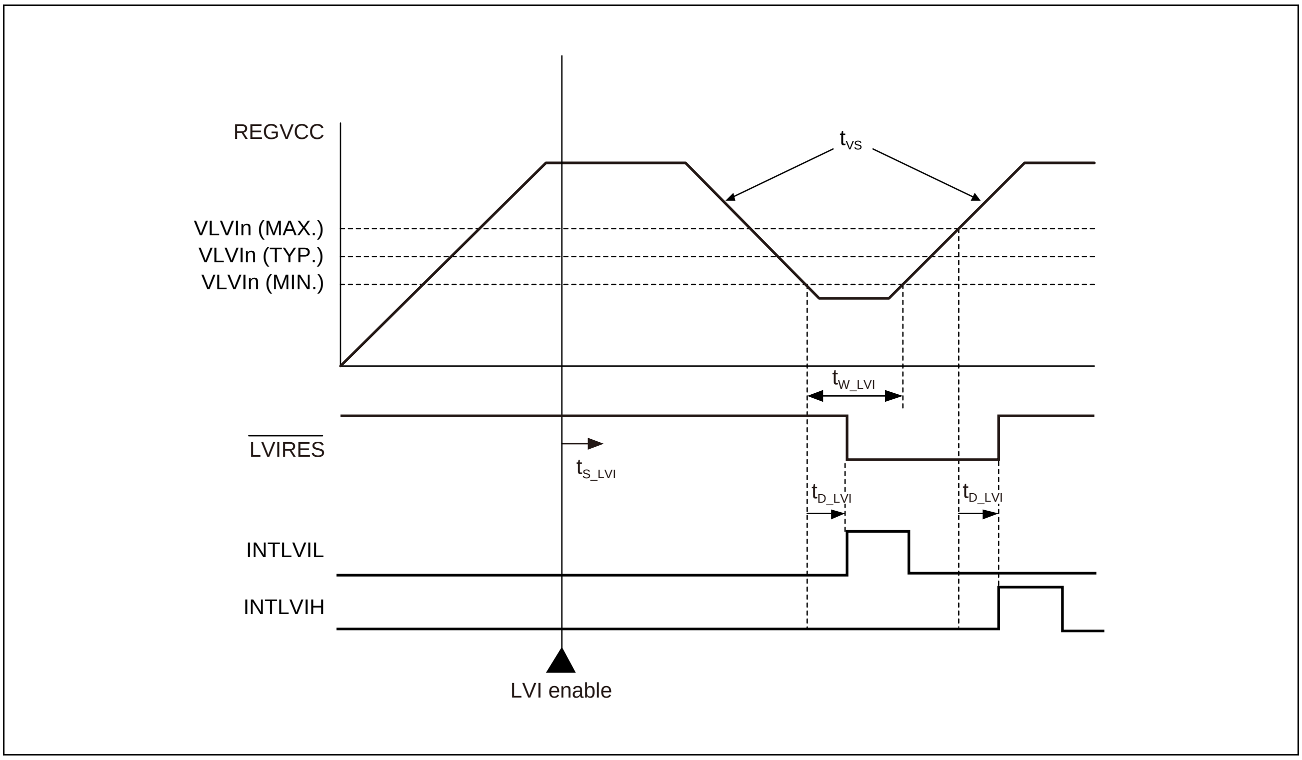
Note 8. The CVM monitors the internal voltage regulator output to ensure that ISOVCL is upper than specified minimum level.

Caution: A detection of the voltage ISOVCL outside the specified level of VCVMH and VCVML is not ensured by CVM.

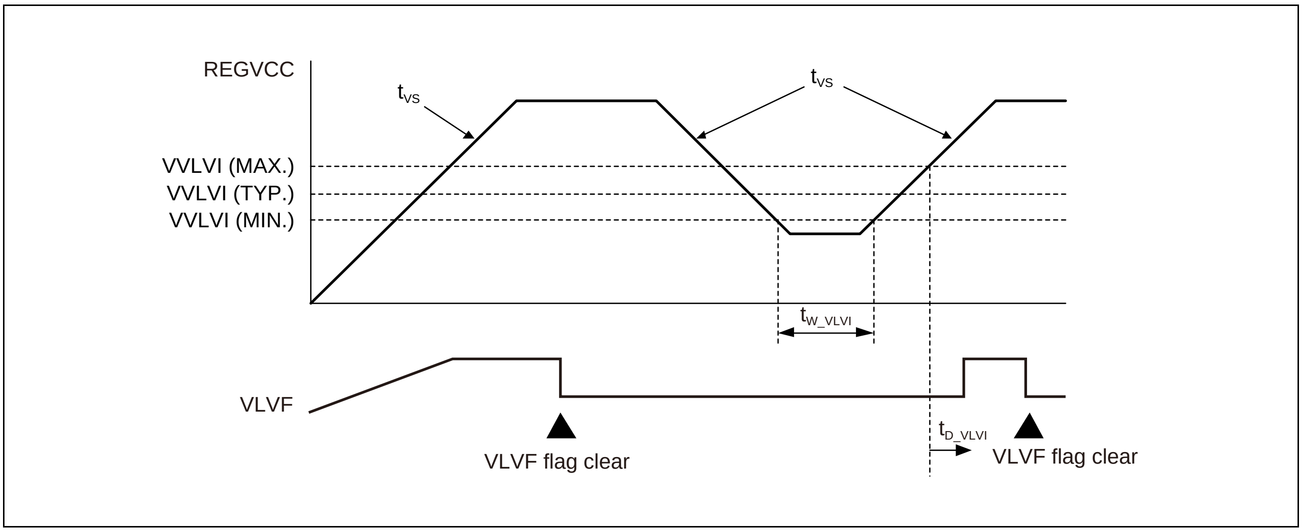
POC


REGVCC
t_{VS}
t_{VS}
VPOC (MAX.)
VPOC (TYP.)
VPOC (MIN.)
t_{W_POC}
POCRES
t_{D_POC1}
t_{D_POC2}
t_{D_POC1}

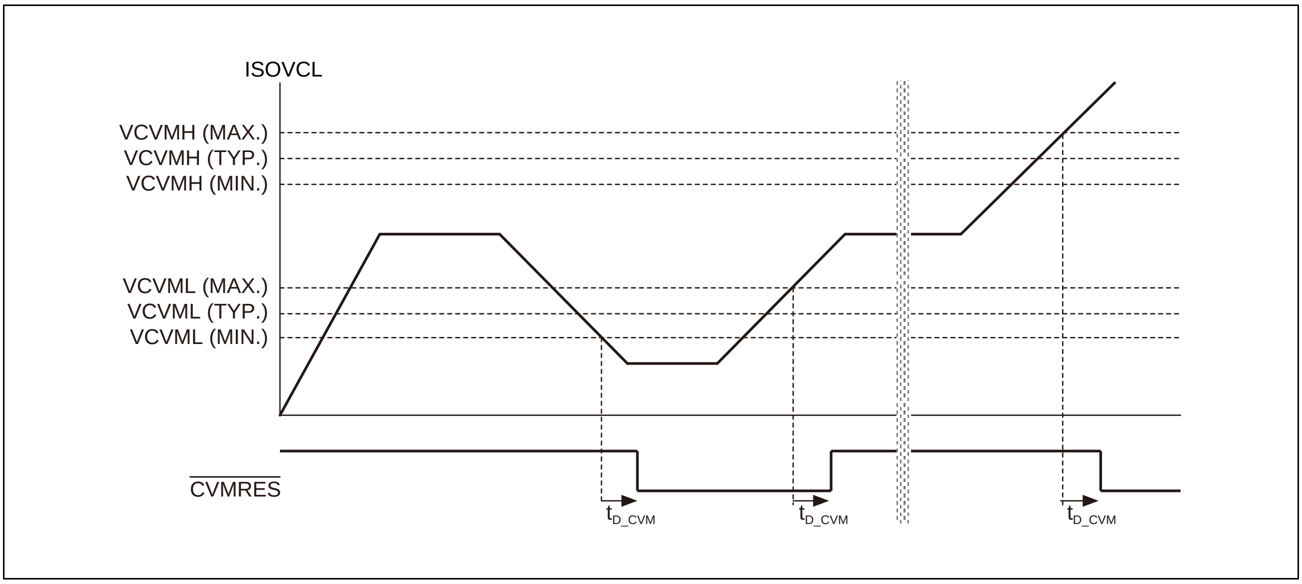
LVI


REGVCC
t_{VS}
VLVIn (MAX.)
VLVIn (TYP.)
VLVIn (MIN.)
t_{W_LVI}
$\overline{\text{LVIRES}}$
t_{S_LVI}
t_{D_LVI}
t_{D_LVI}
INTLVIL
INTLVIH
LVI enable

VLVI


REGVCC
t_{VS}
t_{VS}
VVLVI (MAX.)
VVLVI (TYP.)
VVLVI (MIN.)
t_{W_VLVI}
VLVF
VLVF flag clear
t_{D_VLVI}
VLVF flag clear

CVM


ISOVCL
VCVMH (MAX.)
VCVMH (TYP.)
VCVMH (MIN.)
VCVML (MAX.)
VCVML (TYP.)
VCVML (MIN.)
$\overline{\text{CVMRES}}$
t_{D_CVM}
t_{D_CVM}
t_{D_CVM}

47C.4.5.3 Power Up/Down Timing

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.5 In Case the $\overline{\text{RESET}}$ Pin is Used (for Normal Operating Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*[1])	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 µs/V)	V/ms
REGVCC ↑ and IOVCC*[1]↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↑)	t_{HMDR}		1			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↓)	t_{SMDF}		0			µs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*[1] ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, A0VREF.

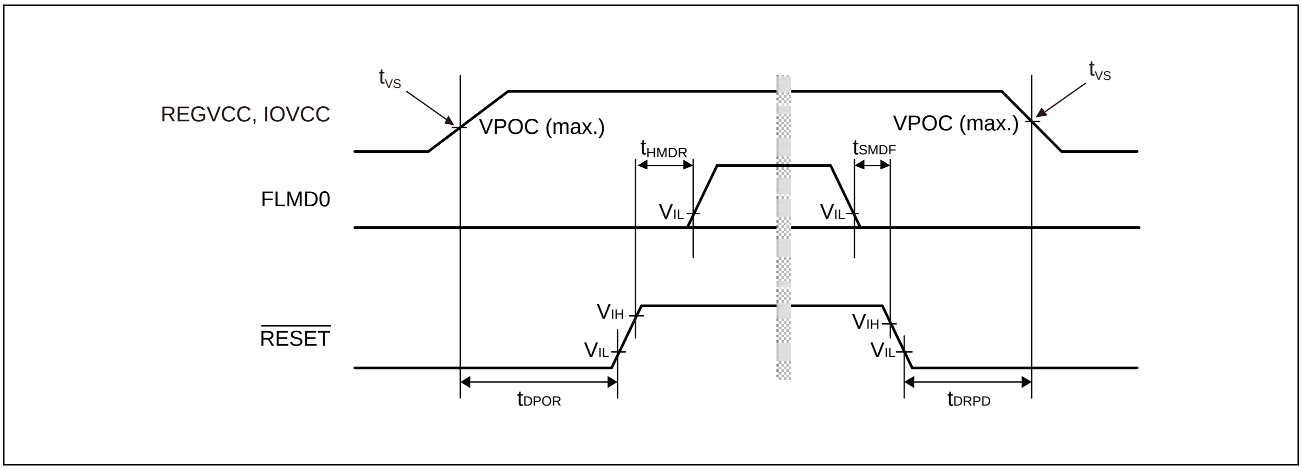


Table 47C.6 In Case the $\overline{\text{RESET}}$ Pin is Used (for Serial Programming Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*¹)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*¹ ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*¹ ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, A0VREF.

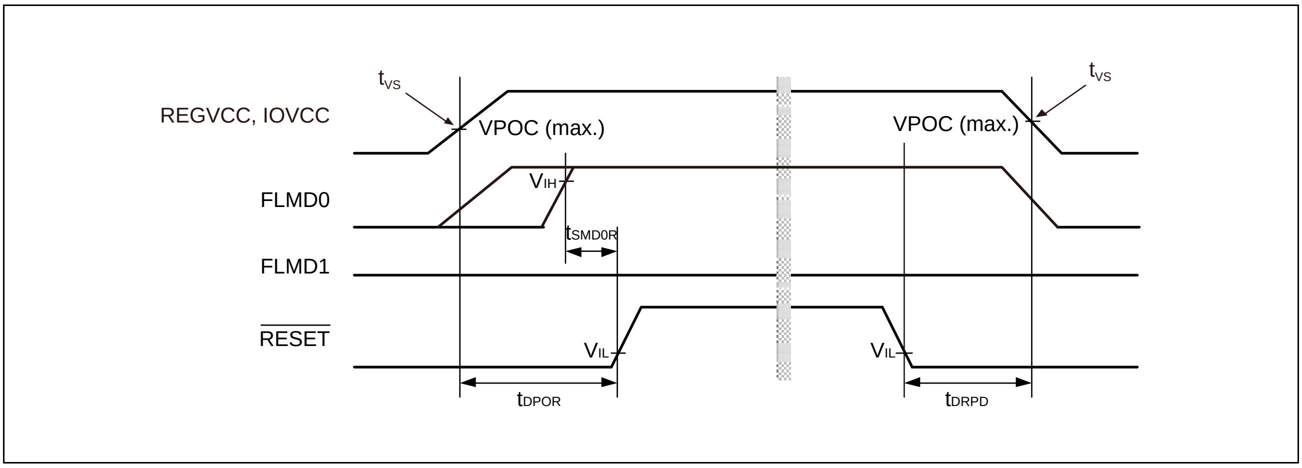


Table 47C.7 In Case the $\overline{\text{RESET}}$ Pin is Used (for Boundary Scan Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*[1])	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*[1] ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1, MODE0, MODE1 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			μs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			μs
FLMD1, MODE0, MODE1 MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			μs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC*[1] ↓ delay time	t_{DRPD}		0			ms
$\overline{\text{DCUTRST}}$ input delay time (vs $\overline{\text{RESET}}$ ↑)	t_{DRTRST}		1			ms
$\overline{\text{RESET}}$ hold time (vs $\overline{\text{DCUTRST}}$ ↓)	t_{HRTRST}		0			ms

Note 1. IOVCC means EVCC, and A0VREF.

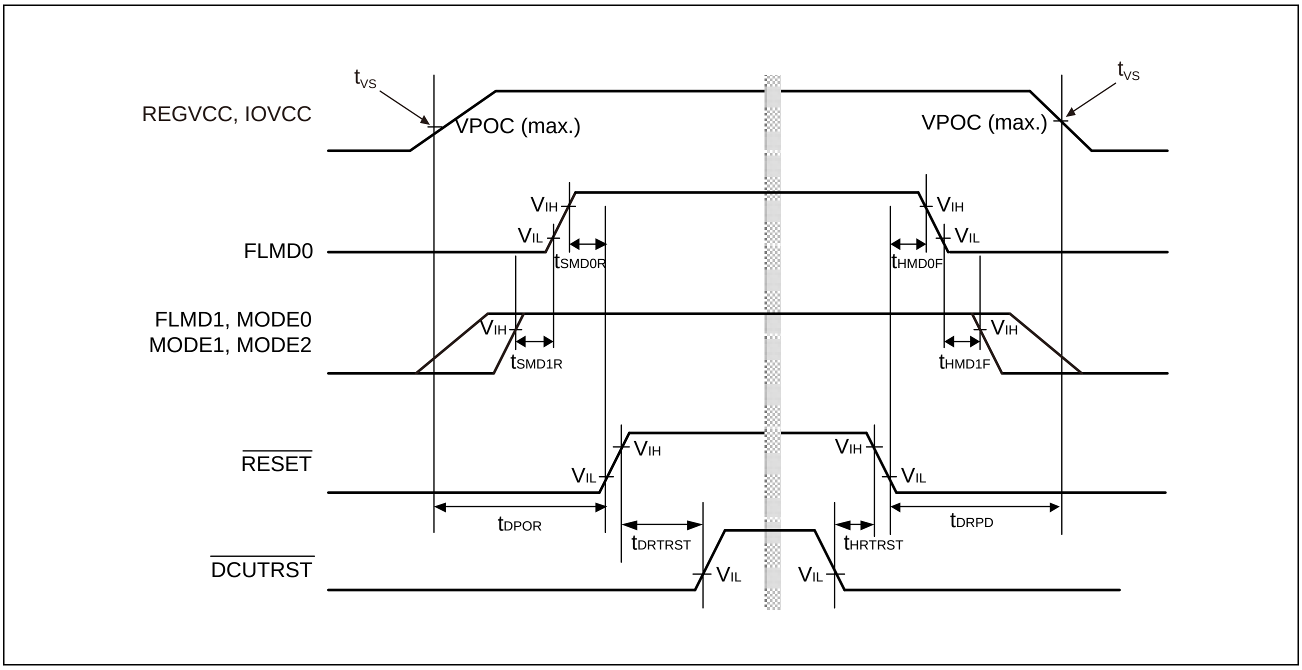


Table 47C.8 In Case the $\overline{\text{RESET}}$ Pin is Used (for User Boot Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*1)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*1 ↑ to $\overline{\text{RESET}}$ ↑ delay time	t_{DPOR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500V/ms	6.3			ms
FLMD0 setup time (vs $\overline{\text{RESET}}$ ↑)	t_{SMD0R}		1			ms
FLMD1, MODE0, MODE1,MODE2 setup time (vs FLMD0 ↑)	t_{SMD1R}		1			μs
FLMD0 hold time (vs $\overline{\text{RESET}}$ ↓)	t_{HMD0F}		1			μs
FLMD1, MODE0, MODE1, MODE2 hold time (vs FLMD0 ↓)	t_{HMD1F}		1			μs
$\overline{\text{RESET}}$ ↓ to REGVCC ↓ and IOVCC ↓ delay time	t_{DRPD}		0			ms

Note 1. IOVCC means EVCC, A0VREF.

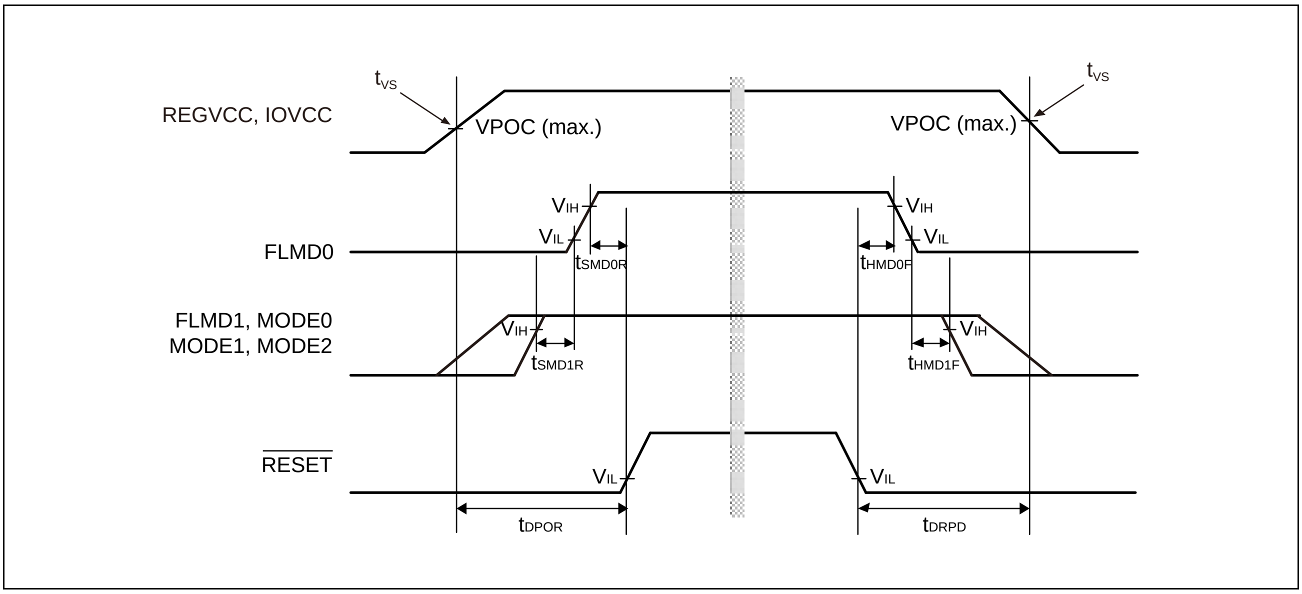
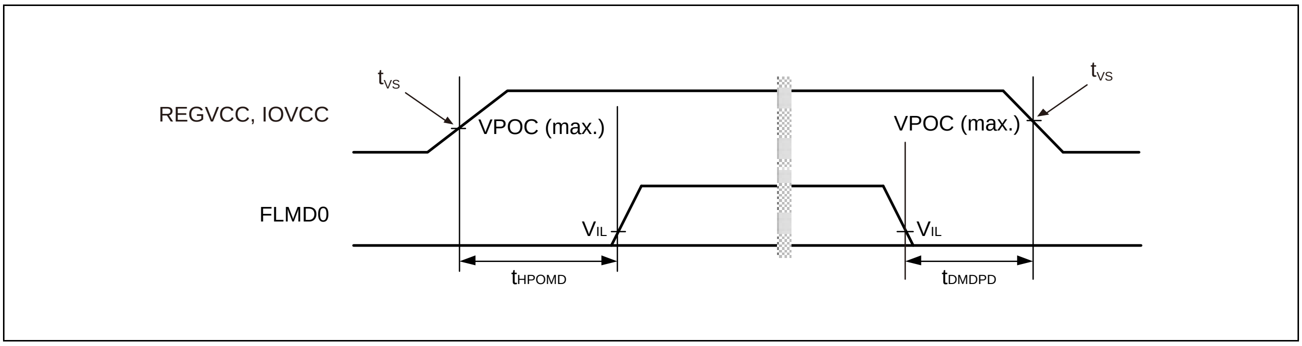


Table 47C.9 In Case the $\overline{\text{RESET}}$ Pin is Not Used and Fixed to High Level by Pull-up*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Voltage slope (REGVCC and IOVCC*2)	t_{VS}		0.02 (= 50 ms/V)		500 (= 2 μs/V)	V/ms
REGVCC ↑ and IOVCC*2 ↑ to FLMD0 hold time	t_{HPOMD}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms	2			ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms	6.3			ms
FLMD0 ↓ to REGVCC ↓ and IOVCC*2 ↓ delay time	t_{DMDPD}		1			μs

Note 1. This operating condition is available only in normal operation mode (include self-programming mode).
When the device is used in except normal operation mode, please use the $\overline{\text{RESET}}$ pin.

Note 2. IOVCC means EVCC, and A0VREF.



47C.4.5.4 CPU Reset Release Timing

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.10 In Case the $\overline{\text{RESET}}$ Pin is Not Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
REGVCC ↑ to CPU reset release*[1]	t_{DPCRR}	Voltage slope (t_{VS}): 0.02 V/ms ≤ t_{VS} ≤ 0.5 V/ms			2.58	ms
		Voltage slope (t_{VS}): 0.5 V/ms < t_{VS} ≤ 500 V/ms			8.3	ms

Note 1. This is reference value.

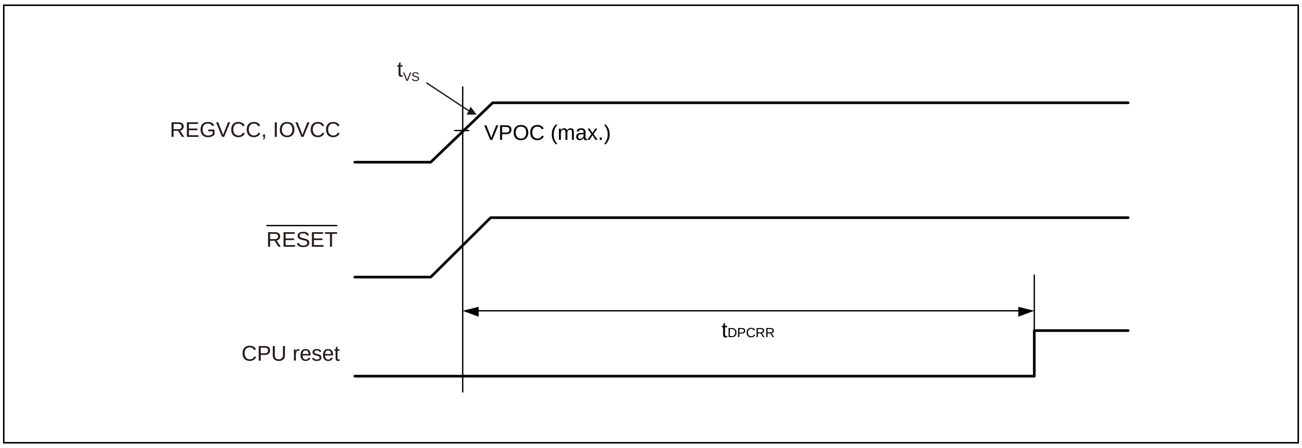
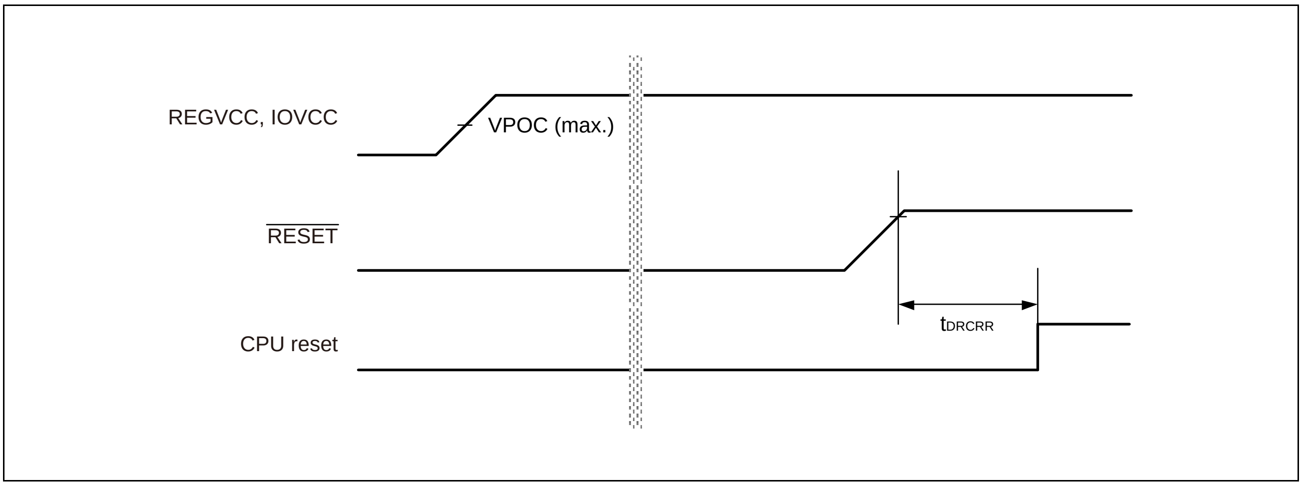


Table 47C.11 In Case the $\overline{\text{RESET}}$ Pin is Used

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ ↑ to CPU reset release*[1]	t_{DRCRR}				16*[2]	μs

Note 1. This is reference value.

Note 2. In case the time until releasing the $\overline{\text{RESET}}$ pin is longer than t_{DPCRR}.



47C.5 AC Characteristics

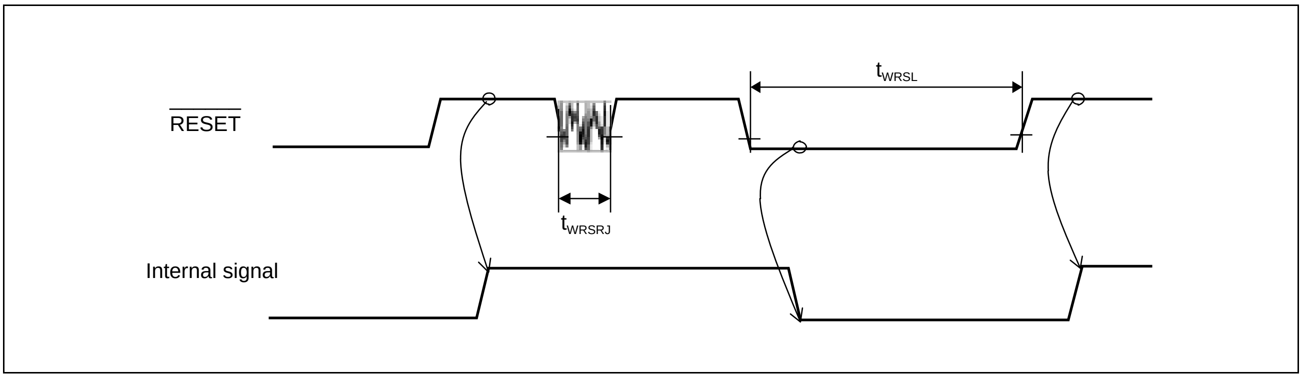
47C.5.1 $\overline{\text{RESET}}$ Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width*[1]	t_{WRSL}	Except power on	600			ns
$\overline{\text{RESET}}$ pulse rejection*[2]	t_{WRSRJ}		100			ns

Note 1. $\overline{\text{RESET}}$ input width is needed to ensure that the internal reset signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



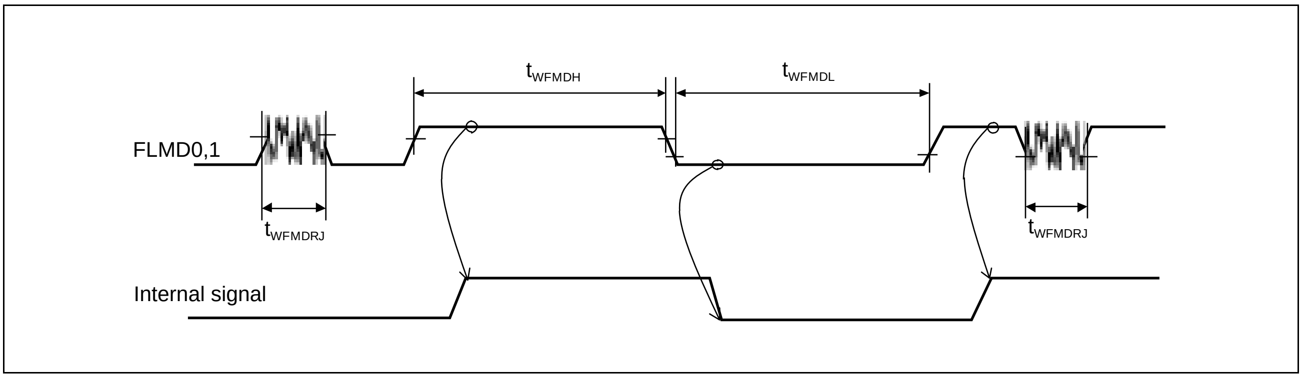
47C.5.2 Mode Timing

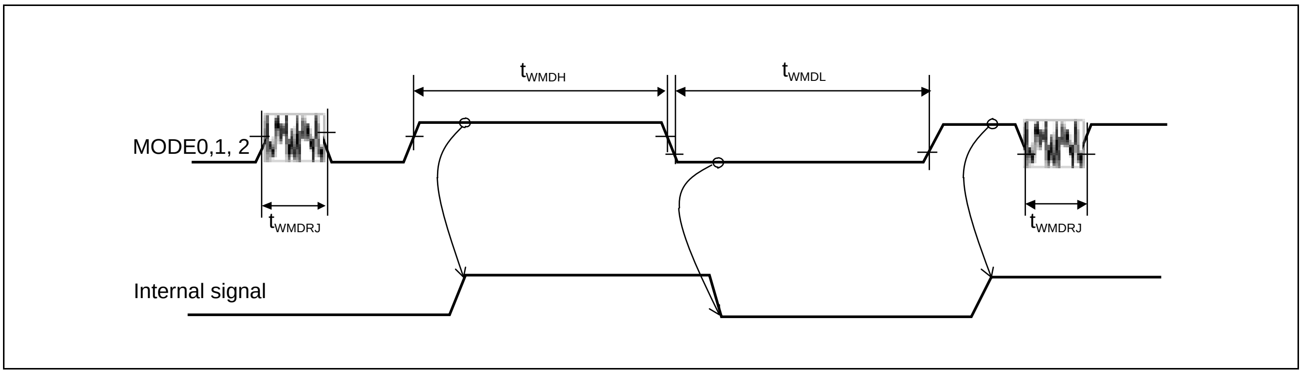
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0, 1 input high/low level width*[1]	t_{WFMDH}/ t_{WFMDL}		600			ns
FLMD0, 1 pulse rejection*[2]	t_{WFMDRJ}		100			ns
MODE0, 1, 2 input high/low level width*[1]	t_{WMDH}/ t_{WMDL}		600			ns
MODE0, 1, 2 pulse rejection*[2]	t_{WMDRJ}		100			ns

Note 1. FLMD0,1 and MODE0, 1, 2 input width is needed to ensure that the internal mode signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.





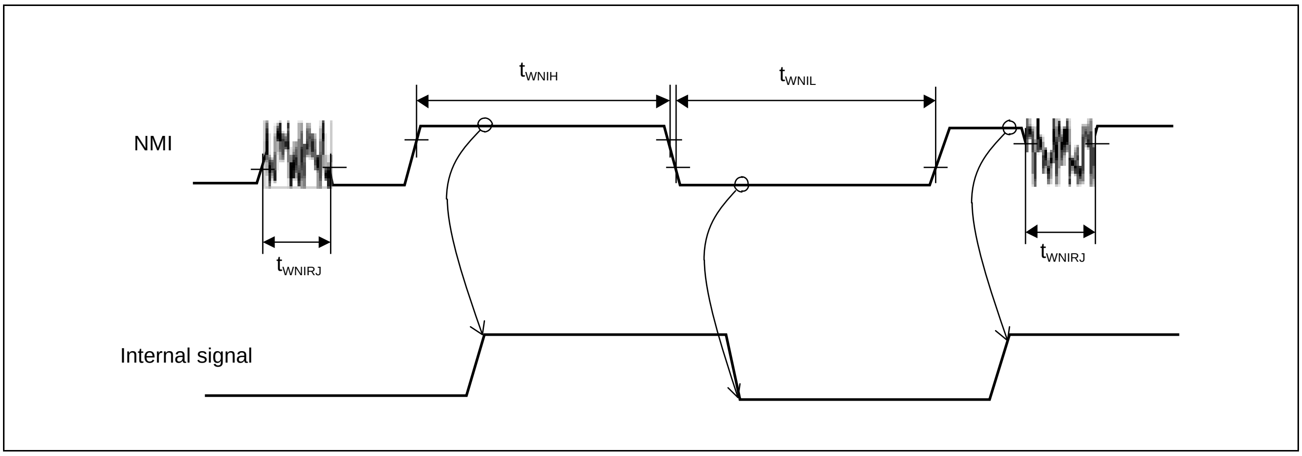
47C.5.3 Interrupt Timing

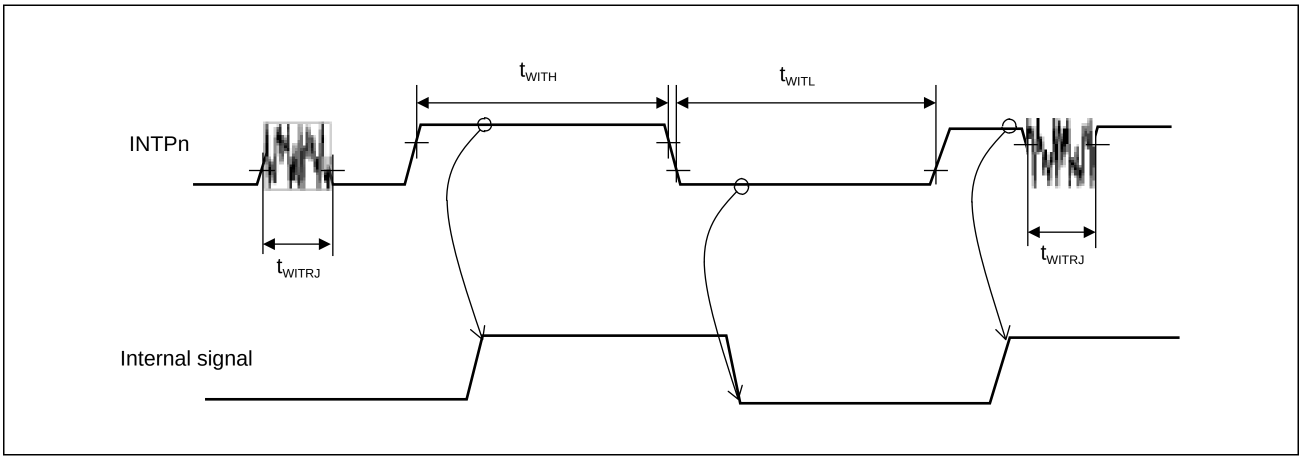
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
NMI input high/low level width*[1]	t_{WNIH}/ t_{WNIL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			μs
NMI pulse rejection*[2]	t_{WNIRJ}		100			ns
INTPn input high/low level width*[1]	t_{WITH}/ t_{WITL}	Edge detection mode	600			ns
		Level detection mode (EMCLK is operated by HS IntOSC)	756			ns
		Level detection mode (EMCLK is operated by LS IntOSC)	24			μs
INTPn pulse rejection*[2]	t_{WITRJ}		100			ns

Note 1. NMI and INTPn input width is needed to ensure that the internal interrupt signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.



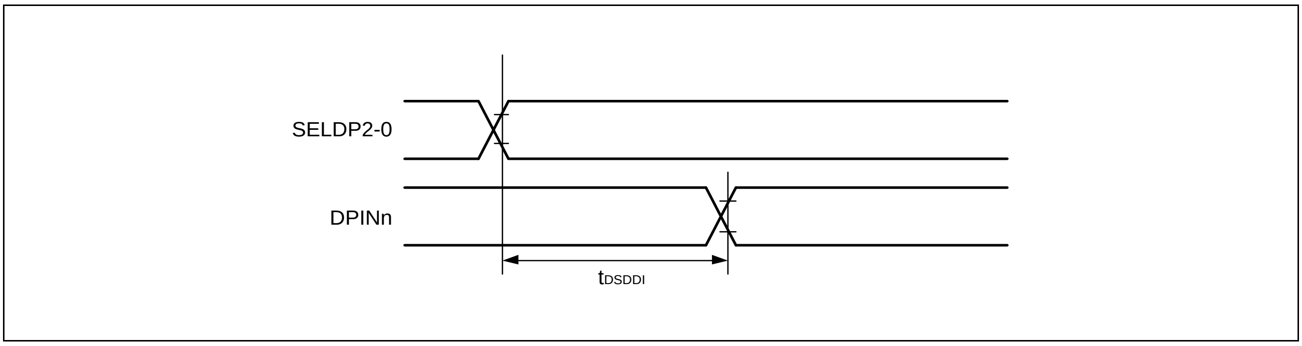


47C.5.4 Low Power Sampler (DPIN input) Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DPINn input delay time (vs SELDP2-0)	t_{DSDDI}				150	ns

Note: n = 7 to 0



47C.5.5 CSCXFOUT Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

<Output driver strength>

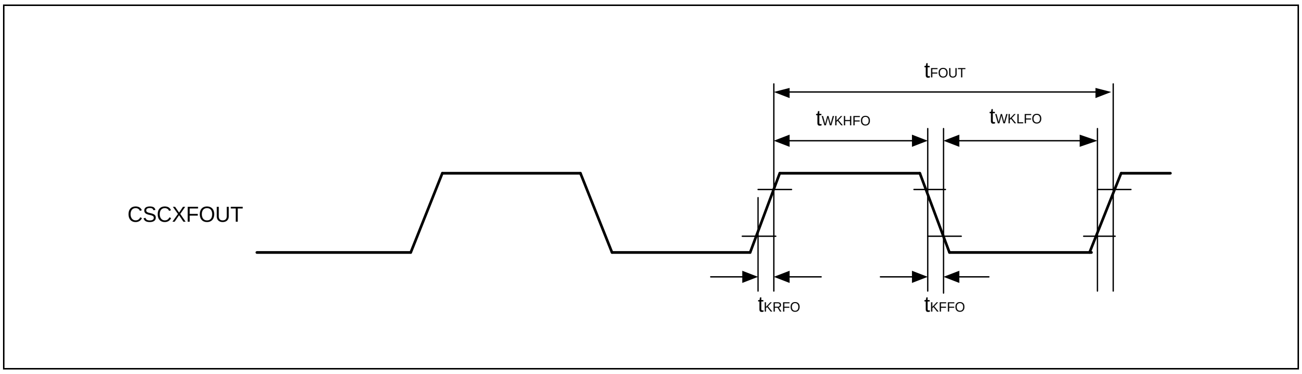
CSCXFOUT: Slow or fast mode (refer to the condition in the following table)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
CSCXFOUT output cycle	t_{FOUT}	Slow mode		100 (max. 10 MHz)			ns
		Fast mode		41.6 (max. 24 MHz)			ns
CSCXFOUT high level width	t_{WKHFO}	Slow mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)[*2, *3]	t_{FOUT} × (N+1) / 2N – 37			ns
		Fast mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)[*2]	t_{FOUT} × (N+1) / 2N – 10			ns
CSCXFOUT low level width	t_{WKLFO}	Slow mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 37			ns
			N: Odd value (N ≥ 5)[*2, *3]	t_{FOUT} × (N–1) / 2N – 37			ns
		Fast mode	N: 1[*1] or even value[*2]	t_{FOUT} / 2 – 10			ns
			N: Odd value (N ≥ 3)[*2]	t_{FOUT} × (N–1) / 2N – 10			ns
CSCXFOUT rise/ fall time	t_{KRFO}/ t_{KFFO}	Slow mode				37	ns
		Fast mode				10	ns

Note 1. When MainOSC, HS IntOSC, or LS IntOSC is selected as source clock with the condition of N = 1, the characteristics of output signal depends on the selected source clock. It is recommended to use output signal after evaluation on an actual environment.

Note 2. “N” is the value of “Clock divisor N” defined by FOUTDIV register.

Note 3. The selection of N = 3 is prohibited when slow mode is used.



47C.5.6 Reserved

47C.5.7 Reserved

47C.5.8 Reserved

47C.5.9 Reserved

47C.5.10 CSI Timing

47C.5.10.1 CSIG Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.12 CSIG Timing (Master Mode)

<Output driver strength>
CSIGnSO, CSIGnSC (output): Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYMGn}		100			ns
CSIGnSC high level width	t_{KWHMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSC low level width	t_{KWLMGn}		0.5 × t_{KCYMGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSIMGn}		30			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSIMGn}		0			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOMGn}				7	ns
CSIGnRYI setup time (vs. CSIGnSC)	t_{SRYIGn}	CSIGnCTL1.CSIGnSIT = x CSIGnCTL1.CSIGnHSE = 1	2 × t_{KCYGn} + 25			ns
CSIGnRYI high level width	t_{WRYIGn}	CSIGnCTL1.CSIGnHSE = 1	t_{KCYGn} + 5			ns

Note: n = 0

Table 47C.13 CSIG Timing (Slave Mode)

<Output driver strength>
CSIGnSO: Fast mode
CSIGnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro operation clock cycle time	t_{KCYGn}		12.5 (max. 80 MHz)			ns
CSIGnSC cycle time	t_{KCYSGn}		200			ns
CSIGnSC high level width	t_{KWHSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSC low level width	t_{KWLSGn}		0.5 × t_{KCYSGn} – 10			ns
CSIGnSI setup time (vs. CSIGnSC)	t_{SSISGn}		20			ns
CSIGnSI hold time (vs. CSIGnSC)	t_{HSISGn}		t_{KCYGn} + 5			ns
CSIGnSO output delay (vs. CSIGnSC)	t_{DSOSGn}				30	ns
CSIGnRYO output delay	t_{SRYOGn}				38	ns
$\overline{\text{CSIGnSSI}}$ setup time (vs. CSIGnSC)	$t_{SSSISGn}$		0.5 × t_{KCYSGn} – 5			ns
$\overline{\text{CSIGnSSI}}$ hold time (vs. CSIGnSC)	$t_{HSSISGn}$		t_{KCYGn} + 5			ns

Note: n = 0

47C.5.10.2 CSIH Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.14 CSIH Timing (Master Mode: 10 Mbps)

<Output driver strength>

CSIHnSO, CSIHnSC (output): Fast mode (CL = 100pF@n=0 / 50pF@n=1-3)

CSIHnCSSx: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYMHn}		100			ns
CSIHnSC high level width	t_{KWHMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSC low level width	t_{KWLMHn}		0.5 × t_{KCYMHn} – 10			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	19			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	14			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSIMHn}	SI positive edge mode (CSIHnCTL1.CSIHnSLRS = 0)	0			ns
		SI negative edge mode (CSIHnCTL1.CSIHnSLRS = 1)	t_{KCYHn}/2			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOMHn}				7	ns
CSIHnRYI setup time (vs. CSIHnSC)	t_{SRYIHn}	CSIHnCTL1.CSIHnSIT = x CSIHnCTL1.CSIHnHSE = 1	2 × t_{KCYHn} + 25			ns
CSIHnRYI high level width	t_{WRYIHn}	CSIHnCTL1.CSIHnHSE = 1	t_{KCYHn} + 5			ns
CSIHnCSS0-7 inactive width	$t_{WSCSBHn}$		CSIDLE × t_{KCYMHn} – 15			ns
CSIHnCSS0-7 setup time (vs. CSIHnSC)	$t_{SSCSBHn0}$	CSIHnCFGx.CSIHnDAP = 0	CSSETUP × t_{KCYMHn} – 23			ns
	$t_{SSCSBHn1}$	CSIHnCFGx.CSIHnDAP = 1	(CSSETUP + 0.5) × t_{KCYMHn} – 23			ns
CSIHnCSS0-7 hold time (vs. CSIHnSC)	$t_{HSCSBHn0}$	CSIHnCTL1.CSIHnSIT = 0	CSSHOLD × t_{KCYMHn} – 5			ns
	$t_{HSCSBHn1}$	CSIHnCTL1.CSIHnSIT = 1	(CSSHOLD + 0.5) × t_{KCYMHn} – 5			ns

Note: n = 0 to 3

NOTE

CSIDLE: Setting value of CSIHnCFGx.CSIHnIDx[2:0]
CSSETUP: Setting value of CSIHnCFGx.CSIHnSPx[3:0]
CSSHOLD: Setting value of CSIHnCFGx.CSIHnHDx[3:0]
x: Depends on number of the chip select signals.

CAUTION

When the serial clock level is changed during the communication (CSIHnCFGx.CSIHnCKPx) and the IDLE has a setting of 0.5 transmission clock cycles, an inactive width time $t_{WSCSBHn}$ of “0.5 × t_{KCYMHn}” is added.

Table 47C.15 CSIH Timing (Slave Mode: 5 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		200			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		20			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				30	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			38	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$38 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 0 to 3

Table 47C.16 CSIH Timing (Slave Mode: 8 Mbps)

<Output driver strength>
CSIHnSO: Fast mode
CSIHnRYO: Slow mode

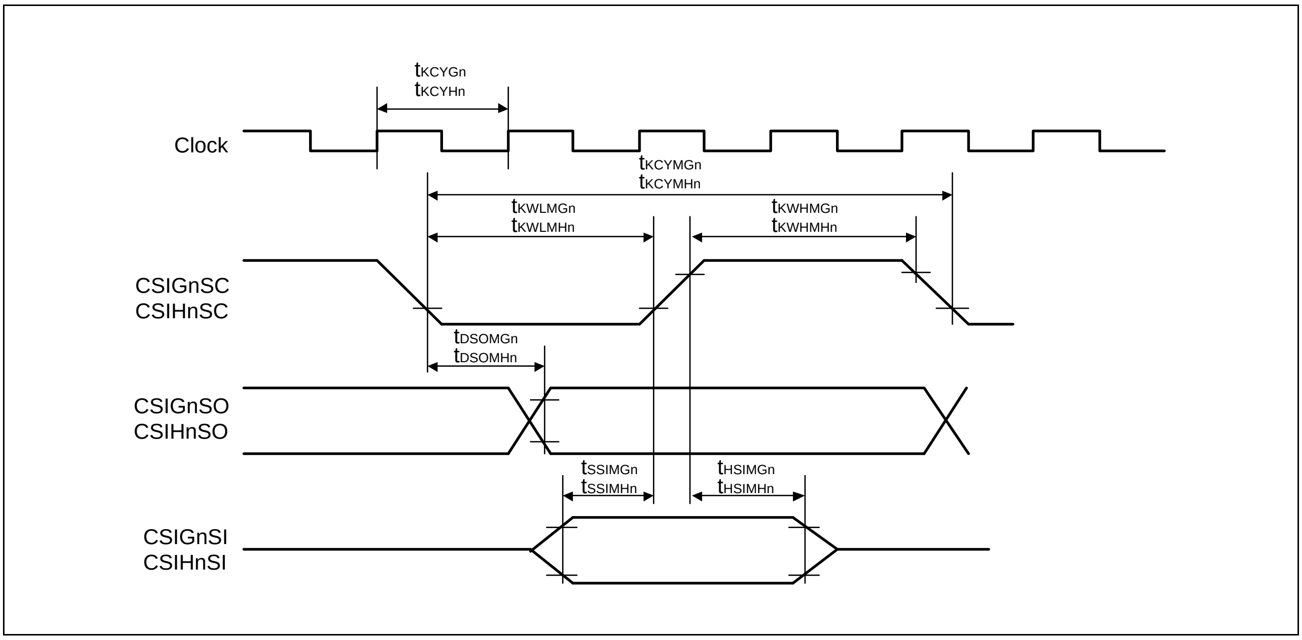
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Macro Operation clock cycle time	t_{KCYHn}		12.5 (max. 80 MHz)			ns
CSIHnSC cycle time	t_{KCYSHn}		125			ns
CSIHnSC high level width	t_{KWHSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSC low level width	t_{KWLSHn}		$0.5 \times t_{KCYSHn} - 10$			ns
CSIHnSI setup time (vs. CSIHnSC)	t_{SSISHn}		12.5			ns
CSIHnSI hold time (vs. CSIHnSC)	t_{HSISHn}		$t_{KCYHn} + 5$			ns
CSIHnSO output delay (vs. CSIHnSC)	t_{DSOSHn}				25	ns
CSIHnRYO output delay	t_{SRYOHn}	$t_{KCYSHn} \geq 8 \times t_{KCYHn}$			27	ns
		$t_{KCYSHn} < 8 \times t_{KCYHn}$			$27 + t_{KCYHn}$	ns
$\overline{\text{CSIHnSSI}}$ setup time (vs. CSIHnSC)	$t_{SSSISHn}$		$0.5 \times t_{KCYSHn} - 5$			ns
$\overline{\text{CSIHnSSI}}$ hold time (vs. CSIHnSC)	$t_{HSSISHn}$		$t_{KCYHn} + 5$			ns

Note: n = 2 (80/100-pin versions), n = 0 (48/64-pin versions)

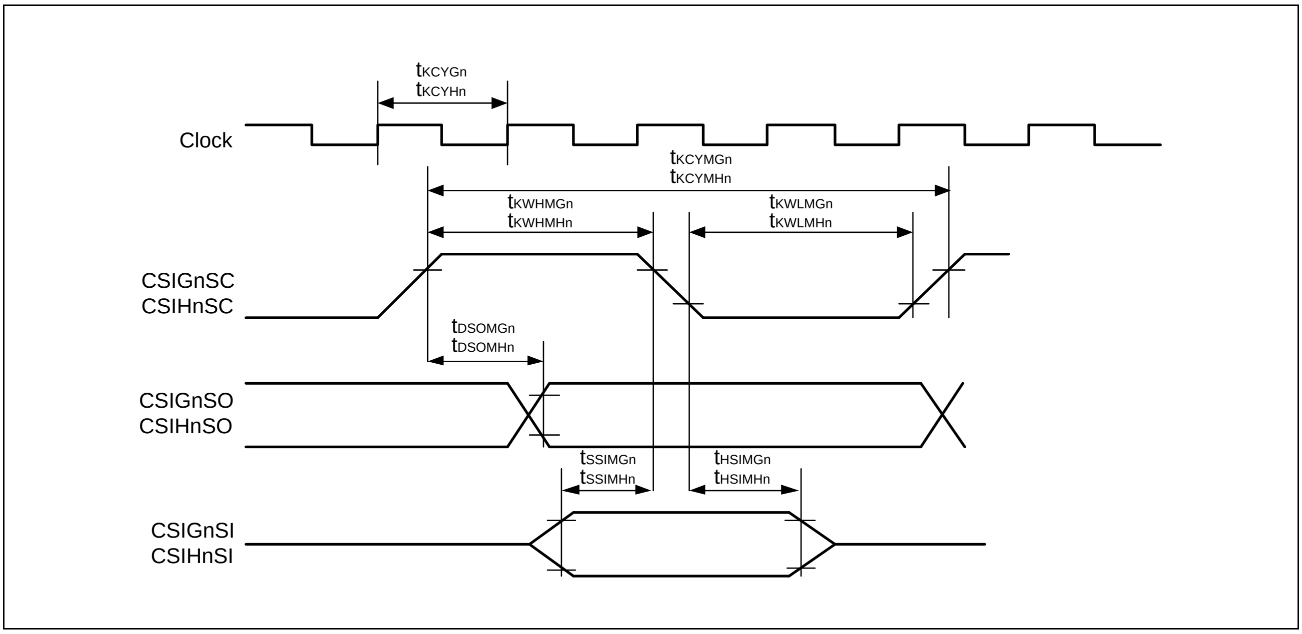
(1) SC/SI/SO

Master mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

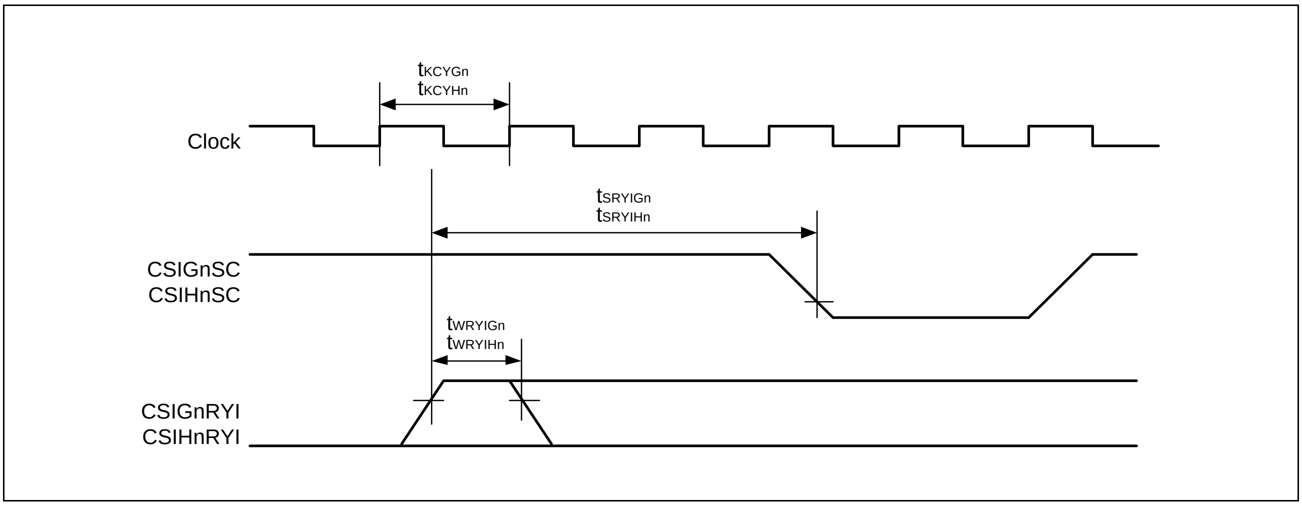


(2) RYI

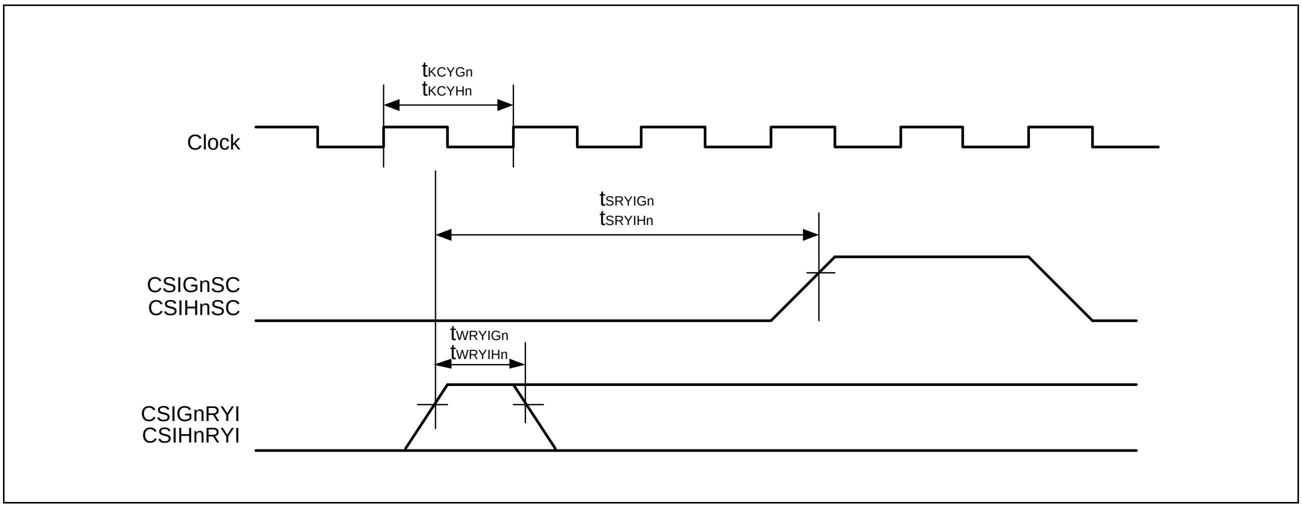
Master mode:

- CSIG: Only master mode (CSIGnCTL1: CSIGnHSE = 1, CSIGnCTL1: CSIGnSIT = 0)
- CSIH: Only master mode (CSIHnCTL1: CSIHnHSE = 1, CSIHnCTL1: CSIHnSIT = 0)

  - CSIG (CSIGnCTL1: CSIGnCKR = 0)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 0)



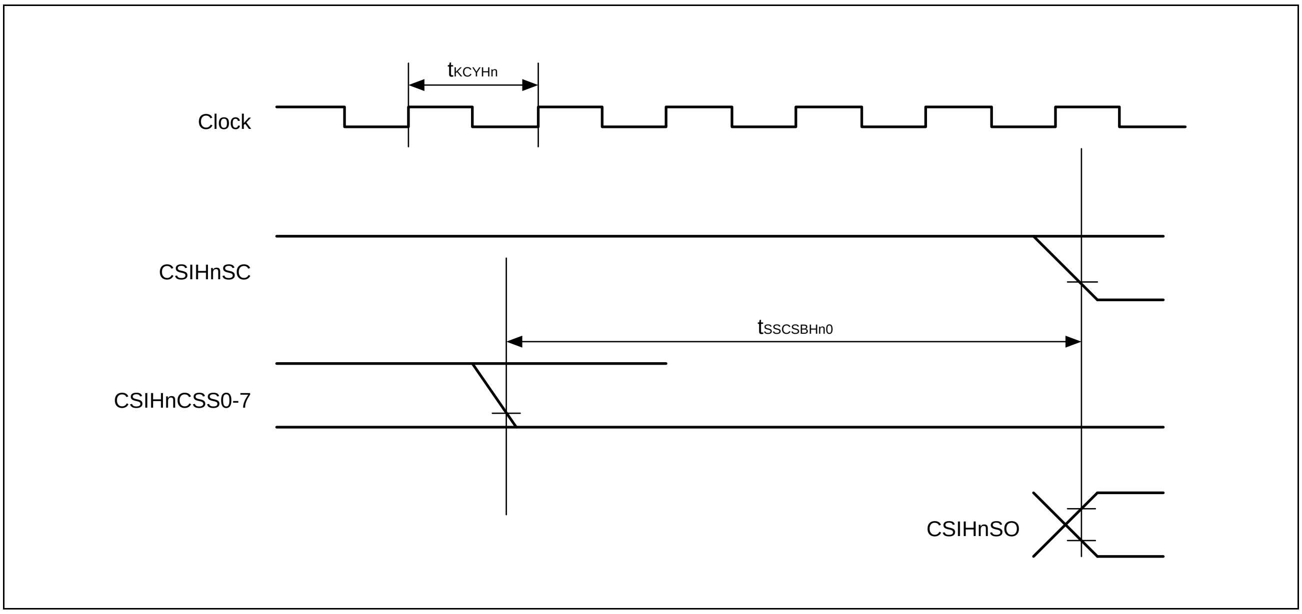
  - CSIG (CSIGnCTL1: CSIGnCKR = 1)
  - CSIH (CSIHnCFGx: CSIHnCKPx = 1)



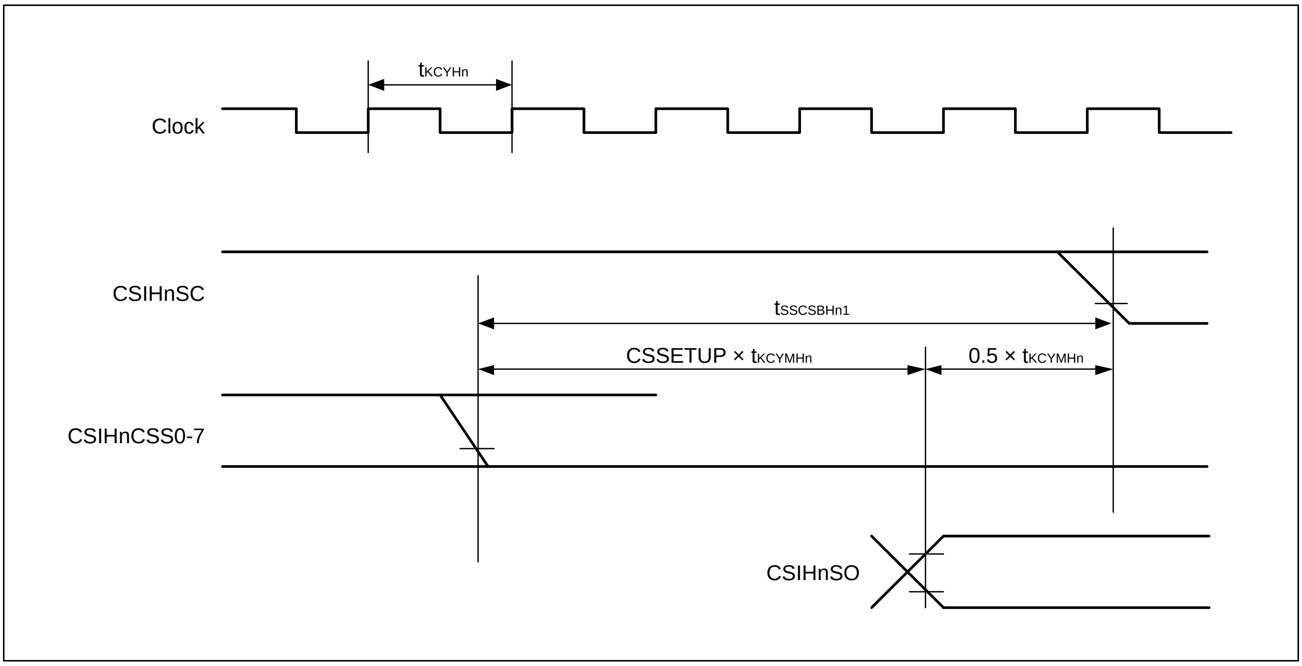
(3) CSSx

Only master mode (setup time):

- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0

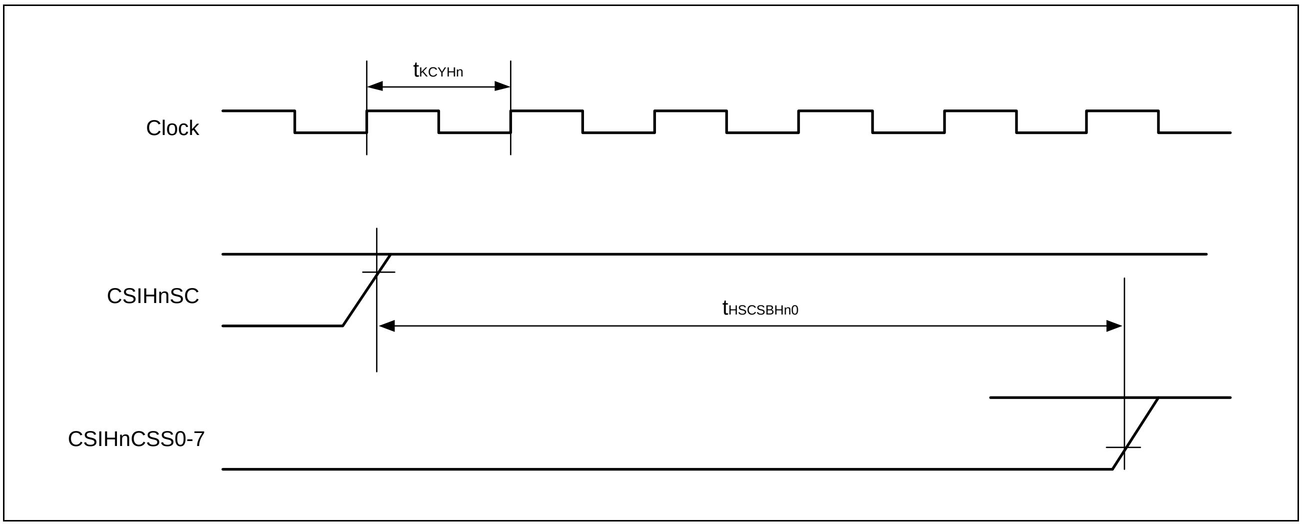


- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 1

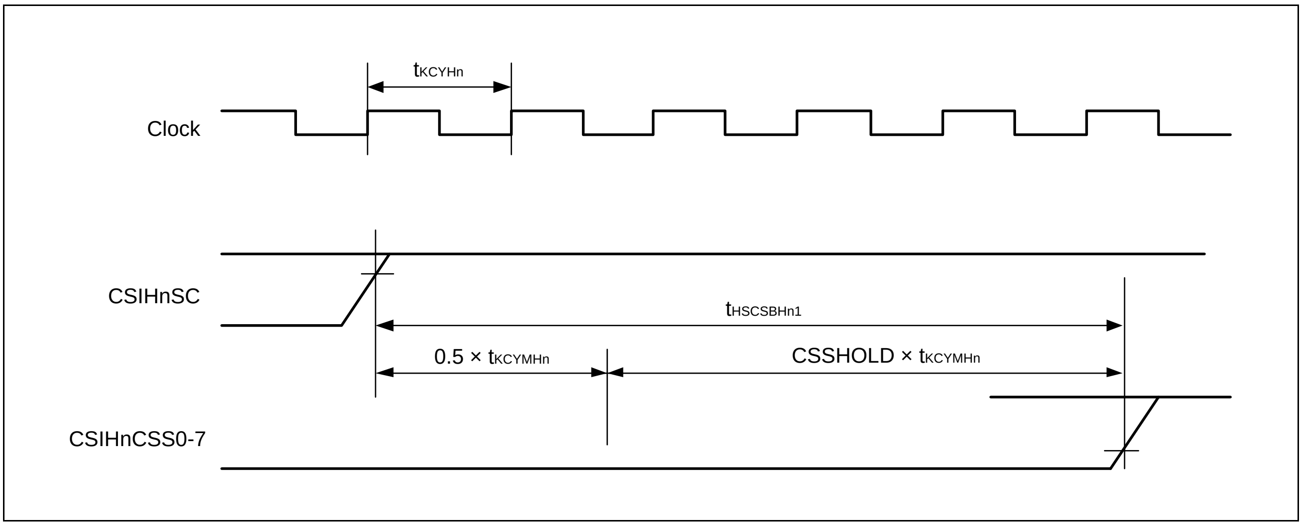


Only master mode (hold time):

- CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



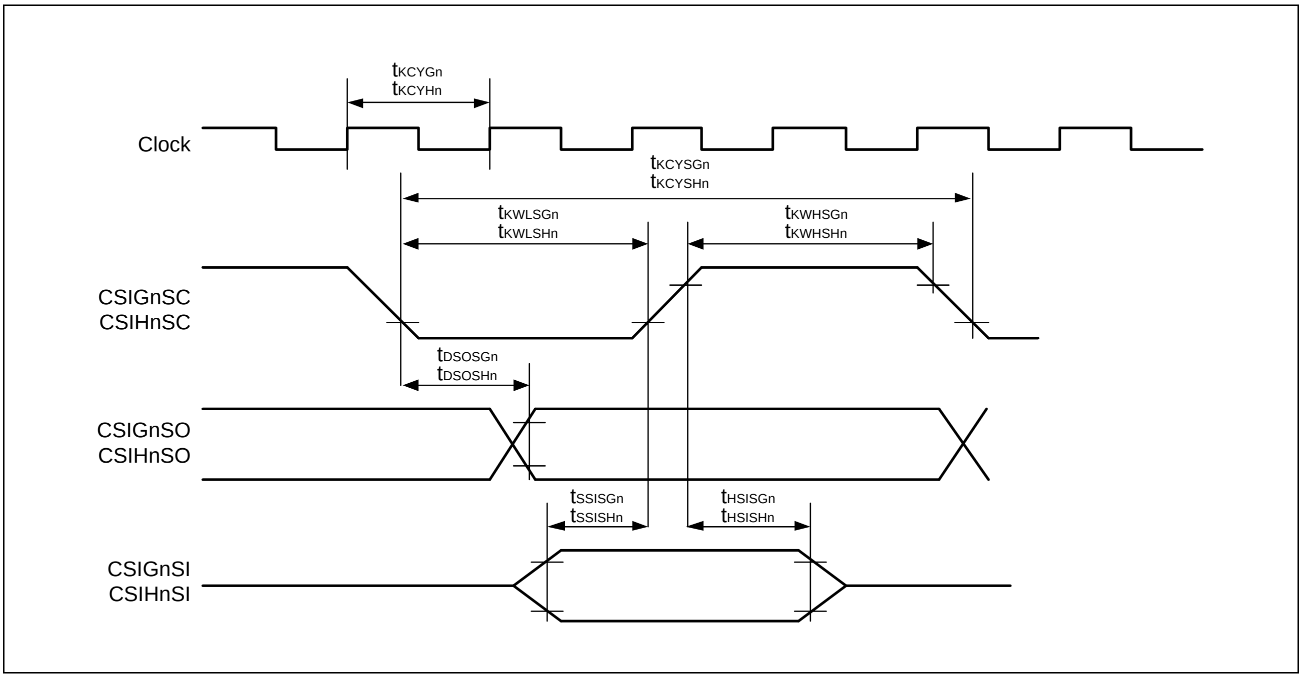
- CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



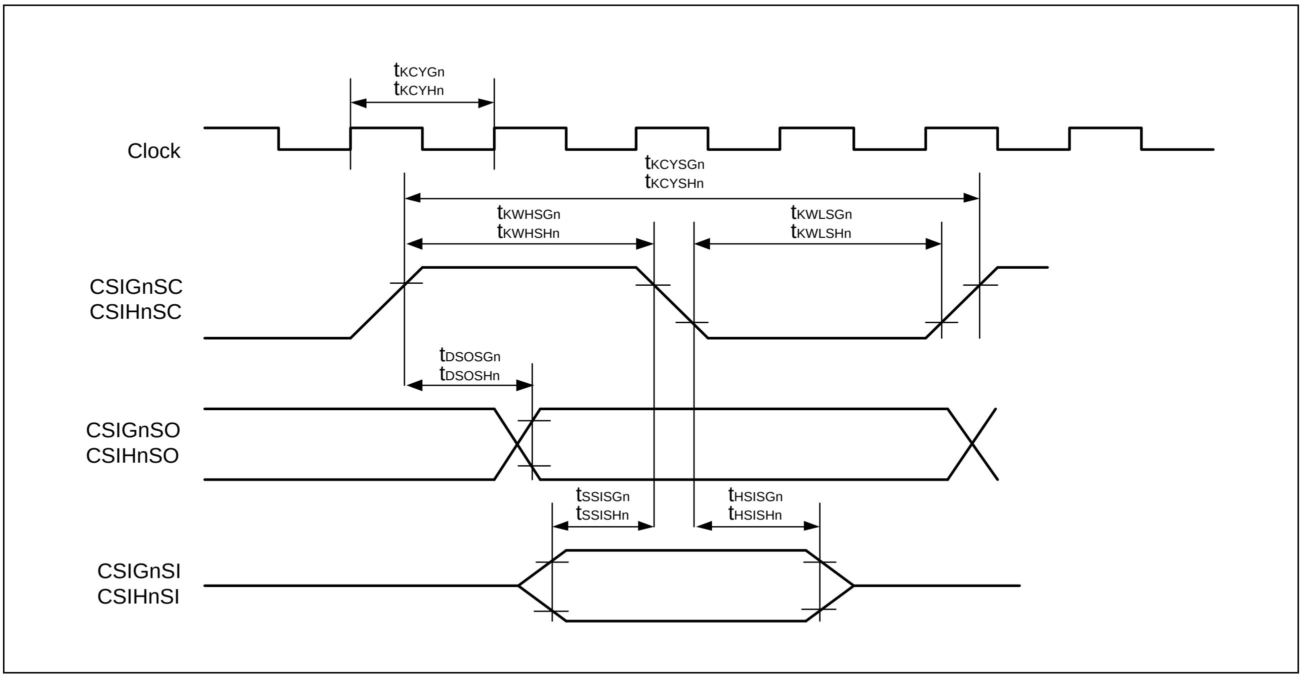
(4) SC/SI/SO

Slave mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)

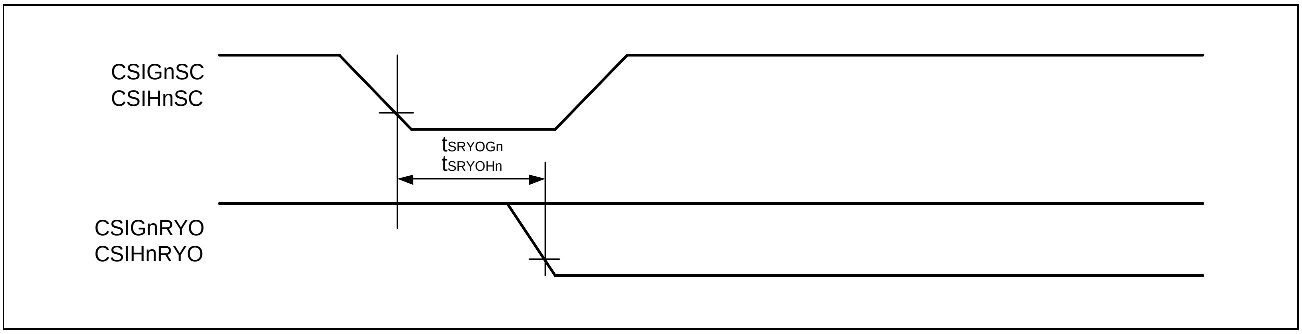


- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

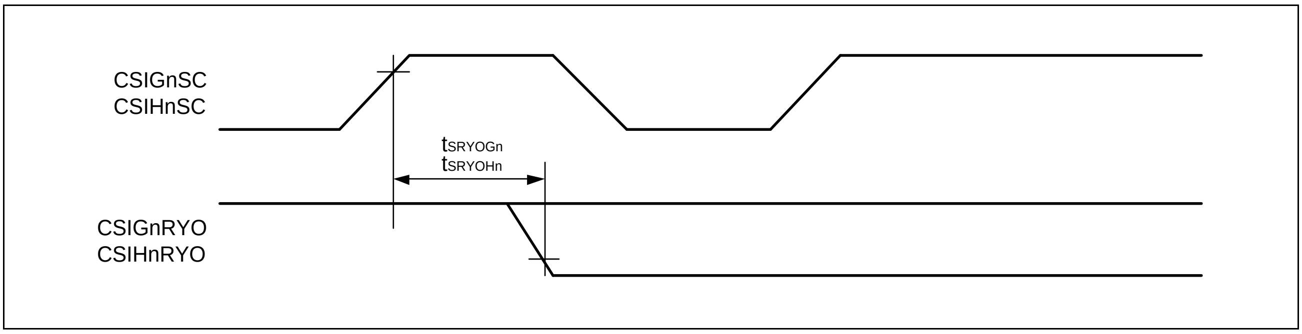


(5) RYO

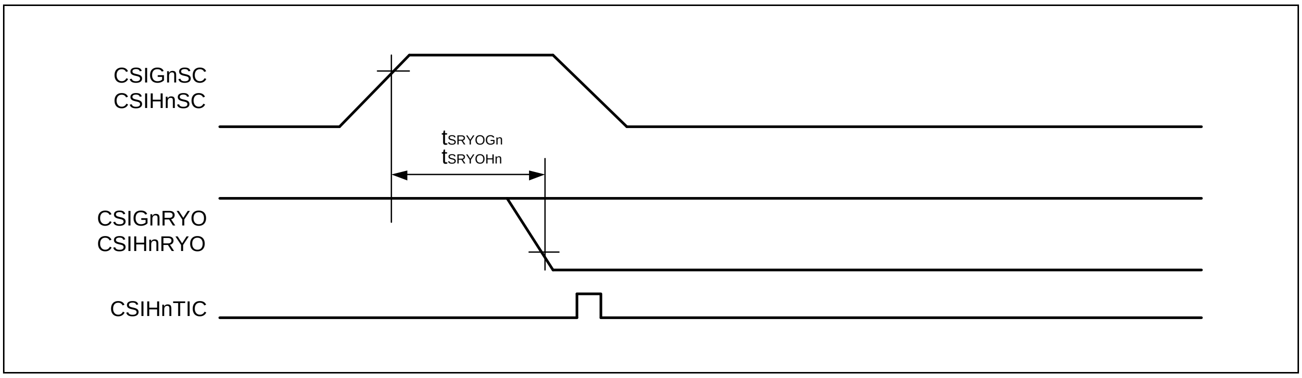
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0)



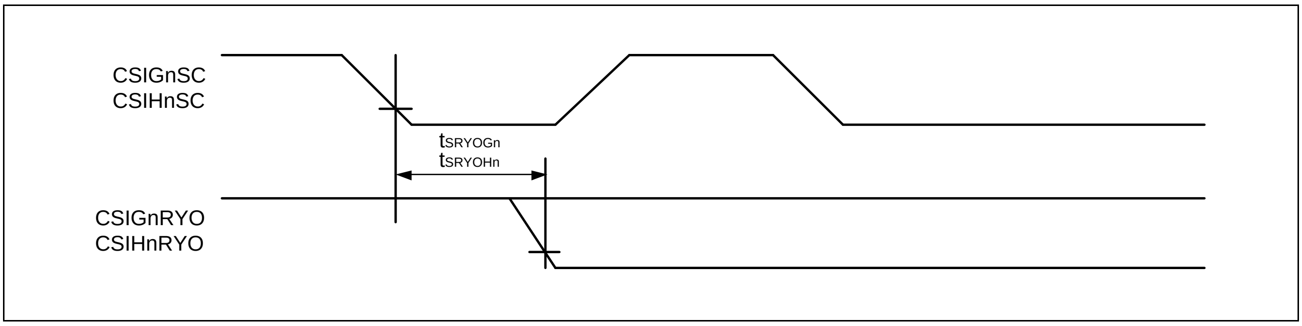
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/1)



- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0)



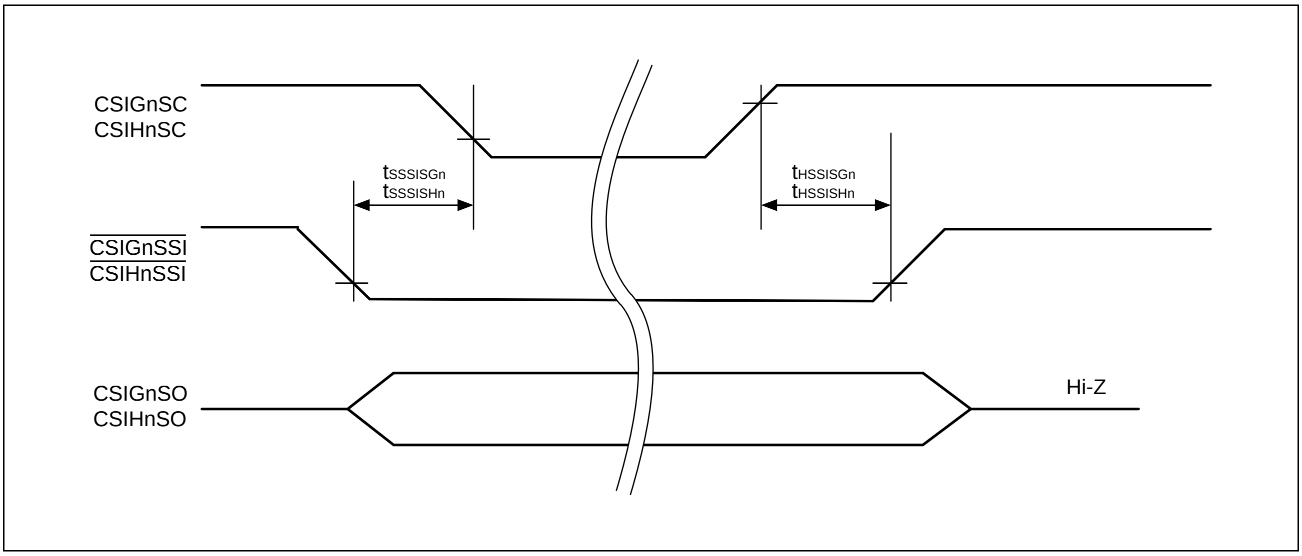
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/1)



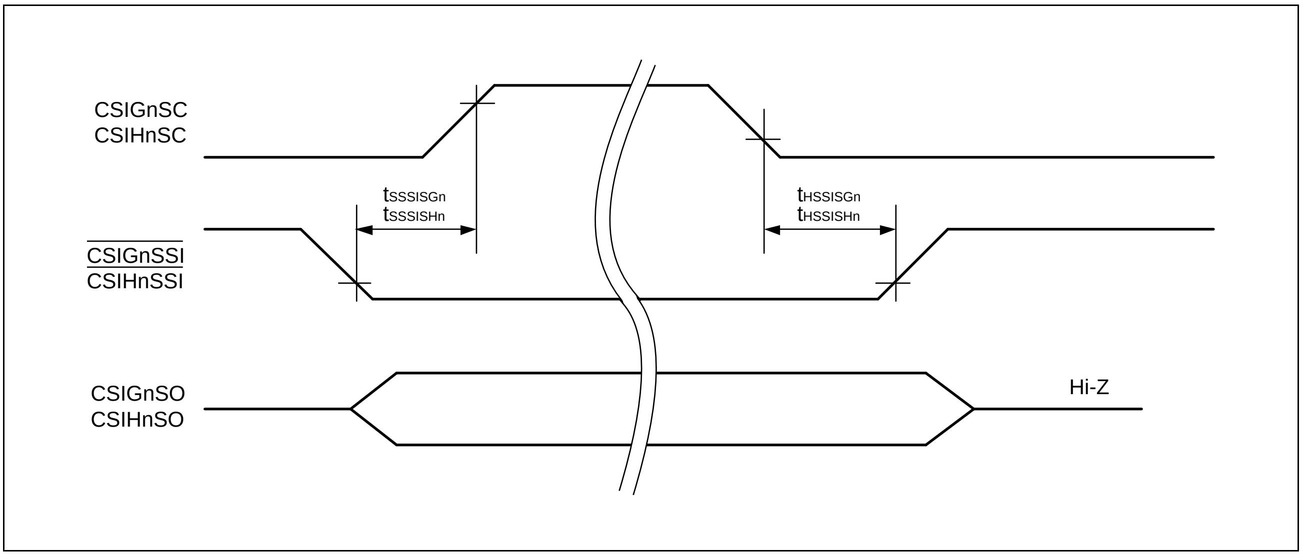
(6) SSI

Slave mode:

- CSIG (CSIGnCTL1: CSIGnSSE=1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0 or 1/1)
- CSIH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSIG (CSIGnCTL1: CSIGnSSE=1, CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 1/0 or 0/1)
- CSIH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)



47C.5.11 RLIN2/RLIN3 Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RLIN3 transfer rate		LIN specification	1		20	kbps
		LIN extended baud rate	1		115.2*[1]	kbps
		UART function			1.5	Mbps
RLIN2 transfer rate		LIN specification	1		20	kbps

Note 1. The LIN extended baud rate is not part of the LIN standard specification.

47C.5.12 RIIC Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C

Table 47C.17 RIIC Timing (Normal Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				100	kHz
Bus free time (between stop/start condition)	t_{BUF}		4.7			µs
Hold time*[1]	t_{HD}: STA		4.0			µs
RIICnSCL clock low-level width	t_{LOW}		4.7			µs
RIICnSCL clock high-level time	t_{HIGH}		4.0			µs
Setup time for start/restart condition	t_{SU}: STA		4.7			µs
Data hold time	t_{HD}: DAT	CBUS compatible master	5.0			µs
		I^{2}C mode	0*[2]			µs
Data setup time	t_{SU}: DAT		250			ns
Stop condition setup time	t_{SU}: STO		4.0			µs
Capacitance load of each bus line	Cb				400	pF

Remark: n = 0, 1

Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Table 47C.18 RIIC Timing (Fast Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				400	kHz
Bus free time (between stop/start condition)	t_{BUF}		1.3			μs
Hold time[*1]	t_{HD}: STA		0.6			μs
RIICnSCL clock low-level width	t_{LOW}		1.3			μs
RIICnSCL clock high-level time	t_{HIGH}		0.6			μs
Setup time for start/restart condition	t_{SU}: STA		0.6			μs
Data hold time	t_{HD}: DAT	I^{2}C mode	0[*2]			μs
Data setup time	t_{SU}: DAT		100[*3]			ns
Stop condition setup time	t_{SU}: STO		0.6			μs
Pulse width with spike suppressed by input filter	t_{SP}		0		50	ns
Capacitance load of each bus line	Cb				400	pF

Remark: n = 0, 1

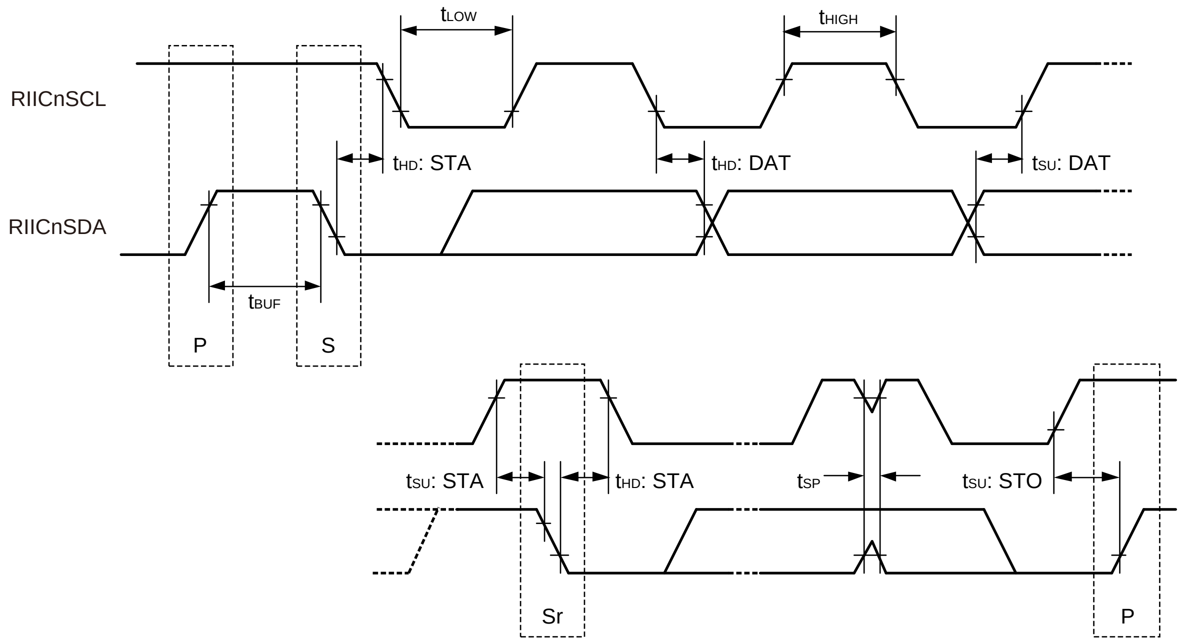
Note: If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL

Note 3. The fast mode I^{2}C bus can be used in normal mode I^{2}C bus system. In this case, set the fast mode I^{2}C bus so that it meets the following conditions.

- If the system does not extend the RIICnSCL signal's low state hold time: t_{SU}: DAT ≥ 250 ns
- If the system extends the RIICnSCL signal's low state hold time:
  Transmit the following data bit to the RIICnSDA line prior to releasing the RIICnSCL line
  (1250 ns: Normal mode I^{2}C bus specification).



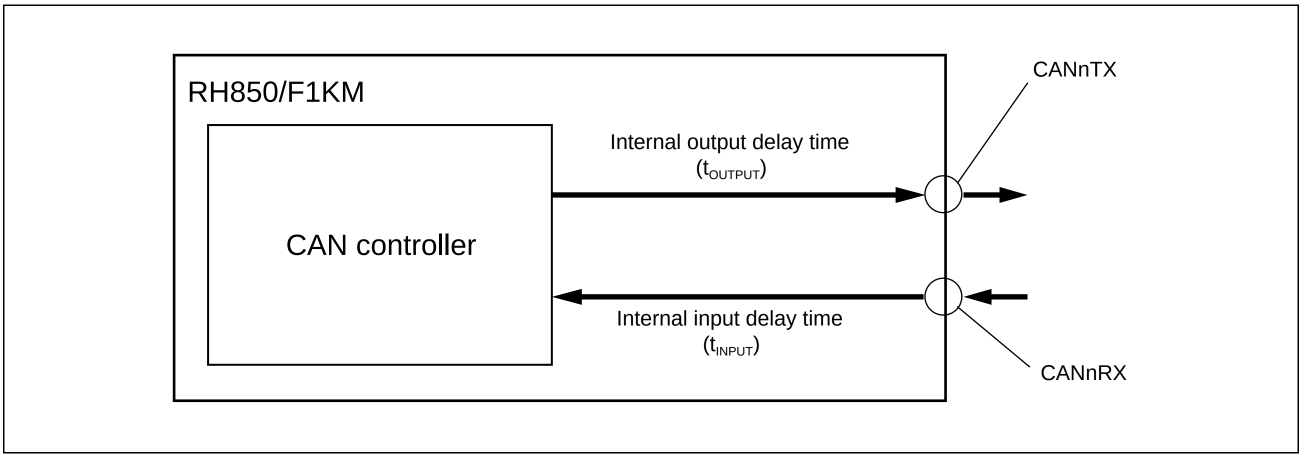
Note: P: Stop condition S: Start condition Sr: Restart condition

47C.5.13 RS-CANFD Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate		Classical CAN mode			1	Mbps
Data bit rate (CAN FD mode)		Nominal bit rate ≤ 500 kbps			5	Mbps
		Nominal bit rate > 500 kbps			2	Mbps
Internal delay time*[1]	t_{NODE}				50	ns

Note 1. t_{NODE} = Internal input delay time (t_{INPUT}) + Internal output delay time (t_{OUTPUT})



47C.5.14 Reserved

47C.5.15 Reserved

47C.5.16 RSENT Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1μF ±30%, CISOVCL: 0.1μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Tick Time			1		90	μs

47C.5.17 Timer Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

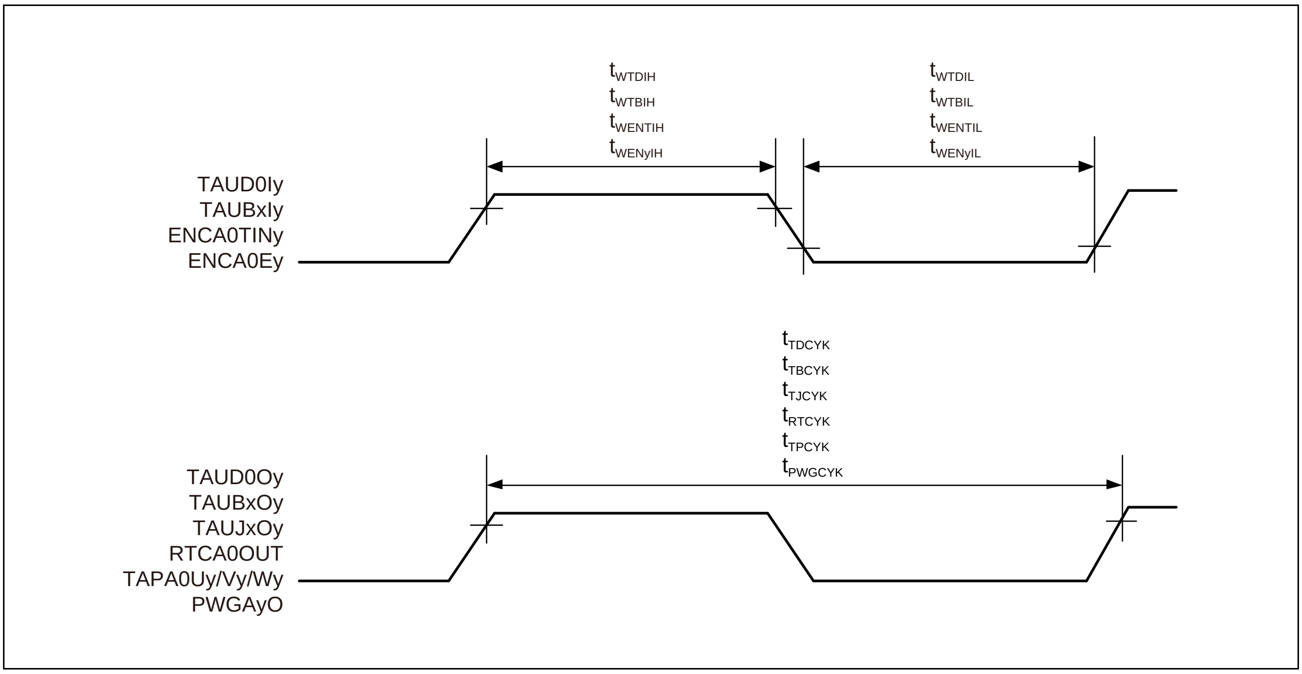
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TAUD0Iy input high/low level width (y = 0 to 15)	t_{WTDIH}/ t_{WTDIL}		n × Tsamp + 20[*1, *2]			ns
TAUD0Oy output cycle (y = 0 to 15)	t_{TDCYK}	Slow mode			10	MHz
TAUBxIy input high/low level width (x = 0, y = 0 to 15)	t_{WTBIH}/ t_{WTBIL}		n × Tsamp + 20[*1, *2]			ns
TAUBxOy output cycle (x = 0, y = 0 to 15)	t_{TBCYK}	Slow mode			10	MHz
TAUJxIy input high/low level width[*3] (x = 0 to 3, y = 0 to 3)	t_{WTJIH}/ t_{WTJIL}		600			ns
TAUJxIy pulse rejection[*4]	t_{WTIJRJ}		100			ns
TAUJxOy output cycle (x = 0 to 3, y = 0 to 3)	t_{TJCYK}	Slow mode			10	MHz
RTCA0OUT output cycle	t_{RTCYK}			1		Hz
TAPA0ESO input high/low level width[*3]	t_{WESIH}/ t_{WESIL}		600			ns
TAPA0ESO pulse rejection[*4]	t_{WESIRJ}		100			ns
TAPA0Uy/Vy/Wy output cycle (y = P, N)	t_{TPCYK}	Slow mode			10	MHz
ENCA0TINy input high/low level width (y = 0, 1)	t_{WENTIH}/ t_{WENTIL}		n × Tsamp + 20[*1]			ns
ENCA0Ey input high/low level width (y = 0,1,C)	t_{WENyIH}/ t_{WENyIL}		n × Tsamp + 20[*1]			ns
PWGAyO output cycle (y = 0 to 47)	t_{PWGCYK}	Slow mode			10	MHz

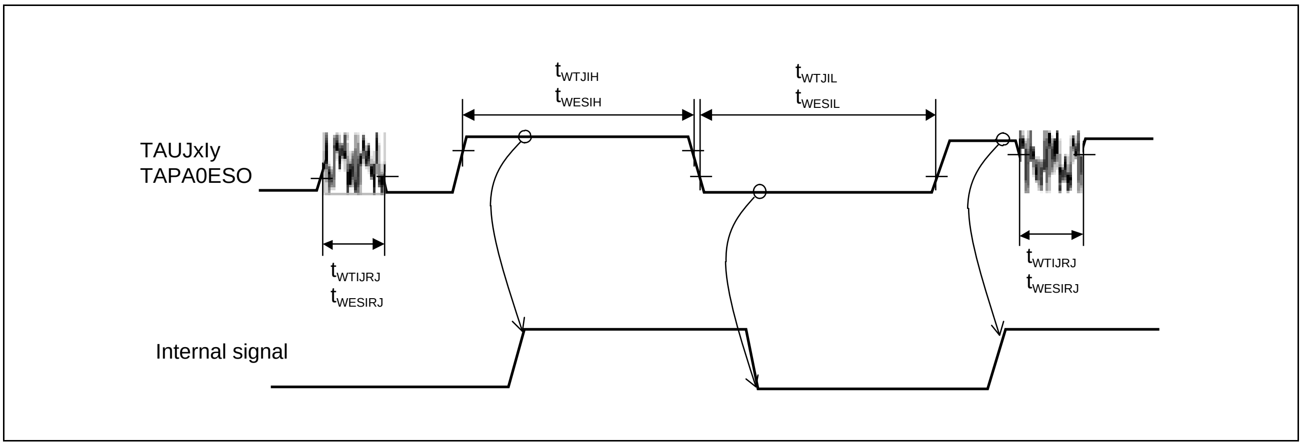
Note 1. n: Sampling number of the digital noise filter for each input. Tsamp: Sampling time of the digital noise filter for each input.

Note 2. Input more than 1 count clock width of each timer counter channel.

Note 3. TAUJxIy and TAPA0ESO input width is needed to ensure that the internal timer input signal is activated.

Note 4. Pulses shorter than this minimum is ignored. This is reference value. Noise such as the figure can be filtered.


tWTDIH
tWTBIH
tWENTIH
tWENyIH
tWTDIL
tWTBIL
tWENTIL
tWENyIL
TAUD0Iy
TAUBxIy
ENCA0TINy
ENCA0Ey
tTDCYK
tTBCYK
tTJCYK
tRTCYK
tTPCYK
tPWGCYK
TAUD0Oy
TAUBxOy
TAUJxOy
RTCA0OUT
TAPA0Uy/Vy/Wy
PWGAyO

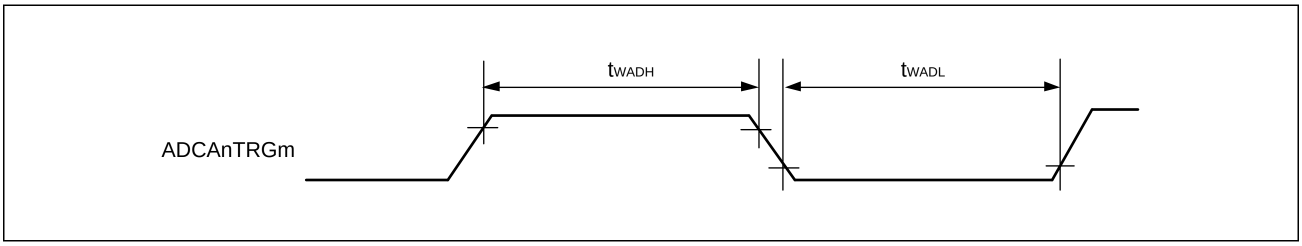

tWTJIH
tWESIH
tWTJIL
tWESIL
TAUJxIy
TAPA0ESO
tWTIJRJ
tWESIRJ
tWTIJRJ
tWESIRJ
Internal signal

47C.5.18 ADTRG Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ADCAnTRGm input high/ low level width	t_{WADH}/t_{WADL}		k × Tsamp + 20*[1]			ns

Note 1. k: Sampling number of the digital noise filter for each input.
Tsamp: Sampling time of the digital noise filter for each input.



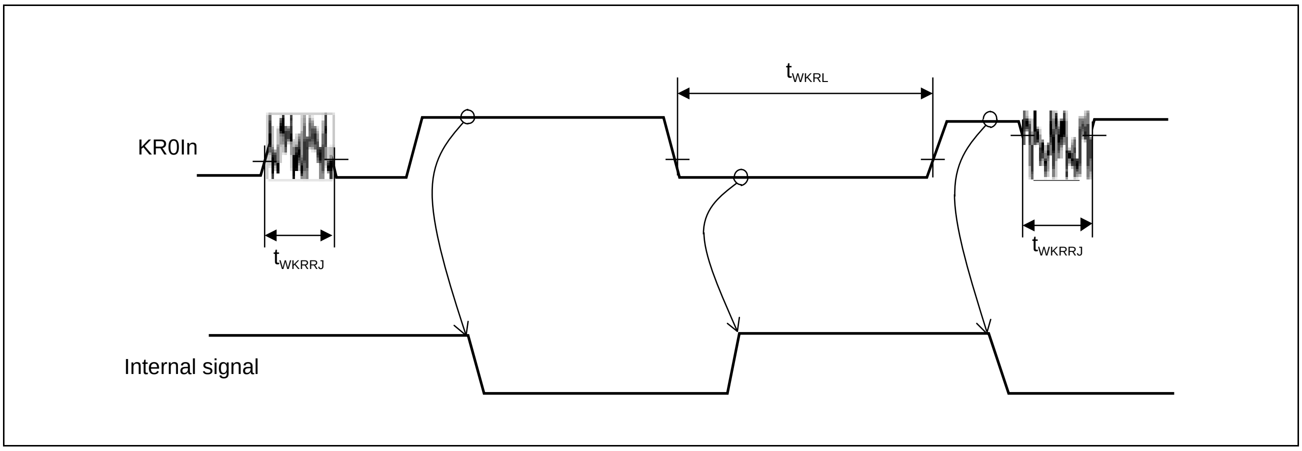
47C.5.19 Key Return Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
KR0In input low level width*[1]	t_{WKRL}		600			ns
KR0In pulse rejection*[2]	t_{WKRRJ}		100			ns

Note 1. KR0In input width is needed to ensure that the internal key input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



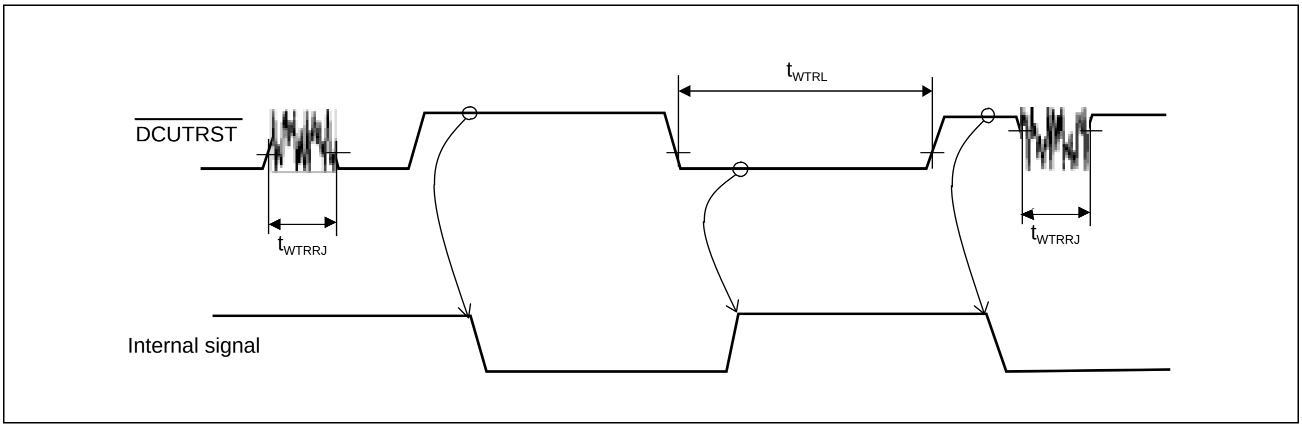
47C.5.20 $\overline{\text{DCUTRST}}$ Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{DCUTRST}}$ input low level width*[1]	t_{WTRL}		600			ns
$\overline{\text{DCUTRST}}$ pulse rejection*[2]	t_{WTRRJ}		100			ns

Note 1. $\overline{\text{DCUTRST}}$ input width is needed to ensure that the internal DCU reset input signal is activated.

Note 2. Pulses shorter than this minimum is ignored. This is reference value.
Noise such as the figure can be filtered.



47C.5.21 Debug Interface Characteristics

47C.5.21.1 Nexus Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

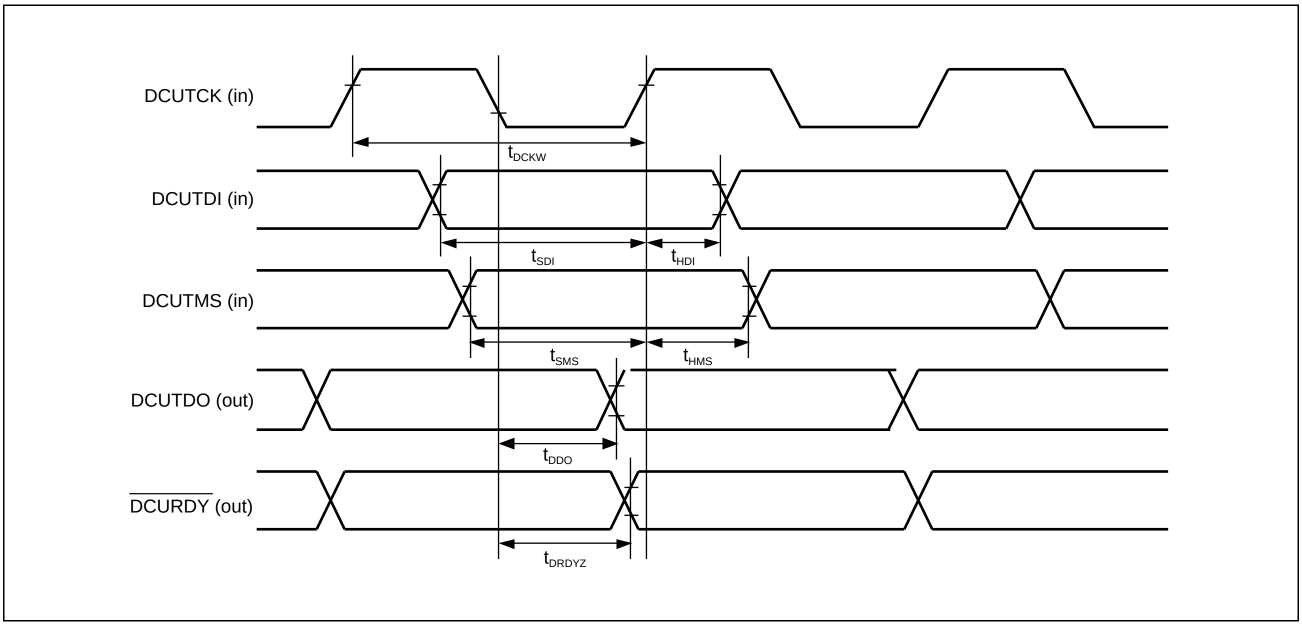
<Input buffer>

DCUTDI, DCUTCK, DCUTMS, $\overline{\text{DCUTRST}}$: TTL

<Output driver strength>

DCUTDO, $\overline{\text{DCURDY}}$: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DCUTCK cycle width	t_{DCKW}		50			ns
DCUTDI setup time (vs DCUTCK ↑)	t_{SDI}		12			ns
DCUTDI hold time (vs DCUTCK ↑)	t_{HDI}		3			ns
DCUTMS setup time (vs DCUTCK ↑)	t_{SMS}		12			ns
DCUTMS hold time (vs DCUTCK ↑)	t_{HMS}		3			ns
DCUTDO delay time (↓ DCUTCK)	t_{DDO}		0		20	ns
$\overline{\text{DCURDY}}$ delay time (↓ DCUTCK)	t_{RDYZ}		0		20	ns



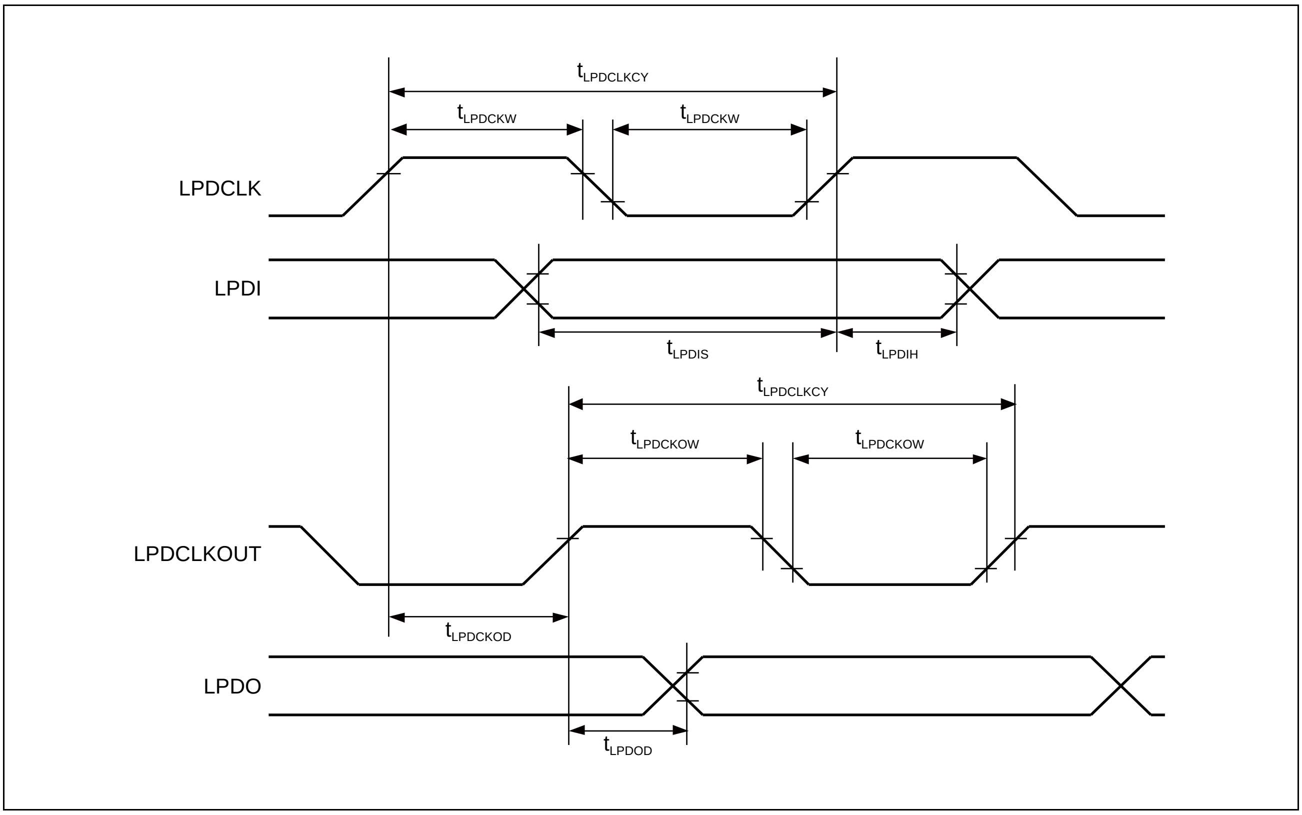
47C.5.21.2 LPD (4 Pins) Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1µF ±30%, CISOVCL: 0.1µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 100 pF

<Input buffer>
LPDCLK, LPDI: TTL

<Output driver strength>
LPDCLKOUT, LPDO: Fast mode

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPDCLK cycle time/ LPDCLKOUT cycle time	$t_{LPDCLKCY}$		83.3 (max.12 MHz)			ns
LPDCLK High-level width/ LPDCLK Low-level width	t_{LPDCKW}		$0.5 \times t_{LPDCLKCY} - 10$			ns
LPDCLKOUT High-level width/ LPDCLKOUT low-level width	$t_{LPDCKOW}$		$t_{LPDCKW} - 10$			ns
LPDI setup time (LPDCLK ↑)	t_{LPDIS}		41			ns
LPDI hold time (LPDCLK ↑)	t_{LPDIH}		3			ns
LPDCLK to LPDCLKOUT delay time	$t_{LPDCKOD}$				44	ns
LPDO delay time (LPDCLKOUT ↑)	t_{LPDOD}		0		15	ns



47C.5.21.3 LPD (1 Pin) Interface Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1µF ±30%, CISOVCL: 0.1µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 50 pF

<Input buffer>
LPDIO: TTL

<Output driver strength>
LPDIO: Fast mode

<External pull-up resistor>
LPDIO: 4.7 kΩ to 10 kΩ

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPD (1 pin) baud rate					2.0	Mbps

47C.6 A/D Converter Characteristics

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1µF ±30%, CISOVCL: 0.1µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

(1/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Conversion clock	ADCLKn				8[*3]		40	MHz
Resolution	RESn	12-bit mode			12	12	12	bit
		10-bit mode			10	10	10	bit
Conversion time	t_{CONn}	ADCAnSMPCR.SMPT[7:0] = 12 H(40 cycle) (8M Hz[*3] ≤ ADCLKn ≤ 32 MHz), External MPX is not used			1.25		5	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (46 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz), External MPX is not used			1.15		5.75	µs
		ADCAnSMPCR.SMPT[7:0] = 12 H (80 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 32 MHz), External MPX is used			2.5[*4]		10	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (92 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz), External MPX is used			2.3[*4]		11.5	µs
Sampling time	t_{SMP}	ADCAnSMPCR.SMPT[7:0] = 12 H (18 cycle) (8 MHz[*3] ≤ ADCLKn ≤ 32 MHz)			0.56		2.25	µs
		ADCAnSMPCR.SMPT[7:0] = 18 H (24cycle) (8 MHz[*3] ≤ ADCLKn ≤ 40 MHz)			0.6		3	µs
Overall error[*1]	TOEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±4.0	LSB
				ADCA0I0-5 (T&H used)			±6.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±6.0	LSB
				ADCA0I0-5 (T&H used)			±8.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Analog input voltage	VAIN0SN	ADCAnIm	T&H not used		AnVSS		AnVREF	V
		ADCA0I0-5	T&H used		0.2		A0VREF – 0.2	V
		ADCA0ImS	A0VREF ≥ EVCC		A0VSS		EVCC	V
			A0VREF < EVCC		A0VSS		A0VREF	V
Operation current	IA0VREF	T&H not used				1.1	3.0	mA
		T&H used (max. 6 pins)					[*2]	mA
STOP, DeepSTOP, Cyclic STOP current (@LPS is stopped)	IA0VREFS					1	10	µA
T&H current	ITH					0.5	1.3	mA/ch
T&H sampling time	t_{THSMP}				450			ns
T&H hold time	t_{THHOLD}						10	µs
Set up time of self diagnosis voltage circuit	t_{BOOT}				500			ns
Set up time of self diagnosis voltage level	t_{OUT}				500			ns
Pull-down resistor for diagnosis of open pins		ADCnIm pins	VI = AnVREF		350	500	650	kΩ
		ADCnImS pins	A0VREF ≥ EVCC: VI = EVCC A0VREF < EVCC: VI = A0VREF		100	215	800	kΩ
Accuracy of self-diagnosis function (except diagnosis of open pins)	TESH0SN	12-bit mode	Self-diagnosis voltage level = AnVREF		4015-|TOEn|		4095	—
			Self-diagnosis voltage level = 2/3AnVREF		2651-|TOEn|	2731	2811+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		1968-|TOEn|	2048	2128+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		1285-|TOEn|	1365	1445+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		80+|TOEn|	—
		10-bit mode	Self-diagnosis voltage level = AnVREF		1003-|TOEn|		1023	—
			Self-diagnosis voltage level = 2/3AnVREF		663-|TOEn|	683	703+|TOEn|	—
			Self-diagnosis voltage level = 1/2AnVREF		492-|TOEn|	512	532+|TOEn|	—
			Self-diagnosis voltage level = 1/3AnVREF		321-|TOEn|	341	361+|TOEn|	—
			Self-diagnosis voltage level = AnVSS		0		20+|TOEn|	—

(2/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Integral nonlinearity error*1	ILEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±2.0	LSB
				ADCA0I0-5 (T&H used)			±3.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.5	LSB
				ADCAnImS			±2.5	LSB
Differential nonlinearity error*1	DLEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±1.0	LSB
				ADCA0I0-5 (T&H used)			±2.0	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±3.0	LSB
				ADCA0I0-5 (T&H used)			±4.0	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0V to 4.5V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Zero scale error (offset error)*1	ZSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB
Full scale error*1	FSEn	12-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm (T&H not used)			±3.5	LSB
				ADCA0I0-5 (T&H used)			±5.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm (T&H not used)			±5.5	LSB
				ADCA0I0-5 (T&H used)			±7.5	LSB
		10-bit mode	AnVREF = 4.5 V to 5.5 V	ADCAnIm			±0.5	LSB
				ADCAnImS			±1.5	LSB
			AnVREF = 3.0 V to 4.5 V	ADCAnIm			±1.0	LSB
				ADCAnImS			±2.0	LSB

Note: Conversion accuracy when ADCA0ImS terminal is converted in 12-bit mode: Conversion accuracy can be applied if lower 2-bit is ignored from conversion result.

Note 1. This does not include quantization error.

Note 2. 3.0 + 1.3 × (the number of used T&H)

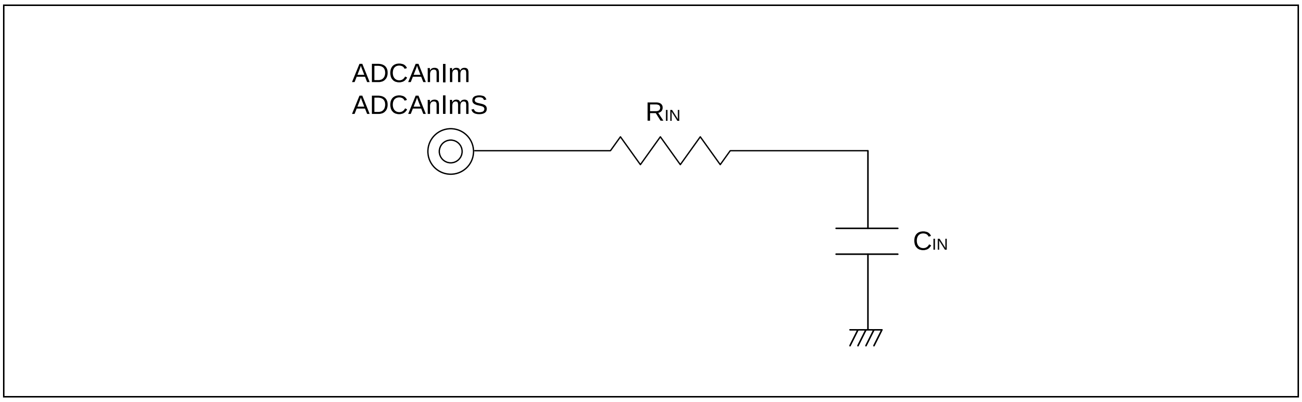
Note 3. Include the oscillation accuracy of HS IntOSC.

Note 4. When the external multiplexer is used, the detailed time of A/D conversion is MPX setup time, sampling time and successive approximation time. MPX setup time is same as “sampling time + successive approximation time”.

CAUTION

When an external digital pulse is applied to AP0, P8, and P9 pins during an A/D conversion this may lead to an A/D conversion result with a larger conversion error as expected due to the coupling noise of the external digital pulse.
The same behavior may apply when the digital buffer is used as an output pin. For the output port the potential degradation increases with the driven total output current of the port. In addition the conversion resolution may drop if the output current fluctuates at adjacent pins due to the coupling effect of the external circuit connected to these port pins.

47C.6.1 Equivalent Circuit of the Analog Input Block



Terminals	Condition	RIN (kΩ)	CIN (pF)
ADCA0I0 to 5	When T&H is used	14.2	2.1
	When T&H is not used	4.2	2.1
ADCA0I6 to 15	—	4.2	2.1
ADCA0I0S to 3S, 5S to 16S	—	5.6	9.5
ADCA0I4S, 17S to 19S	—	6.2	9.5

CAUTION

This specification is not tested during outgoing inspection. Therefore RIN and CIN are reference values only and not guaranteed. In addition these values are specified as maximum values.

47C.7 Flash Programming Characteristics

47C.7.1 Code Flash

The code flash memory is shipped in the erased state. If the code flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 μF ±30%, CISOVCL: 0.1 μF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.19 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*[3]		4*[4]		30	MHz
Number of rewrites*[1]	CWRT	Data retention of 20 years*[2]	1000			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 1000), the device can be erased “n” times for each block. For example, when a block of 32 KB is erased after 256 bytes of writing have been performed for different addresses 128 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the code flash memory.

Note 3. f_{PCLK} = 1/4 f_{CPUCLK_M}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47C.20 Programming Characteristic

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		f_{PCLK} ≥ 20 MHz CWRT < 100 times	256 B		0.4*[1]	6*[1]	ms
			8 KB		20	90	ms
			32 KB		80	360	ms
			256 KB		0.6	2.7	s
			384 KB		0.9	4.1	s
			512 KB		1.2	5.4	s
			768 KB		1.7	8.1	s
			1 MB		2.3	10.8	s
		f_{PCLK} ≥ 20 MHz CWRT ≥ 100 times	256 B		0.5*[1]	7.2*[1]	ms
			8 KB		24	108	ms
			32 KB		96	432	ms
			256 KB		0.7	3.3	s
			384 KB		1.1	4.9	s
			512 KB		1.4	6.5	s
			768 KB		2.1	9.8	s
			1 MB		2.7	13	s
Erase time		f_{PCLK} ≥ 20 MHz CWRT < 100 times	8 KB		39	120	ms
			32 KB		141	480	ms
			256 KB		1.2	3.5	s
			384 KB		1.7	5.3	s
			512 KB		2.3	7	s
			768 KB		3.4	10.5	s
			1 MB		4.5	14	s
		f_{PCLK} ≥ 20 MHz CWRT ≥ 100 times	8 KB		47	144	ms
			32 KB		169	576	ms
			256 KB		1.4	4.2	s
			384 KB		2.1	6.3	s
			512 KB		2.7	8.4	s
			768 KB		4.1	12.6	s
			1 MB		5.4	16.8	s

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

47C.7.2 Data Flash

The data flash memory is shipped in the erased state. If the data flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Condition: REGVCC = EVCC = VPOC to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Table 47C.21 Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{PCLK}*3		4*4		30	MHz
Number of rewrites*1	CWRT	Data retention 20 years*2	125 k			times
		Data retention 3 years*2	250 k			times

Note 1. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n” (n = 125000), the device can be erased “n” times for each block. For example, when a block of 64 bytes is erased after 4 bytes of writing have been performed for different addresses 168 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 2. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the data flash memory.

Note 3. f_{PCLK} = 1/4 f_{CPUCLK_M}: System operating frequency for internal flash.

Note 4. Only for program/erase operation.

Table 47C.22 Programming Characteristics

Item	Symbol	Condition	Block Size	MIN.	TYP.	MAX.	Unit
Programming time		$f_{PCLK} \geq 20$ MHz	4 B		0.16*1	1.7*1	ms
			32 KB		1.4	6.8	s
			64 KB		2.79	13.44	s
Erase time		$f_{PCLK} \geq 20$ MHz	64 B		1.7*1	10*1	ms
			32 KB		0.9	5.2	s
			64 KB		1.74	10.24	s
Blank check time		$f_{PCLK} \geq 20$ MHz	4 B			30*1	µs
			64 B			100*1	µs
			32 KB			35.2	ms
			64 KB			70.4	ms

Note 1. Only the processing time of the hardware. The overhead required by the software is not included.

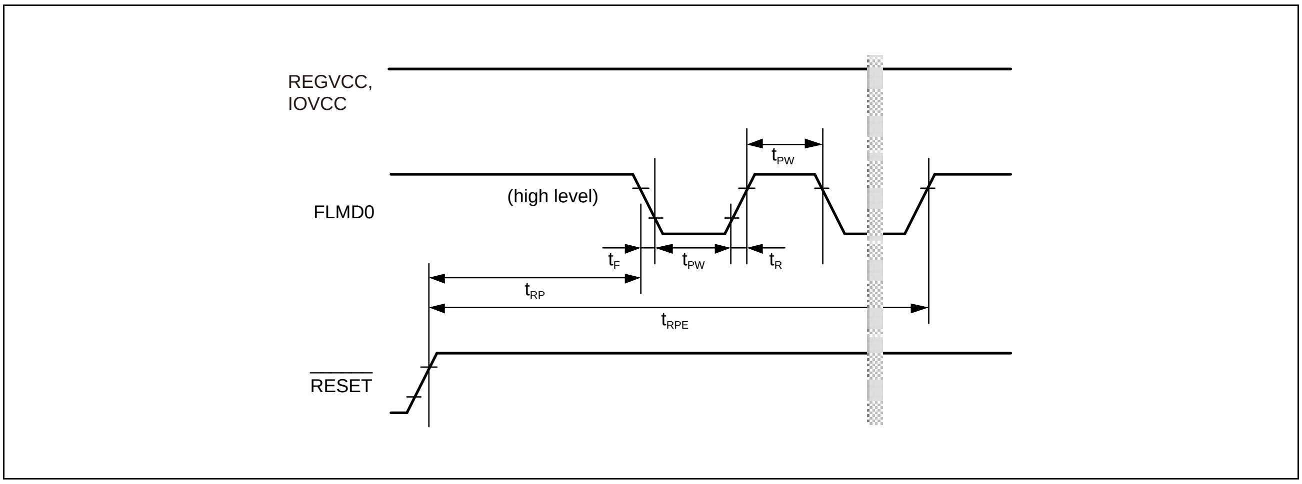
47C.7.3 Serial Programming Interface

47C.7.3.1 Serial Programmer Setup Timing

Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0 pulse input start time	t_{RP}		1.5			ms
FLMD0 pulse input end time	t_{RPE}				101.5	ms
FLMD0 low/high level width	t_{PW}		1.6			µs
FLMD0 rise time	t_R				20	ns
FLMD0 fall time	t_F				20	ns

Note: IOVCC: EVCC = A0VREF



47C.7.3.2 Flash Programming Interface

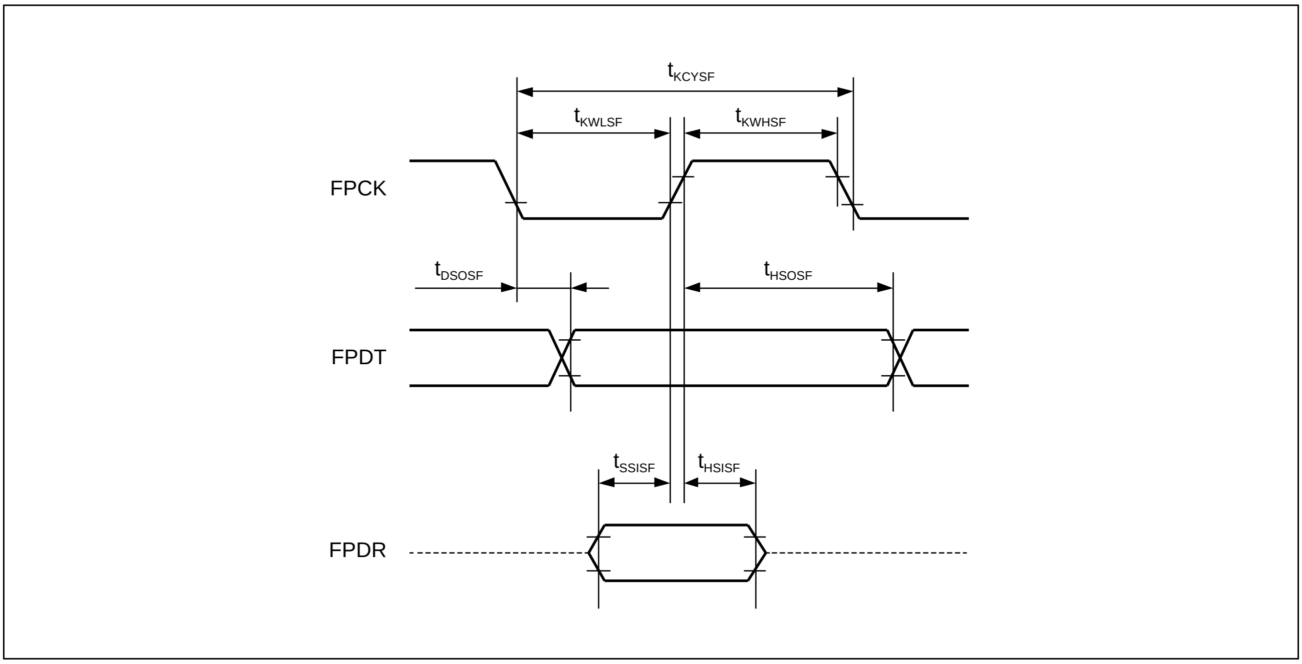
Condition: REGVCC = EVCC = 3.0 V to 5.5 V, A0VREF = 3.0 V to 5.5 V, AWOVSS = ISOVSS = EVSS = A0VSS = 0 V,
CAWOVCL: 0.1 µF ±30%, CISOVCL: 0.1 µF ±30%,
Tj = –40 to (depend on the product) °C, CL = 30 pF

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Flash Programming transfer rate		1-wired UART mode			1	Mbps
		2-wired UART mode			1.5	Mbps
FPCK cycle time	t_{KCYSF}	3-wired clock sync mode	200*[1]			ns
FPCK high level width	t_{KWHSF}	3-wired clock sync mode	t_{KCYSF} / 2 – 15			ns
FPCK low level width	t_{KWLSF}	3-wired clock sync mode	t_{KCYSF} / 2 – 15			ns
FPDR setup time (vs. FPCK)	t_{SSISF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns
FPDR hold time (vs. FPCK)	t_{HSISF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns
FPDT output delay (vs. FPCK)	t_{DSOSF}	3-wired clock sync mode Not continuous transfer (data: 1st bit)			0	ns
		3-wired clock sync mode Not continuous transfer (data: except 1st bit)			$-t_{KWHSF} + 3 \times t_{Pcyc} + 36$	ns
FPDT hold time (vs. FPCK)	t_{HSOSF}	3-wired clock sync mode	$t_{Pcyc} \times 2$			ns

Note 1. Input an external clock that is more than 6 clocks of PCLK.

NOTE

t_{Pcyc} is period of PCLK.



47C.8 Thermal Characteristics

47C.8.1 Parameters

Package	Item	Symbol	Estimate	Unit	Note
100-pin LQFP	Thermal Resistance	Θja	44.3	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	34.7		
80-pin LQFP	Thermal Resistance	Θja	44.5	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	34.0		
64-pin LQFP	Thermal Resistance	Θja	45.3	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	33.8		
48-pin LQFP	Thermal Resistance	Θja	47.0	°C/W	Conforming to JESD51-7 (4 layers)
	Thermal Characterization Parameter	ψjb	33.8		

NOTE

The thermal resistance and characterization parameters depend on the usage environment.

47C.8.2 Board

Conforming to JESD51-7 (4 layers)

	Board Size (mm)		Area (mm^2)
	X	Y	
Board	76.2	114.3	8709.66
Remaining copper rates	Thickness of conductors		
50-95-95-50%	70-35-35-70 μm		

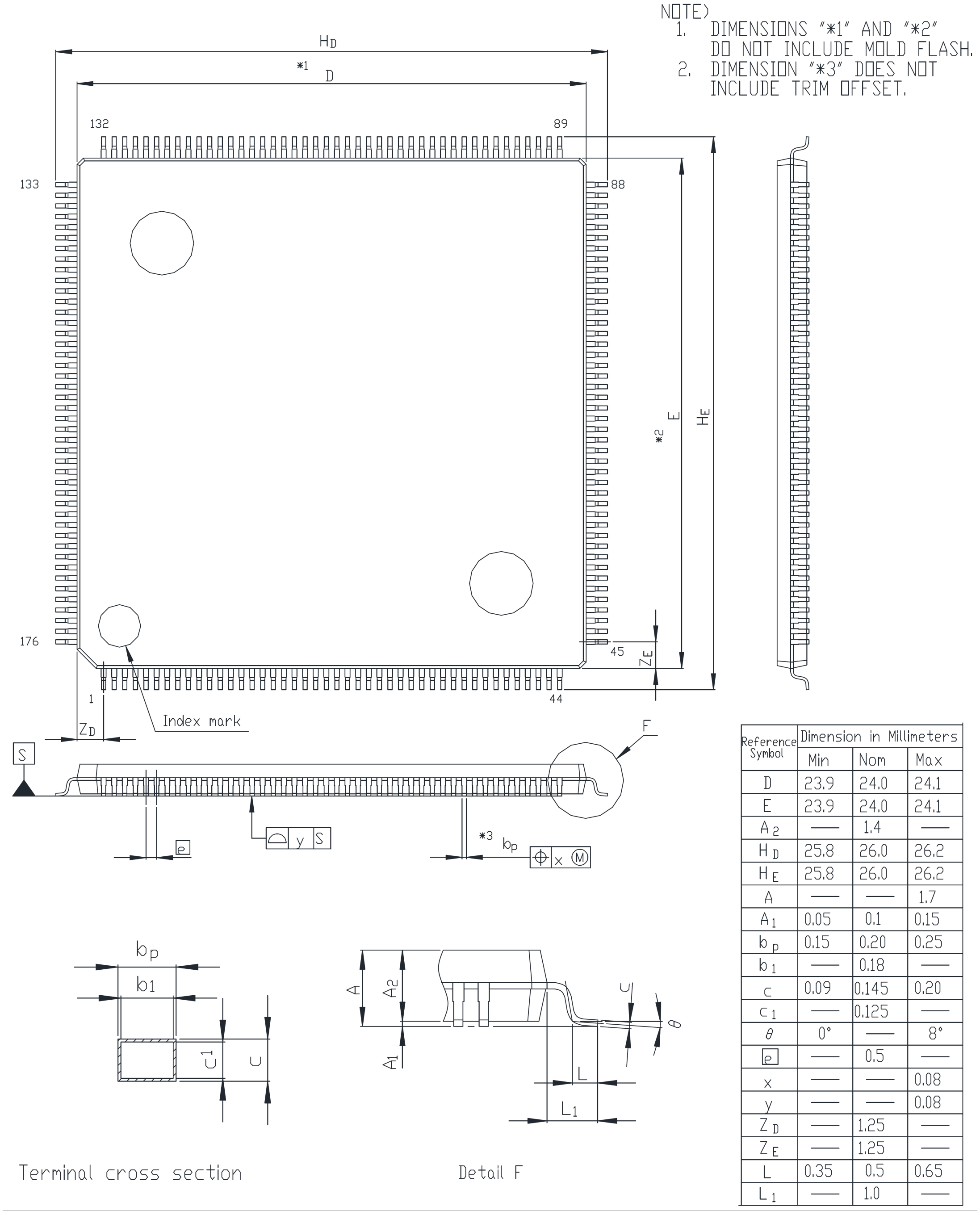
Appendix A Package

A.1 Package Dimensions of RH850/F1KH-D8

A.1.1 176 Pins

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-LFQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A / FP-176E / FP-176EV	1.8

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

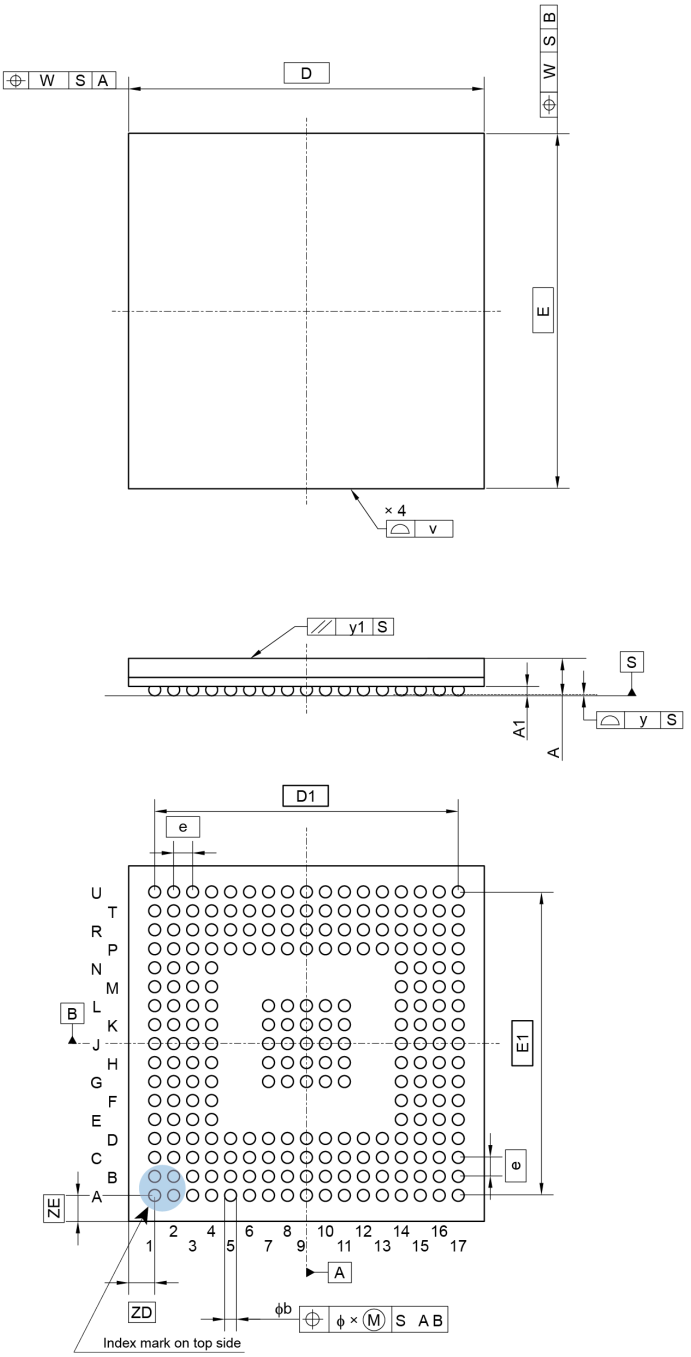


Reference Symbol	Dimension in Millimeters Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.1.2 233 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-FBGA233-15x15-0.80	PRBG0233GA-A	—	0.75

Unit: mm



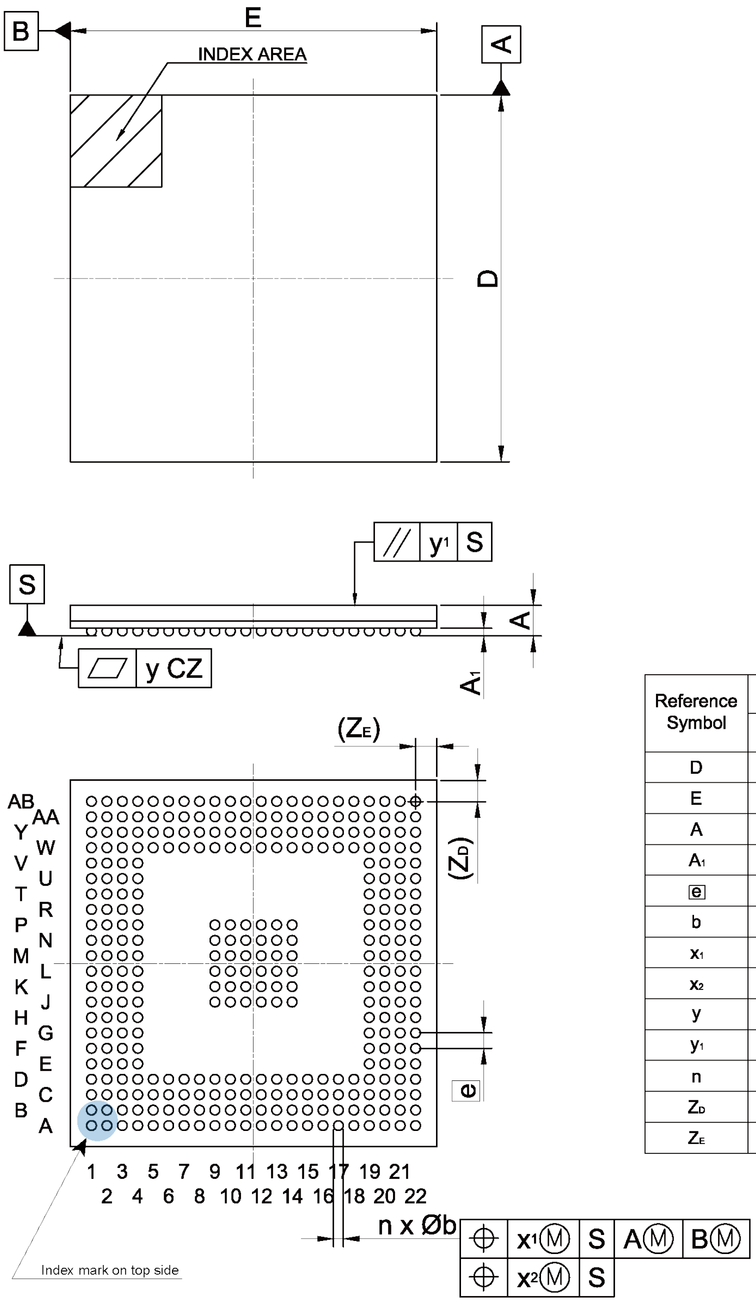
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	—	15.00	—
D1	—	12.80	—
E	—	15.00	—
E1	—	12.80	—
v	—	—	0.15
w	—	—	0.20
e	—	0.80	—
A	—	1.58	1.90
A1	0.30	0.35	0.40
b	0.49	0.54	0.59
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.20
ZD	—	1.10	—
ZE	—	1.10	—

Note: The index mark on the top side can be displayed by different characters (e.g. triangle, square).

A.1.3 Reserved

A.1.4 324 Pins

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-FBGA324-19x19-0.80	PRBG0324GB-A	1.10



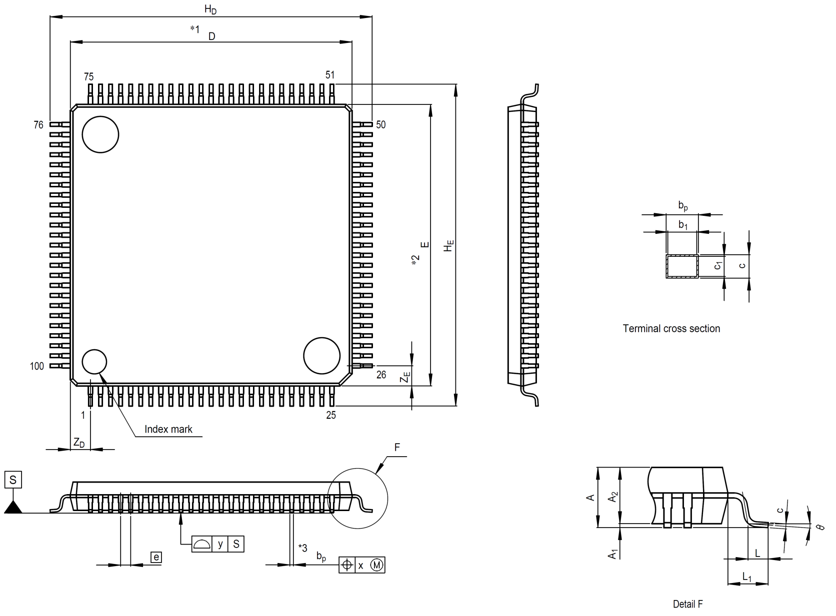
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	18.90	19.00	19.10
E	18.90	19.00	19.10
A	—	—	2.00
A_1	0.30	0.35	0.40
$\boxed{e}$	—	0.80	—
b	0.49	0.54	0.59
x_1	—	—	0.15
x_2	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
n	—	324	—
Z_D	—	1.10	—
Z_E	—	1.10	—

Note: The index mark on the top side can be displayed by different characters (e.g. triangle, square).

A.2 Package Dimensions of RH850/F1KM-S4

A.2.1 100 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g

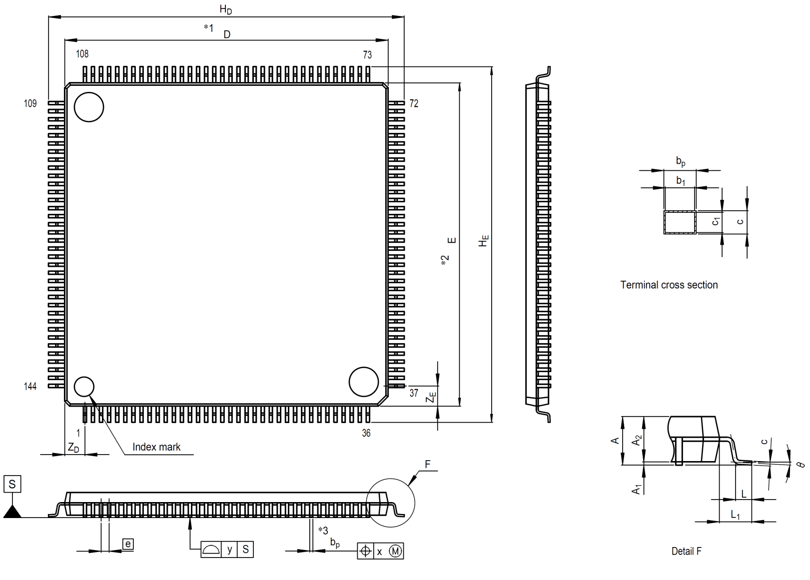


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.2.2 144 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g



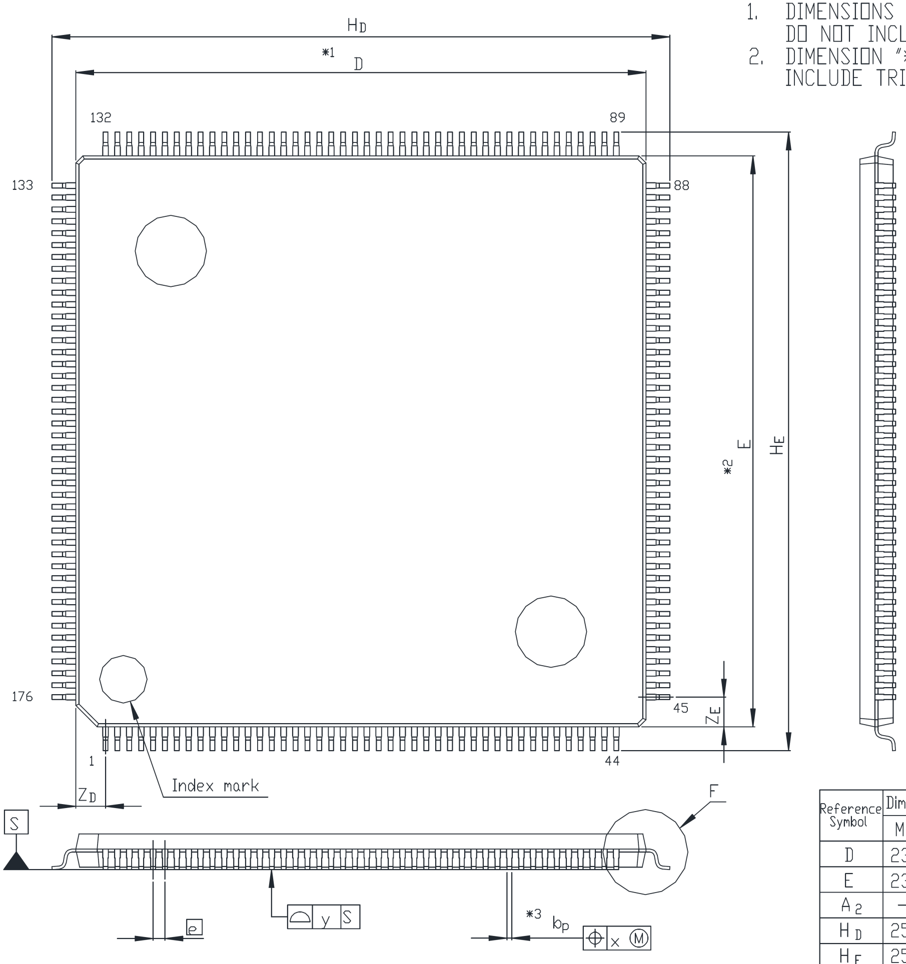
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

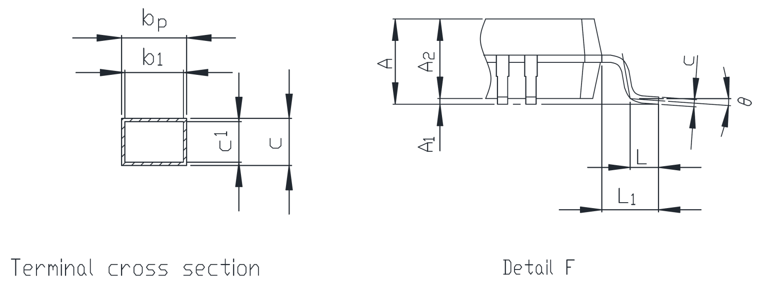
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.2.3 176 Pins

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-LFQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A / FP-176E / FP-176EV	1.8

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.





Terminal cross section

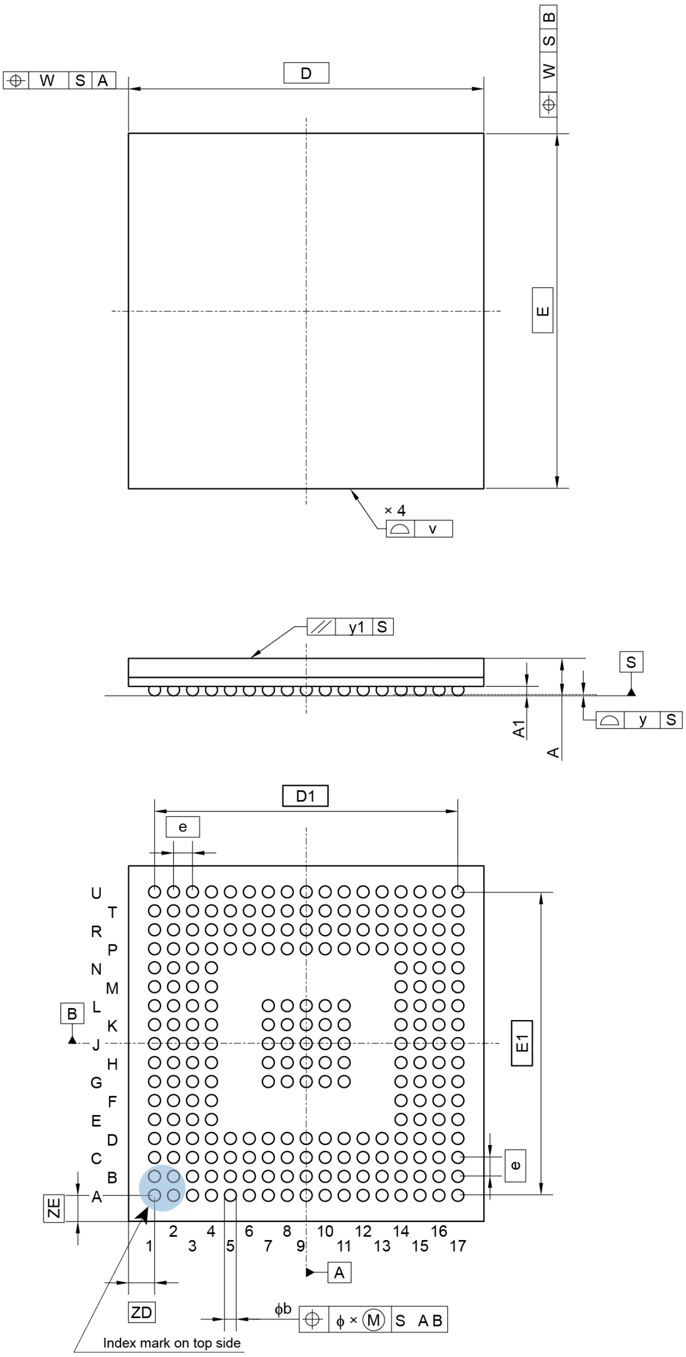
Detail F

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.2.4 233 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-FBGA233-15x15-0.80	PRBG0233GA-A	—	0.75

Unit: mm



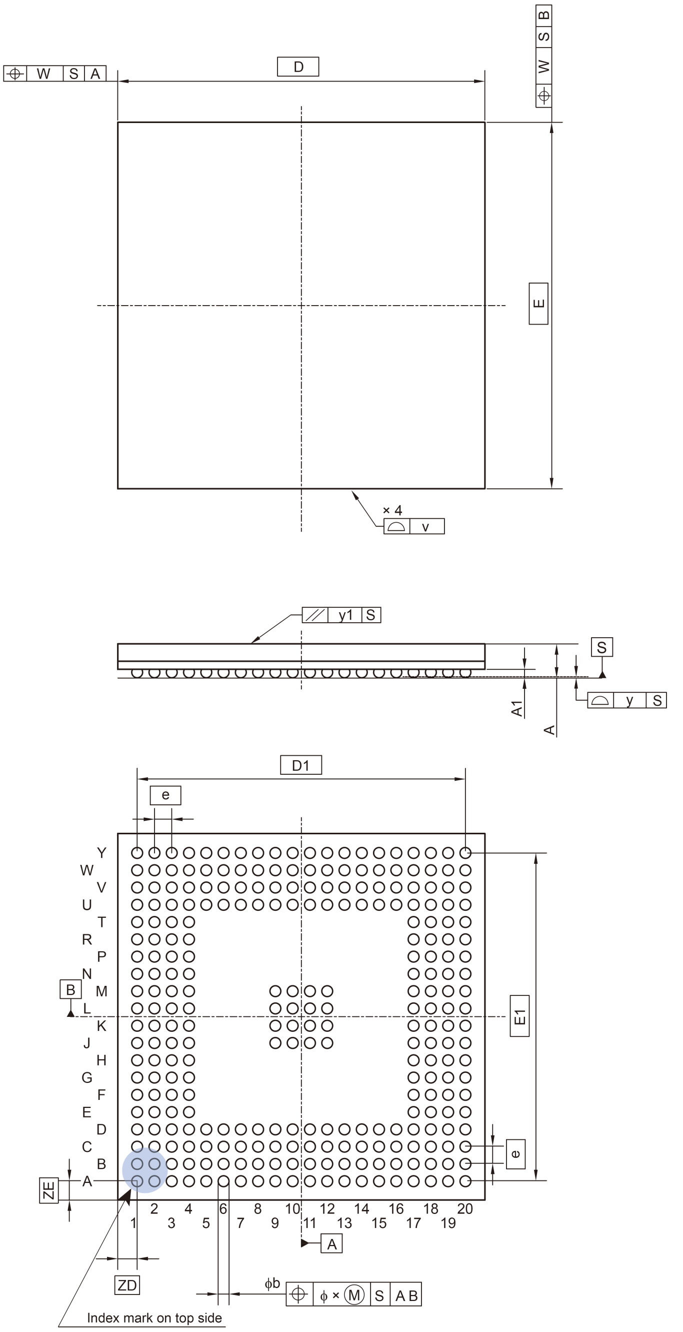
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	—	15.00	—
D1	—	12.80	—
E	—	15.00	—
E1	—	12.80	—
v	—	—	0.15
w	—	—	0.20
e	—	0.80	—
A	—	1.58	1.90
A1	0.30	0.35	0.40
b	0.49	0.54	0.59
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.20
ZD	—	1.10	—
ZE	—	1.10	—

Note: The index mark on the top side can be displayed by different characters (e.g. triangle, square).

A.2.5 272 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-FBGA272-17x17-0.80	PRBG0272GB-A	—	0.90

Unit: mm



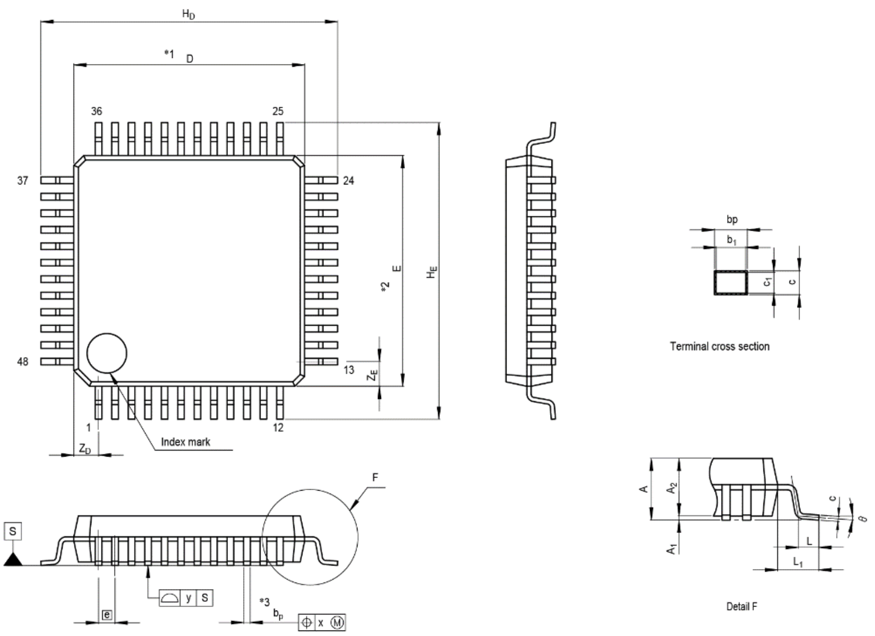
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	—	17.00	—
D1	—	15.20	—
E	—	17.00	—
E1	—	15.20	—
v	—	—	0.15
w	—	—	0.20
e	—	0.80	—
A	—	1.58	2.00
A1	0.30	0.35	0.40
b	0.49	0.54	0.59
x	—	—	0.08
y	—	—	0.10
y1	—	—	0.20
ZD	—	0.90	—
ZE	—	0.90	—

Note: The index mark on the top side can be displayed by different characters (e.g. triangle, square).

A.3 Package Dimensions of RH850/F1KM-S1

A.3.1 48 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-A	48P6Q-A	0.2g

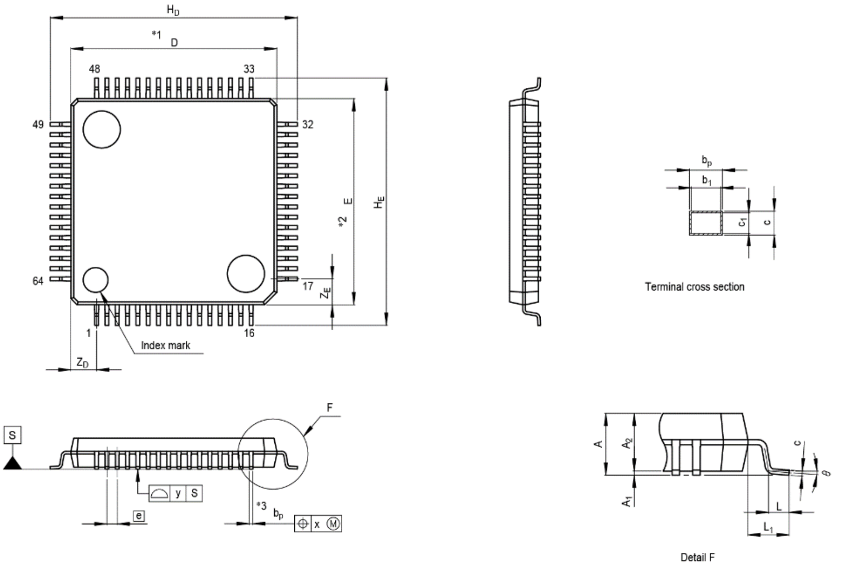


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.75	—
Z_E	—	0.75	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.3.2 64 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP64-10x10-0.50	PLQP0064KB-A	64P6Q-A / FP-64K / FP-64KV	0.3g

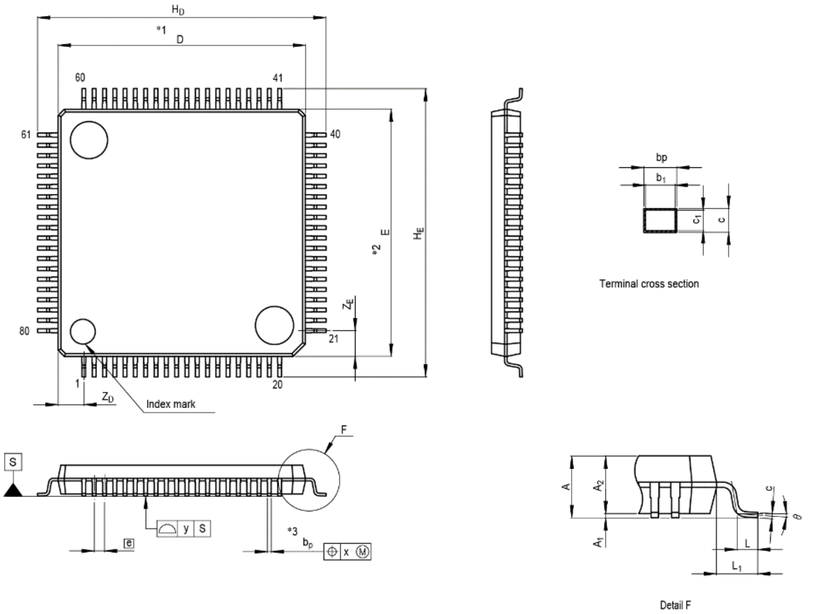


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

A.3.3 80 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP80-12x12-0.50	PLQP0080KB-A	80P6Q-A	0.5g

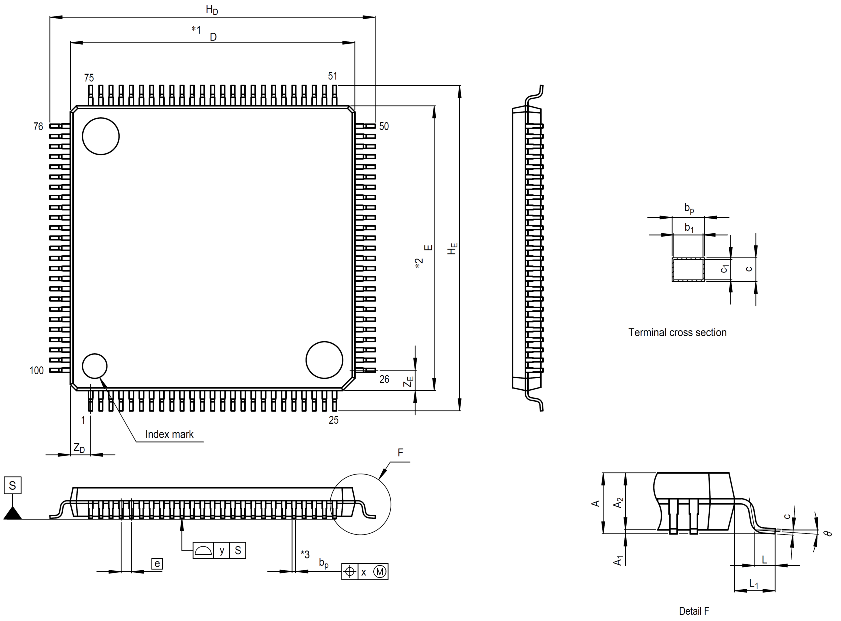


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	10°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.3	0.5	0.7
L_1	—	1.0	—

A.3.4 100 Pins

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

RH850/F1KH, RH850/F1KM User’s Manual: Hardware

Publication Date:	Rev.0.40	Oct 07, 2016
	Rev.1.10	Dec 26, 2018

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 5.2

RH850/F1KH, RH850/F1KM

Renesas Electronics Corporation